Mémoire d'habilitation à diriger des recherches

Cryptographie basée sur les corps quadratiques : cryptanalyses, primitives et protocoles

Guilhem Castagnos

Après avis des rapporteurs :

David Pointcheval	Directeur de recherche	École normale supérieure Paris
Renate Scheidler	Professeure	Université de Calgary
Damien Stehlé	Professeur	École normale supérieure de Lyon

Soutenu le 8 novembre 2019 devant le jury composé de :

Pierrick Gaudry	Directeur de recherche	Loria
Louis Goubin	Professeur	Université de Versailles-Saint-Quentin
David Pointcheval	Directeur de recherche	École normale supérieure Paris
Damien Stehlé	Professeur	École normale supérieure de Lyon
Damien Vergnaud	Professeur	Sorbonne université
Gilles Zémor	Professeur	Université de Bordeaux

Contents

Introduction

This manuscript presents my results in cryptology based on class groups of quadratic fields. In a nutshell, my research in this area has begun ten years ago with a cryptanalysis of the NICE family of cryptosystems. After that, I designed a new linearly homomorphic scheme based on tools studied for this cryptanalysis. Then, my work in this field has focused on applications and developments of this scheme in order to design efficient cryptographic protocols.

Let us take a look at the path that led to these results. At the core of this document is the notion of *linearly homomorphic encryption scheme*. A public key encryption scheme is said to be linearly homomorphic if it is possible to compute linear combinations on ciphertexts without knowing the secret key. To be a bit more precise, from a set of ciphertexts and the public key it is possible to efficiently generate a ciphertext which, when decrypted, gives a linear combination of the underlying plaintexts. While they offer minimal capabilities in terms of operations on ciphertexts compared to fully homomorphic encryption schemes, these linearly homomorphic encryption schemes are the basis of many efficient protocols, *e.g.*, electronic voting, private information retrieval, advanced encryption schemes, and secure multiparty computation.

One of the most versatile and efficient linearly homomorphic encryption scheme was proposed by *Paillier* in [Pai99]. This scheme is based on the properties of the kernel of the surjection from the group of invertible integers modulo N^2, $(\mathbf{Z}/N^2\mathbf{Z})^\times$, to the group of invertible integers modulo N, $(\mathbf{Z}/N\mathbf{Z})^\times$, where N is an RSA integer.

The first use of *ideal class groups of imaginary quadratic fields in cryptography* is due to Buchmann and Williams, in [BW88], with a Diffie-Hellman key exchange. After this work, several systems based on the discrete logarithm problem in class groups of imaginary quadratic field and adaptations in real quadratic fields where proposed during the nineties. At the end of this decade, a new family of cryptosystems based on class groups of imaginary quadratic fields, called NICE, for *New Ideal Coset Encryption*, was introduced by Paulus and Takagi in [PT98]. Jacobson, Scheidler and Weimer have then designed an adaptation of this scheme in the context of real quadratic fields in [JSW08].

The security of the NICE schemes is related to the problem of factoring integers and their main feature is that their decryption algorithm has a quadratic complexity as no exponentiation is required when decrypting. At the heart of this main feature is the properties of the kernel of the surjection between the class group of a non maximal order of discriminant $\Delta_q = -pq^2$, $C(\mathcal{O}_{\Delta_q})$, to the class group of a maximal order of discriminant $\Delta_K = -p$, $C(\mathcal{O}_{\Delta_K})$, where p and q are some distinct primes, and q is called the conductor.

During my PhD thesis, I studied many variants of the Paillier cryptosystem and, in particular, I described an adaptation of this scheme in some quotients of quadratic fields (cf. [Cas08]). It was then natural to study a possible adaptation of Paillier in class groups of quadratic fields, using the properties of the kernel of the surjection between two class groups similarly to the kernel used by Paillier in $(\mathbf{Z}/N^2\mathbf{Z})^\times$. While studying carefully this kernel, we actually showed, in collaboration with F. Laguillaumie that the representation of an element of this kernel does not hide the value of the conductor. As a consequence, we have proposed in [CL09] *a total break of the NICE family of cryptosystems in imaginary quadratic fields*.

While the real version of NICE of [JSW08] is also based on working with two class groups, this cryptanalysis can not be directly adapted. Classes of ideals in imaginary and real quadratic fields behave quite differently. As a consequence, this system – and other schemes based on real quadratic fields – mainly uses cycles of reduced ideals. With A. Joux, F. Laguillaumie and P. Q. Nguyen, we took an approach based on binary quadratic forms and we proposed *a cryptanalysis of this real version of NICE* in [CJLN09].

After these cryptanalyses, cryptography based on class groups of quadratic fields was still of interest as the security of the systems based on the discrete logarithm problem is not affected by our attacks. Moreover this problem is harder than its counterpart in finite fields and the problem of factoring integers. Thus, the resulting schemes benefit from shorter keys.

With F. Laguillaumie we came back on the problem of designing a linearly homomorphic encryption scheme with class groups. Instead of relying on the problem of factorisation as in the NICE cryptosystem we used the discrete logarithm problem and we have shown in [CL15] how to use constructively the ideas of the cryptanalysis of NICE to design *a new linearly homomorphic encryption scheme*, denoted CL in the following.

Thanks to the hardness of the discrete logarithm problem in class groups, the CL scheme has shorter parameters than Paillier, which relies on factoring, and asymptotically better performance. Last not but least, the CL scheme is homomorphic modulo a prime p and allows flexibility on the choice of this prime and its size with respect to the security parameter. This is not the case for Paillier which is homomorphic modulo N where N is an RSA integer whose size is fixed by the security level. As a result when used as a building block to design more complex cryptographic protocols, the CL scheme and its prime p might be more suited than Paillier and its RSA integer N. Indeed N might be too large or not adapted to cryptographic proofs.

We have obtained several results in this vein. In the context of *secure two party computation* we have shown that working with a linearly homomorphic encryption scheme modulo a prime such as CL or variants allows to simplify protocols, to reduce the number of rounds and the number of bits exchanged by the parties. Such a result was obtained with L. Imbert and F. Laguillaumie in [CIL17], in the context of *encryption switching protocols*, a primitive introduced by Couteau, Peters and Pointcheval in [CPP16], and with D. Catalano, F. Laguillaumie, F. Savasta and I. Tucker in [CCL+19] where we revisit a protocol of Lindell [Lin17] for *two-party ECDSA signing*.

We also showed that variants of CL can be used to obtain more advanced encryption schemes. Together with F. Laguillaumie and I. Tucker ([CLT18]), we have designed efficient inner product functional encryption schemes, a special case of functional encryption introduced by Abdalla, Bourse, De Caro and Pointcheval in [ABDP15] and further developed by Agrawal, Libert and Stehlé in [ALS16].

The manuscript is organised as follows. Chapter I reviews class groups of quadratic fields and their use in cryptography. In particular, we present in this chapter the properties of the kernel of the surjection between two class groups that will be use later for cryptanalysis and the design of CL. Chapter II is devoted to the cryptanalysis of the NICE family of cryptosystems. Chapter III presents the CL scheme and variants together with the framework of a cyclic group with an easy DL subgroup. Chapter IV presents the protocols built upon the CL scheme. Then we conclude with some perspectives.

These chapters present the main ideas and results with links to the proofs in the corresponding articles provided in the appendices. These articles are :

[CL09] G. Castagnos and F. Laguillaumie. On the security of cryptosystems with quadratic decryption: The nicest cryptanalysis. In *EUROCRYPT 2009*, *LNCS* 5479, pages 260–277. Springer, Heidelberg, April 2009.

[CJLN09] G. Castagnos, A. Joux, F. Laguillaumie, and P. Q. Nguyen. Factoring pq^2 with quadratic forms: Nice cryptanalyses. In *ASIACRYPT 2009*, *LNCS* 5912, pages 469–486. Springer, Heidelberg, December 2009.

[CL15] G. Castagnos and F. Laguillaumie. Linearly homomorphic encryption from DDH. In *CT-RSA 2015*, *LNCS* 9048, pages 487–505. Springer, Heidelberg, April 2015.

[CIL17] G. Castagnos, L. Imbert, and F. Laguillaumie. Encryption switching protocols revisited: Switching modulo p. In *CRYPTO 2017, Part I*, *LNCS* 10401, pages 255–287. Springer, Heidelberg, August 2017.

[CLT18] G. Castagnos, F. Laguillaumie, and I. Tucker. Practical fully secure unrestricted inner product functional encryption modulo p. In *ASIACRYPT 2018, Part II*, *LNCS* 11273, pages 733–764. Springer, Heidelberg, December 2018.

[CCL+19] G. Castagnos, D. Catalano, F. Laguillaumie, F. Savasta, and I. Tucker. Two-party ecdsa from hash proof systems and efficient instantiations. In *CRYPTO'19*, 2019.

Chapter I

Cryptography based on Quadratic Fields

In this chapter we give a presentation of class groups of quadratic fields with a focus on the properties needed for cryptography. We first start with class groups of maximal orders and their cryptographic applications. We then move to non maximal orders, which are the basis of the NICE family of cryptosystems and of the CL scheme. We will give some properties of the kernel of the surjection between the class group of a non maximal order to the class group of the maximal order. These properties were proven in [CL09, CJLN09, CL15] and will be used in Chapter II for the cryptanalysis of NICE and in Chapter III for the design of CL.

The presentation will be informal. For more details, the interested reader is referred to [Cox99] for the mathematical aspects, to [Coh00, BV07, JW09] for algorithms and cryptographic applications.

I.1. Class Groups of Maximal Orders

We give here an informal presentation of class groups of maximal orders. A more formal (but concise) presentation can be found in [CJLN09, Section B.2].

Let $D \neq 0,1$ be a squarefree integer and consider the quadratic number field $K = \mathbf{Q}(\sqrt{D})$. If $D < 0$ (resp. $D > 0$), K is called an *imaginary* (resp. a *real*) quadratic field. The *fundamental discriminant* Δ_K of K is defined as $\Delta_K = D$ if $D \equiv 1 \pmod 4$ and $\Delta_K = 4D$ otherwise.

The ring $\mathscr{O}_{\Delta_K}$ of *algebraic integers* in K, which is also the *maximal order* of K can be written as $\mathbf{Z} + \omega_K\mathbf{Z}$, where $\omega_K = \frac{1}{2}(\Delta_K + \sqrt{\Delta_K})$.

Class Groups

In general the fundamental theorem of arithmetic fails to hold in $\mathcal{O}_{\Delta_K}$: it is not a unique factorisation domain. However this property is always true for ideals: every non zero proper ideal of $\mathcal{O}_{\Delta_K}$ factors into a product of prime ideals as $\mathcal{O}_{\Delta_K}$ is a Dedekind domain. Moreover, $\mathcal{O}_{\Delta_K}$ is a unique factorisation domain if and only if it is a principal ideal domain. The ideal class group of $\mathcal{O}_{\Delta_K}$ can be viewed as a measure of this obstruction to being a principal ideal domain or equivalently to have a unique factorisation. In order to define a group with the ideals of $\mathcal{O}_{\Delta_K}$, and in particular to have invertible elements, one has to consider the set of non zero fractional ideals of $\mathcal{O}_{\Delta_K}$, $I(\mathcal{O}_{\Delta_K})$. Then, the *ideal class group* of $\mathcal{O}_{\Delta_K}$ is defined as

$$C(\mathcal{O}_{\Delta_K}) = I(\mathcal{O}_{\Delta_K})/P(\mathcal{O}_{\Delta_K}),$$

where $P(\mathcal{O}_{\Delta_K})$ is the subgroup consisting of principal fractional ideals.

This group is trivial if and only if $\mathcal{O}_{\Delta_K}$ is a principal ideal domain. For imaginary quadratic fields, it occurs only 9 times, for $D = -1, -2, -3, -7, -11, -19, -43, -67, -163$. But for real quadratic fields it is fairly common (it is conjectured by the Cohen Lenstra heuristics that the proportion should be around 75%).

Cardinality

This group is finite and its cardinality is the *class number* denoted by $h(\mathcal{O}_{\Delta_K})$. For imaginary quadratic fields, one has

$$h(\mathcal{O}_{\Delta_K}) \approx \sqrt{|\Delta_K|}.$$

The difference of situation between the imaginary and real cases, is due to the unit group. When $\Delta_K < -0$, $\mathcal{O}_{\Delta_K}^\star$ is equal to $\{\pm 1\}$ except when Δ_K is equal to -3 and -4 ($\mathcal{O}_{-3}^\star$ and $\mathcal{O}_{-4}^\star$ are respectively the group of sixth and fourth roots of unity). When $\Delta_K > 0$, then the unit group has rank 1 and $\mathcal{O}_{\Delta_K}^\star = \langle -1, \varepsilon_{\Delta_K} \rangle$ where $\varepsilon_{\Delta_K} > 0$ is called the *fundamental unit*. The real number $R_{\Delta_K} = \log(\varepsilon_{\Delta_K})$ is the *regulator* of $\mathcal{O}_{\Delta_K}$. For real quadratic fields, one has $h(\mathcal{O}_{\Delta_K})R_{\Delta_K} \approx \sqrt{\Delta_K}$ and in general $h(\mathcal{O}_{\Delta_K})$ is small and

$$R_{\Delta_K} \approx \sqrt{\Delta_K}.$$

Representation of the Classes and Computation

Working with ideals modulo the equivalence relation of the class group is essentially equivalent to work with binary quadratic forms modulo the action of $\mathsf{SL}_2(\mathbf{Z})$. Every (primitive) ideal $\mathfrak{a}$ of $\mathcal{O}_{\Delta_K}$ can be written with a *two-element representation*, as

$$\mathfrak{a} = \left(a\mathbf{Z} + \frac{-b + \sqrt{\Delta_K}}{2}\mathbf{Z} \right), \tag{I.1}$$

with $a \in \mathbf{N}$ and $b \in \mathbf{Z}$ such that $b^2 \equiv \Delta \pmod{4a}$. This notation also represents the binary quadratic form $ax^2 + bxy + cy^2$ with $b^2 - 4ac = \Delta_K$. In the following, we will note $\mathfrak{a} = (a, b)$.

Computing a product of ideals in the class group corresponds to the composition of quadratic forms, already known to Gauss. It can be computed in quadratic time and even in quasi linear time (see [Sch91b]).

To represent classes, one has the notion of reduced ideals (or reduced forms). Again, the situation is different for imaginary and real quadratic fields. In the imaginary case, an ideal (or a form) is called reduced if $-a < b \leqslant a$, $a \leqslant c$, and $b \geqslant 0$ for $a = c$. In every class of $\mathcal{O}_{\Delta_K}$-ideals there exists exactly one reduced ideal and one can compute it with a reduction algorithm due to Lagrange and Gauss in quadratic time or even quasi linear time.

In the real case, the definition of a reduced ideal is a bit more complex (see [CJLN09, Def. B−3]). The reduction process is similar to the imaginary case. However, the main difference is that there will not exist a unique reduced ideal per class, but several organised in a cyclic structure: when an ideal has been reduced then subsequent applications of the reduction step give other reduced ideals. The complete cycle of the reduced ideals in a class can be obtained by a finite number of application of the reduction step as the process is periodic. The period is roughly equal to the regulator.

Hard Problems in Imaginary Quadratic Fields

In imaginary quadratic fields, the computation of the class number and the discrete logarithm problem in class groups are two hard problems. A sub exponential algorithm was proposed in 1989 by Hafner and Mc-Curley in [HM89]. Then they have been improvements (*e.g.*, [Jac98, Bia10, Kle16]) and the conjectured sub exponential complexity is of $L_{|\Delta_K|}[1/2, 1+o(1)]$, where $L_x[\alpha, c] = exp(c\log(x)^\alpha log(log(x))^{1-\alpha})$.

A record of computation of a class group of an imaginary quadratic field has been very recently set (in May 2019), in [BKV19] by Beullens, Kleinjung and Vercauteren in order to implement efficiently some isogeny based cryptosystems. They report the computation of the class group of the imaginary quadratic field central to the CSIDH-512 isogeny based cryptosystem with a 512 bits discriminant in 52 core years (with a 3.3GHz processor).

In [BJS10] (prior to this record), the authors gave estimates on the sizes of a discriminant Δ_K and of an RSA modulus N to have a discrete logarithm problem in $C(\mathcal{O}_{\Delta_K})$ as hard as the problem of factorisation of N. Recall that the best algorithm to factor N has complexity $L_N[1/3, \sqrt[3]{64/9} + o(1)]$. Using the standard estimation of levels of security for RSA, one obtains the bit sizes of Fig. I.1, that we have used for all our cryptographic applications.

These estimates do not seem to need to be revised upwards in light of the recent record. Let us give some intuition on that. To calibrate their estimates, the authors of [BJS10] use the current record on factorisation, RSA-768 (cf. [KAF+10]). This computation is estimated equivalent to computing a discrete logarithm in a class group of a 640 bits negative discriminant. Let us see if it is large enough compared to the 512 bits record. In [KDL+17], the cost of factoring RSA-768 is estimated to have taken 1700 core years (with a 2.2GHz processor). Let us risk extrapolating the cost of computing a discrete logarithm with a 640 bits discriminant from the record of [BKV19]: we com-

Security Parameters	N	Δ_K
112	2048	1348
128	3072	1827
192	7680	3598
256	15360	5971

Figure I.1: Comparison of bit sizes for factoring N and computing DLP in $C(\mathcal{O}_{\Delta_K})$.

pute $L_{2^{640}}[1/2,1]/L_{2^{512}}[1/2,1] \approx 572$, which gives $52 \times 572 = 29744$ core years: an order of magnitude more than the record on factoring. This rough estimate seems to indicate that Fig. I.1 provides a good safety margin.

For cryptographic applications of class groups of imaginary quadratic fields, parameters can thus be chosen smaller than for RSA. An element of the class group is represented by the reduced ideal of the class, so with the two element representation (a, b) of eq. I.1, where $|b| \leqslant a$ and $a \leqslant \sqrt{\Delta_K/3}$. As a consequence the bit size of the representation of an element of $C(\mathcal{O}_{\Delta_K})$ is roughly the bit size of Δ_K. We see from Fig. I.1 that there will be always a gain in term of bit size compared to factoring based cryptosystem, moreover the gain dramatically improves for high levels of security.

Note that it seems unlikely that there will be major improvements on algorithms for imaginary quadratic class groups to reach a $L_{|\Delta_K|}[1/3, \cdot]$ complexity as argued in [BH03] by Bauer and Hamdy.

In general, for cryptosystems based on the discrete logarithm problem, one considers cyclic groups of prime order, to avoid small factors that can be used to speed up the computation of discrete logarithms with the Pohlig-Hellman algorithm or breaking the Diffie-Hellman assumption.

The particularity of $C(\mathcal{O}_{\Delta_K})$ is that computing the class number is hard. So the *order will be unknown* in general in cryptographic applications, and there is no guarantee that it is not divisible by small primes.

In [HM00a], Hamdy and Möller discuss the selection of a discriminant Δ_K. It is advised to construct a fundamental discriminant Δ_K and to minimize to 2-Sylow subgroup of the class group. For example selecting $\Delta_K = -p$ where $p \equiv 3 \pmod 4$ ensures that $h(\mathcal{O}_{\Delta_K})$ is odd. Following the Cohen-Lenstra heuristics, cf. [Coh00, Chapter 5.10.1], the probability that the odd part of the class group is cyclic is 97.757% and the probability that an odd prime r divides $h(\mathcal{O}_{\Delta_K})$ is approximately $1/r + 1/r^2$. As a result, the order of the odd part will be divisible by small primes with non negligible probability. Nevertheless, as indicated in [HM00a], this does not lead to a weakness on the discrete logarithm problem, as there is no efficient algorithm to compute $h(\mathcal{O}_{\Delta_K})$ or odd multiples or factors of $h(\mathcal{O}_{\Delta_K})$, hence an adaptation of the Pohlig-Hellman algorithm is not possible. Moreover, [HM00a] argues that selecting Δ_K to defeat the index-calculus methods ensures that $h(\mathcal{O}_{\Delta_K})$ will be smooth only with negligible probability so an algorithm similar to the $(p-1)$-factoring algorithm can not be applied to compute the class number.

Hard Problems in Real Quadratic Fields

As we have seen, in real quadratic fields the class group is usually very small so can not be used for cryptographic applications based on the discrete problem. Instead one works with the cycle of reduced ideal of the principal class. This set is not a group but, as shown by Shanks, it can be equipped with a distance to get a so called *infrastructure* (see [BV07, Chapter 10]). Although this structure is not a group (but can be interpreted in terms of Arakelov class group cf. [Sch08] and [Fon08]), it can be used to adapt schemes based on the discrete logarithm problem.

Indeed, it is possible to define an analogue of exponentiation in this infrastructure. From a reduced principal ideal $\mathfrak{g}$, and an integer k, one can compute, using the composition and reduction algorithms, a reduced principal ideal $\mathfrak{h}$ very "close" in terms of the Shanks distance to the (non reduced) ideal $\mathfrak{g}^k$. We thus have an analogous of the discrete logarithm problem: from $\mathfrak{h}$ and $\mathfrak{g}$, compute k.

In [JW09, Section 14.4], it is shown that this problem can be solved by solving *the principal ideal problem*, *i.e.*, finding a generator of a principal ideal, given its two-element representation of fig. I.1, and by computing the regulator. Both problems can be solved in conjecture time $L_{\Delta_K}[1/2, 1 + o(1)]$ using index-calculus methods (cf. [BJS10]). Recommended positive discriminant bit sizes are again given in [BJS10]. They are slightly smaller than in the imaginary case, as it is noted that the infrastructure discrete logarithm problem requires more time to solve on average than the discrete logarithm in the imaginary case.

I.2. Cryptography in Maximal Orders

A survey on cryptography in imaginary quadratic fields can be found in [BH01]. Another interesting reference on the subject is the chapter 14 of the book of Jacobson and Williams [JW09], in both the real and imaginary cases. Some remarks on the adaptation in class groups of imaginary quadratic fields of the Elgamal cryptosystem can be found in the appendix of this document in [CL15, Subsection C.B.4]. We give below a non exhaustive list of cryptographic protocols in class groups of maximal orders.

Elgamal Adaptations in Imaginary Quadratic Fields

The first use of ideal class groups of imaginary quadratic fields in cryptography is due to Buchmann and Williams, in [BW88], with a Diffie-Hellman *key exchange* and a brief description of an *adaptation of the Elgamal cryptosystem* in the same setting. Efficient implementations of these cryptosystems are discussed in [BDW90,SP05,BH01] and [BV07].

At a high level, the key generation process of these adaptations of Elgamal can be sketched as follows:

- Generate Δ_K a fundamental negative discriminant, such that $|\Delta_K|$ is large enough to thwart the computation of discrete logarithms (cf. above) ;

- choose g a class of $C(\mathscr{O}_{\Delta_K})$ of even order (one expects that the order of g will be close to $h(\Delta_K) \approx \sqrt{|\Delta_K|}$) ;
- the private key is some random x and the public key is (g, h), where $h = g^x$.

To implement Elgamal, it remains the problem of the embedding of a message. Several propositions have been made, but it is not clear that they provide semantically secure schemes under the DDH assumption in class groups. As proposed in [BH01], a "hashed" version can be used. A bit-string m is encrypted as $(g^r, m \oplus H(h^r))$ where H is a random oracle. In [BV07], an adaptation of DHIES is described.

As note in [CL15, Subsection C.B.4], an important remark is that it is necessary to work in the group of squares, *i.e.*, the *principal genus*, otherwise, in a similar way to what is happening in $(\mathbf{Z}/p\mathbf{Z})^\times$, one can defeat the DDH assumption.

Digital Signatures in Imaginary Quadratic Fields

Digital signature has also been proposed in imaginary quadratic fields. The adaptation of classical schemes is more difficult than with Elgamal as the order of the group is unknown. A solution is to adapt variants of DSA *à la* Poupard-Stern that do not need the knowledge of the group order ([BH01]). This gives a scheme secure in the random oracle model under the intractability of the discrete logarithm problem. Another solution, is to use another algorithmic problem, the *root problem*: given a class h and an integer x, find g such that $h = g^x$. An adaptation of GQ signatures secure in the random oracle model under the intractability of this root problem is given in [BH01]. A variant of DSA, RDSA related to this problem was proposed by Biehl *et al.* in [BBHM02] but unfortunately broken by Fouque and Poupard in [FP03].

Paradox of the Unknown Order and Recent Developments

The difficulty of adapting signatures in imaginary quadratic fields illustrates a paradox in cryptography based on class groups. The fact that the order is unknown will be central in our developments. In particular, it makes it possible to have security while having a subgroup with an easy discrete logarithm problem in [CL15]. However, it is also a source of annoyance: it complicates the sampling of exponents, some parts of security proofs, and zero knowledge proofs (cf. [CCL+19, Subsection F.4.3]).

Recently, there have been new proposals of cryptographic protocols relying on class groups of maximal orders of imaginary quadratic fields. One can mention the work on *accumulators* of Lipmaa in [Lip12] and the recent proposal of *verifiable delay functions* in [BBBF18, Wes19]. In both works, the fact that the order is unknown is crucial to *avoid a trusted setup*.

Real Quadratic Fields

In [BW90], Buchmann and Williams gave the ideas of a counterpart in the real case of their work in imaginary quadratic fields. As seen in the previous section, this adaptation in the infrastructure of real quadratic field is challenging as this structure is not a group. There have been thus many efforts in order to give a more detailed description of this scheme and a concrete and efficient implementation of the "exponentiation" (cf. [SBW91, SBW94, HP01, JSW06, DJS12]).

Compared to exponentiation in imaginary quadratic fields, one has to perform extra operations: at each step of the exponentiation algorithm, additional reduction steps have to be done as one need to find a reduced ideal "close" to the non reduced result. Moreover, one has to compute real approximations of the generators of the principal ideals, and this must be done with sufficient accuracy in order that Alice and Bob agree to the same ideal at the end of the protocol. A detail description of this line of research can be found in [JW09, Section 14.4].

Apart form the Diffie-Hellman key exchange, an adaptation of the Elgamal signature scheme has been proposed by Biehl, Buchmann and Thiel in [BBT94] and an adaptation of the Fiat-Shamir signature protocol by Buchmann, Maurer, and Möller ([BMM00]).

I.3. Class Groups of non Maximal Orders

A more detailed background on class groups of non maximal orders can be found in [Cox99, JW09]. Let $\mathrm{K} = \mathbf{Q}(\sqrt{\mathrm{D}})$ be a quadratic field.

Orders and Class Groups

An *order* $\mathscr{O}$ in K is a subset of K such that $\mathscr{O}$ is a subring of K containing 1 and $\mathscr{O}$ is a free $\mathbf{Z}$-module of rank 2. The ring $\mathscr{O}_{\Delta_\mathrm{K}}$ of algebraic integers in K of the last section is the *maximal* order in K in the sense that if contains all the orders in K. Recall that it can be written as $\mathbf{Z} + \omega_\mathrm{K}\mathbf{Z}$, where $\omega_\mathrm{K} = \frac{1}{2}(\Delta_\mathrm{K} + \sqrt{\Delta_\mathrm{K}})$.

If we set $f = [\mathscr{O}_{\Delta_\mathrm{K}} : \mathscr{O}]$ the finite index of any order $\mathscr{O}$ in $\mathscr{O}_{\Delta_\mathrm{K}}$, then $\mathscr{O} = \mathbf{Z} + f\omega_\mathrm{K}\mathbf{Z}$. The integer f is called the *conductor* of $\mathscr{O}$. The discriminant of $\mathscr{O}$ is then $\Delta_f = f^2\Delta_\mathrm{K}$. In the following, we will note $\mathscr{O}_\Delta$ an order of discriminant Δ.

The definition of class groups generalises to non maximal orders (but not every fractional ideals is invertible). We will note as before $\mathrm{C}(\mathscr{O}_\Delta)$ the class group of $\mathscr{O}_\Delta$. Similarly, the notion of regulator can be generalised, and we will note R_Δ the regulator of $\mathscr{O}_\Delta$.

Relations between Class Groups

Let us consider $\mathscr{O}_{\Delta_f}$ an order of conductor f with $\Delta_f = f^2\Delta_\mathrm{K}$. A central fact in our work is that there is a well known correspondence between ideals of $\mathscr{O}_{\Delta_f}$ and $\mathscr{O}_{\Delta_\mathrm{K}}$ by considering ideals prime to the conductor f (equivalently whose norm, the integer a in the two-element representation, is prime to f) :

- If $\mathfrak{A}$ is an $\mathscr{O}_{\Delta_\mathrm{K}}$-ideal prime to f, then $\mathfrak{A} \cap \mathscr{O}_{\Delta_f}$ is an $\mathscr{O}_{\Delta_f}$-ideal prime to f of the same norm.
- If $\mathfrak{a}$ is an $\mathscr{O}_{\Delta_f}$-ideal prime to f, then $\mathfrak{a}\mathscr{O}_{\Delta_\mathrm{K}}$ is an $\mathscr{O}_{\Delta_\mathrm{K}}$-ideal prime to f of the same norm.

- The map $\varphi_f : \mathrm{I}(\mathscr{O}_{\Delta_f}, f) \to \mathrm{I}(\mathscr{O}_{\Delta_\mathrm{K}}, f)$, $\mathfrak{a} \mapsto \mathfrak{a}\mathscr{O}_{\Delta_\mathrm{K}}$ is an isomorphism (where $\mathrm{I}(\mathscr{O}_\Delta, f)$ is the subgroup of ideals of $\mathscr{O}_\Delta$ prime to f).

Then the map φ_f induces a surjection

$$\bar{\varphi}_f : \mathrm{C}(\mathscr{O}_{\Delta_f}) \twoheadrightarrow \mathrm{C}(\mathscr{O}_{\Delta_\mathrm{K}}).$$

All these maps can be efficiently computed knowing the conductor f. See [Cox99, Section 7] for proofs in the imaginary case and [Wei04, Chapter 3] for extension in the real case. Algorithms can be found in [HJPT98, PT00] (or in this document: [CL09, Algo. A.1, A.2]).

Moreover one has the following formula (cf. [Cox99, Theorem 7.24] in the imaginary case, or [Wei04, Chapter 2] for the general case) that relates the class numbers of the two orders, and gives the order of the kernel of $\bar{\varphi}_f$:

$$h(\mathscr{O}_{\Delta_f}) = h(\mathscr{O}_{\Delta_\mathrm{K}})\kappa f \prod_{p \mid f}\left(1 - \left(\frac{\Delta_\mathrm{K}}{p}\right)\frac{1}{p}\right),$$

where $\kappa = |\mathscr{O}_{\Delta_f}^\times|/|\mathscr{O}_{\Delta_\mathrm{K}}^\times|$ if $\Delta < 0$ and $\kappa = \mathrm{R}_{\Delta_\mathrm{K}}/\mathrm{R}_{\Delta_f}$ if $\Delta > 0$. Note that $h(\mathscr{O}_{\Delta_f})$ is always an integer multiple of $h(\mathscr{O}_{\Delta_\mathrm{K}})$. In particular, for the case of real quadratic fields, $\epsilon_{\Delta_q} = \epsilon_{\Delta_\mathrm{K}}^t$ for a positive integer t, hence $\mathrm{R}_{\Delta_q} = t\mathrm{R}_{\Delta_\mathrm{K}}$. In the following we will denote $\phi_{\Delta_\mathrm{K}}(f)$ the order of the kernel of $\bar{\varphi}_f$.

In all our settings, we will use a prime conductor $f = q$ and consider $\Delta_q = q^2\Delta_\mathrm{K}$, for a large fundamental discriminant Δ_K. In that case, the order $\phi_{\Delta_\mathrm{K}}(q)$ of the kernel of $\bar{\varphi}_q$ is given by

$$\frac{h(\mathscr{O}_{\Delta_q})}{h(\mathscr{O}_{\Delta_\mathrm{K}})} = \begin{cases} q - (\Delta_\mathrm{K}/q) & \text{if } \Delta_k < -4, \\ (q - (\Delta_\mathrm{K}/q))\mathrm{R}_{\Delta_\mathrm{K}}/\mathrm{R}_{\Delta_q} & \text{if } \Delta_k > 0. \end{cases}$$

Structure of the Kernel of $\bar{\varphi}_q$

We focus now on the case where the conductor $f = q$ is prime and consider $\Delta_q = q^2\Delta_\mathrm{K}$, for a large fundamental discriminant $\Delta_\mathrm{K} < -4$. We have just seen that the order of the kernel of $\bar{\varphi}_q$ is $q-(\Delta_\mathrm{K}/q)$. This result is actually classically obtained by proving that this kernel is isomorphic to

$$\left(\mathscr{O}_{\Delta_\mathrm{K}}/q\mathscr{O}_{\Delta_\mathrm{K}}\right)^\times / \left(\mathbf{Z}/q\mathbf{Z}\right)^\times, \tag{I.2}$$

moreover this isomorphism is effective see [CL09, Lemma A−1].

By using this effective isomorphism and a well chosen system of representatives of this quotient, we have proven the following theorem:

Theorem (A−2 [CL09])**.** *Let Δ_K be a fundamental negative discriminant, different from -3 and -4 and q an odd prime conductor. There exists an ideal of norm q^2 in each nontrivial class of* $\ker \bar{\varphi}_q$.

This theorem has been the starting point of the cryptanalysis of the NICE family of cryptosystems. This representation of $\ker\bar\varphi_q$ has also been proven useful to obtain q^2-isogeny cycles to compute classical modular polynomials $\Phi_q(X,Y)$ using graphs of q-isogenies, see [BLS12].

During the proof of this theorem, we actually show that the set of ideals of $\mathcal{O}_{\Delta_q}$ of the form (q^2, kq) for some integers k forms a system of representatives of the nontrivial classes of $\ker\bar\varphi_q$. Let us denote by $f \in \ker\bar\varphi_q$, the class of the ideal (q^2, q). Then, from the previous theorem, for an integer m, f^m contains an ideal of the form $(q^2, L(m)q)$ for some integer $L(m)$.

Suppose now that $q|\Delta_K$. In that case, the kernel is cyclic of order q and we prove in [CL15], Proposition C−1[1], that f is a generator of the kernel and that $L(m)$ has a simple expression: For $m \in \{1, \dots, q-1\}$, $L(m)$ is the odd integer in $[-q, q]$ such that

$$L(m) \equiv m^{-1} \pmod{q}.$$

Moreover, if one chooses $\Delta_K < -4q^2$, then it is easy to see that all these ideals of norm q^2 will be reduced. As a result, $\ker\bar\varphi_q$ is a cyclic subgroup of order q of $C(\mathcal{O}_{\Delta_q})$ where *the discrete logarithm problem is easy*. Any element of this kernel will be represented by a reduced ideal of the form $(q^2, L(m)q)$ and its discrete logarithm in basis f is $L(m)^{-1} \pmod{q}$.

This proposition has been fundamental to design the CL scheme. It is an analogue to the fact that modulo an integer N^2, the discrete logarithm problem is easy in the subgroup of order N of $(\mathbf{Z}/N^2\mathbf{Z})^\times$. This last property plays an important role in the Paillier cryptosystem.

Real Case

We now consider the case of positive discriminant Δ_K still with a prime conductor q and consider $\Delta_q = q^2\Delta_K$. As we have seen, $C(\mathcal{O}_{\Delta_q})$ has small order in general and so is the kernel of $\bar\varphi_q$. We have seen that the order of this kernel is $(q - (\Delta_K/q))R_{\Delta_K}/R_{\Delta_q}$ and we still denote t the integer such that $R_{\Delta_q} = tR_{\Delta_K}$. As a consequence, this integer t is in general a large divisor of $(q - (\Delta_K/q))$.

For the cryptanalysis of the real version of NICE, we need to find the ideals of norm (q^2, kq) of $\mathcal{O}_{\Delta_q}$. We prove in the following theorem that these ideals are principal in general, and exhibit the expression of a generator in terms of q and powers of the fundamental unit ε_{Δ_K}.

Theorem (B−1 [CJLN09]). *Let Δ_K be a fundamental positive discriminant, $\Delta_q = \Delta_K q^2$ where q is an odd prime conductor. Let ε_{Δ_K} (resp. ε_{Δ_q}) be the fundamental unit of $\mathcal{O}_{\Delta_K}$ (resp. $\mathcal{O}_{\Delta_q}$) and t such that $\varepsilon_{\Delta_K}^t = \varepsilon_{\Delta_q}$. Then the principal ideals of $\mathcal{O}_{\Delta_q}$ generated by $q\varepsilon_{\Delta_K}^i$ correspond to quadratic forms $f_{k(i)} = q^2x^2 + k(i)qxy + (k(i)^2 - p)/4y^2$ with $i \in \{1, \dots, t-1\}$ and $k(i)$ is an integer defined modulo $2q$ computable from $\varepsilon_{\Delta_K}^i \mod q$.*

[1] In this proposition the notation are not the same: p plays the role of q and q is Δ_K/q.

One can find a detailed description of this relation between ideals of the maximal order and those of a non-maximal order using exclusively the language of binary quadratic forms in [Arn14].

I.4. Cryptography in non Maximal Orders

Imaginary Quadratic Fields and the NICE Family

In 1998, Hühnlein, Jacobson, Paulus and Takagi [HJPT98] made the first proposal of cryptography based on non maximal orders. This work has allowed the emergence of a new generation of cryptosystems based on class groups.

Their goal was to improve the efficiency of the adaptation of Elgamal in class groups. For this they propose to switch between the class group of the maximal order and the class group of a non maximal order, which can be done efficiently knowing the conductor. A traditional setup of Elgamal is done in $C(\mathcal{O}_{\Delta_q})$, $h = g^x$. A ciphertext is (g^r, mh^r) in $C(\mathcal{O}_{\Delta_q})$ where m is an ideal of small norm. To decrypt, the ciphertext is moved in the maximal order by applying $\bar{\varphi}_q$ with the trapdoor q and a traditional decryption is made to recover the message in $C(\mathcal{O}_{\Delta_K})$. Eventually, the message is lifted back in $C(\mathcal{O}_{\Delta_q})$. This variant can be seen as an Elgamal with a CRT decryption procedure: its advantage is that most of the decryption computation is done in $C(\mathcal{O}_{\Delta_K})$ and Δ_K can be chosen relatively small: big enough such the factorization of Δ_q is intractable, the discrete logarithm problem can be easy in $C(\mathcal{O}_{\Delta_K})$.

Soon after, *quadratic decryption time* was eventually reached with a new encryption scheme, called *NICE*, for *New Ideal Coset Encryption*, described in [HPT99,PT98,PT00]. This scheme uses a fundamental discriminant $\Delta_K = -p$ where p is prime and $\Delta_q = -pq^2$. The security of the scheme relies on the problem of factoring Δ_q. The key idea of NICE is to mask the plaintext message, encoded as an element of $C(\mathcal{O}_{\Delta_q})$ by a random element h^r where h is an element of the kernel of $\bar{\varphi}_q$ that is part of the public key. This mask h^r naturally disappears from the ciphertext when applying the map $\bar{\varphi}_q$.

In [HPT99], it is shown that the decryption time of NICE is comparably as fast as the encryption time of RSA with public exponent $e = 2^{16} + 1$ and an even better implementation is described by Hühnlein in [Hüh99].

A chosen-ciphertext attack against this cryptosystem and the scheme of [HJPT98] has been proposed by Jaumes and Joux in [JJ00]. An enhanced scheme has then been proposed by Buchmann, Sakurai and Takagi to obtain a chosen-ciphertext security in the random oracle model [BST02]. This scheme, based on REACT [OP01], is called NICE-X.

The ideas of NICE also led to the design of signature schemes in [HM00b,Hüh01] with an adaptation of Schnorr signatures. Again an element of the kernel of the switching between two class groups is published: this element is crucial for the efficiency of the signature generation. An undeniable signature scheme has been designed in [BPT04], and again, the public element of the kernel is needed for the design of an efficient scheme.

Parallel to the development of this NICE family of cryptosystem, Hühnlein, Meyer and Takagi have also built in [HMT98] Rabin and RSA analogues based on non-maximal imaginary quadratic orders. The advantage over the original systems is their natural immunity against low exponent attacks and some chosen-ciphertext attacks. For the adaptation of RSA, they proposed to work with *totally non maximal orders*, $\mathcal{O}_{\Delta_q}$ where the class group $\mathrm{C}(\mathcal{O}_{\Delta_K})$ is trivial. In this case, all the computation is done in the kernel of $\bar{\varphi}_q$. Moreover, using the isomorphism with the quotient group of equation I.2, one can actually compute in this last group. In [Hüh99], Hühnlein showed how to speed up the arithmetic in this group. But [HT99] also showed that using similar ideas one can reduce the discrete logarithm problem in these totally non-maximal imaginary quadratic orders to the discrete logarithm problem in finite fields. As a consequence, adaptation of schemes based on the discrete logarithm problem in such orders are only as secure as the original schemes in finite fields.

Adaptation of NICE in Real Quadratic Fields

As for the adaptation of the Diffie-Hellman key exchange, the adaptation of NICE to real quadratic fields is quite challenging. In class groups of real quadratic fields, there are many reduced elements in a class. This might be a problem during decryption in order to recover the right plaintext. Moreover we have seen that the order of the kernel of $\bar{\varphi}_q$ is small in general, so a direct adaptation NICE that masks the plaintext by a random element of this kernel of $\bar{\varphi}_q$ could be efficiently cryptanalysed by a simple exhaustive search.

Jacobson, Scheidler, and Weimer proposed in [JSW08] to hide a plaintext ideal of $\mathcal{O}_{\Delta_q}$ in the set of reduced ideals of its own equivalence class. (*e.g.*, by a multiplication by a random principal ideal). As a consequence, this set has to be large, which means that the regulator R_{Δ_q} must be large. Decryption is still done in $\mathcal{O}_{\Delta_K}$ by applying the map $\bar{\varphi}_q$ thanks to trapdoor q. To recover the message one has to perform an exhaustive search in a set of reduced ideals of $\mathcal{O}_{\Delta_K}$. This means that the regulator R_{Δ_K} must be small, which is very unlikely to happen if Δ_K has no special properties. The authors of [JSW08] have proposed to choose the prime Δ_K to be a so-called *Schinzel sleeper* which are known to have a regulator of the order $\log(\Delta_K)$ (see [CW05]).

As for the original NICE scheme, security relies on the intractability of factoring $\Delta_q = q^2\Delta_K$. Moreover, the public key consists of the sole discriminant Δ_q, without any additional element, whereas the public key of the original NICE scheme includes an element of the kernel of $\bar{\varphi}_q$. The main feature of the original NICE scheme is preserved: decryption is still done in quadratic time.

Chapter II

Cryptanalyses

In this chapter we give an overview of the cryptanalysis of the NICE family of cryptosystems done in [CL09] and [CJLN09]. We have seen in the previous chapter that both the NICE based systems in imaginary quadratic fields and the real version use a prime fundamental discriminant Δ_K and an order of prime conductor q, associated with the discriminant $\Delta_q = q^2\Delta_K$.

The public key of these systems contains the discriminant Δ_q and their security is based on the intractability of factoring this discriminant. We will show how to factor Δ_q in polynomial time based on hints. More precisely, our arithmetic hints can be either of the following two:

1. The hint is an ideal that represents a class of the kernel of $\bar{\varphi}_q$, the surjection from the class group of the non maximal order, $C(\mathscr{O}_{\Delta_q})$ to the class group of the maximal order $C(\mathscr{O}_{\Delta_K})$. In the imaginary systems based on NICE, such an ideal is disclosed by the public key.

2. The hint is the knowledge that the regulator R_{Δ_K} of the maximal order is unusually small. This is the case in the real version of NICE.

To recover the factorisation of Δ_q from those hints, we will look for ideals of norm q^2. As discuss in Section I.3, and more precisely in [CL09, Therorem A−2], these ideals form a system of representatives of $\ker\bar{\varphi}_q$ in the imaginary case. In the real case, these ideals are likely to be principals, and [CJLN09, Therorem B−1] give their generators in terms of the fundamental unit ε_{Δ_K} of $\mathscr{O}_{\Delta_K}$.

For the original NICE system, the first hint directly gives a reduced ideal equivalent to an ideal of norm q^2. For the real version, thanks to the second hint we will be able to do a polynomial exhaustive search for a particular such reduced ideal. In order to recover an ideal of norm q^2 from the reduced ideal, we have proposed an approach based on lifting in a sub order in [CL09]. This approach fails in the real version as it is based on the uniqueness of a reduced ideal in each class. However, we then have devised in [CJLN09] a method that works in both cases by using a variant of Coppersmith's root finding algorithm. In the following we describe these two approaches.

II.1. The Lifting Method

In this section we consider the imaginary case. Let $\mathfrak{h}$ be the reduced ideal given in the public key of NICE. The class of this ideal is in the kernel of the map $\bar{\varphi}_q$. Our goal is to recover the equivalent non reduced ideal of norm q^2 that belongs to the same class. As explored in [CL09] (see p. 55 of this document), trying to do a brute force ascent of the reduction algorithm from $\mathfrak{h}$ in order to recover the targeted ideal is infeasible for the parameters proposed in NICE. Therefore we have to establish another strategy.

Going up in the reduction algorithm is hard, but going down is easy. Suppose that the ideals of norm q^2 are actually reduced, it would then be easy to find them. Of course this is not the case in NICE, but if we consider an order of $\mathscr{O}_{\Delta_{qr}} \subset \mathscr{O}_{\Delta_q}$ of discriminant $\Delta_{qr} = r^2\Delta_q$ with a large enough integer r this would be the case. More precisely, with $r > 2q/\sqrt{|\Delta_K|}$, ideals of norm q^2 are reduced in $\mathscr{O}_{\Delta_{qr}}$.

As a consequence, we need a way to lift the class of $\mathfrak{h}$ in $C(\mathscr{O}_{qr})$ in a way that preserve the fact that $\mathfrak{h}$ is equivalent to an ideal of norm q^2. As r is prime to q, Chinese remaindering gives that the kernel of the map $\bar{\varphi}_{qr}$ from $C(\mathscr{O}_{qr})$ to $C(\mathscr{O}_{\Delta_K})$ is isomorphic to the product of two groups of the same form than in eq. I.2. One is defined modulo q and is isomorphic to $\ker\bar{\varphi}_q$ and the other one modulo r with known order $\phi_{\Delta_K}(r)$. We prove in [CL09, Lemma A−2] that the elements of $\ker\bar{\varphi}_{qr}$ that corresponds to elements that are trivial modulo r and non trivial modulo q, are classes with reduced element an ideal of norm q^2.

We thus only need to lift the class of $\mathfrak{h}$ to such elements of $\bar{\varphi}_{qr}$. This is done by computing

$$[\mathfrak{h} \cap \mathscr{O}_{qr}]^{\phi_{\Delta_K}(r)}.$$

which gives a reduced ideal of norm q^2 and the factorisation of Δ_q.

Intuitively, the exponentiation to the power $\phi_{\Delta_K}(r)$ makes trivial the component modulo r (cf. [CL09, Theorem A−3] for a detailed proof). Some technical details have to be considered. For example to lift the ideal, its norm must be prime to r. Moreover, this computation gives the trivial class if the order of $\mathfrak{h}$ divides $\phi_{\Delta_K}(r)$. By choosing first r in order to deal with these details, and then performing the lifted exponentiation to the power $\phi_{\Delta_K}(r)$ this eventually gives [CL09, Algorithm A.3]. The following corollary summarises the result:

Corollary (A−1 [CL09] adapted)**.** *There exists an algorithm that totally breaks the NICE family of cryptosystems in cubic time in the security parameter.*

In practice, the main computation is the exponentiation and performing the cryptanalysis on a cryptographic example takes less than a second on a standard PC.

II.2. A Method *à la* Coppersmith

We describe here a method that works both in the imaginary and real cases. This method does not use the uniqueness of a reduced ideal in a class. We still consider a reduced ideal $\mathfrak{h}$ of $\mathcal{O}_{\Delta_q}$ equivalent to a non reduced ideal of norm q^2. More precisely we suppose that $\mathfrak{h}$ is the output of the reduction algorithm applied to an ideal of norm q^2. This ideal is the hint that will allow us to factor the discriminant $\Delta_q = \Delta_K q^2$ in polynomial time. In the imaginary case, this ideal is still given in the public key of NICE. In the real case, we will see later how to do an exhaustive search for such an ideal $\mathfrak{h}$ sufficiently "close" to the ideal of norm q^2.

There is a well known connection between the ideal class group of an order of discriminant Δ and the set of binary quadratic forms of discriminant Δ modulo the action of $\mathrm{SL}_2(\mathbf{Z})$. We have seen on Chapter I that in practice computation in the ideal class group is done with algorithms designed for binary quadratic forms. Moreover, some properties of the ideal class group have often more natural proofs by translating them in terms of binary quadratic forms and *vice versa*. This fact makes the class group and incredibly rich structure. Our problem here, translated in terms of quadratic forms is quite standard. Let denote $h = ax^2 + bxy + cy^2$ the form of discriminant Δ_q associated to the ideal $\mathfrak{h}$. We know that h is the reduction of a form whose first coefficient is q^2. As a result q^2 is represented by the form h: there exists integers x_0, y_0 such that

$$h(x_0, y_0) = ax_0^2 + bx_0y_0 + cy_0^2 = q^2.$$

If we can find this representation of q^2 by h then we can factor Δ_q.

Moreover x_0, y_0 are relatively small compared to $|\Delta_q|$ and q^2 is an unknown factor of Δ_q. As a result, our problem can be solved with a variant of a standard tool in cryptanalysis: the Coppersmith method for finding small roots of polynomials modulo an integer N (cf. [Cop97]). Here this is the so-called *gcd variant*, where we try to find small roots modulo some unknown factor of N. The usual technique for this kind of problems in two variables is only heuristic. Fortunately, because our polynomial is homogeneous, we can design a variant that is quite similar to the one-variable standard Coppersmith method. This variant was independently developed by Bernstein, in [Ber11] in the different context of Goppa codes decoding. The result is given in following theorem. Some extensions can also be found in [LZPL15].

Theorem (B-2 [CJLN09]). *Let $f(x,y) \in \mathbf{Z}[x,y]$ be a homogeneous polynomial of degree δ satisfying $f(x,0) = x^\delta$, N be a non-zero integer and α be a rational number in $[0,1]$, then one can retrieve in polynomial time in $\log \mathrm{N}$, δ and the bit-size of α, all the rationals x_0/y_0, where x_0 and y_0 are integers such that $\gcd(f(x_0,y_0),\mathrm{N}) \geq \mathrm{N}^\alpha$ and $|x_0|, |y_0| \leq \mathrm{N}^{\alpha^2/(2\delta)}$.*

For our cryptanalysis, $\delta = 2$, $\mathrm{N} = |\Delta_q| = pq^2$ with p and q of the same size, *i.e.*, $\alpha = 2/3$, and we can asymptotically get roots up to N^β with $\beta = \frac{1}{9}$. The proof of this theorem is constructive, as all Coppersmith variants, with the definition of an explicit lattice that has to be reduced by the LLL algorithm. More concretely, if we take a lattice of dimension 13 in the proof, we get $\beta \approx \frac{1}{10.63}$.

In practice, this is sufficient for the cryptanalysis of both versions of NICE as experimentally, the roots are smaller than $|\Delta_q|^{1/11.7}$. As a result, for the imaginary version, it is sufficient to apply this variant of Coppersmith method to the form h obtained in the public key.

For the adaptation of NICE in real quadratic fields, we have still to show how to build the form h as the public key contains only Δ_q. This form must be the reduction of a form whose first coefficient is q^2 (in order to have small roots). For this we use Theorem B−1 of [CJLN09] that we saw in Chapter 1. This result tells us that in real NICE, the forms

$$f_{k(i)} = q^2x^2 + k(i)qxy + (k(i)^2 - p)/4y^2$$

are likely to be principal, and their corresponding generators are $q\varepsilon_{\Delta_K}^i$ for some i where $\varepsilon_{\Delta_K} \in \mathscr{O}_{\Delta_K}$ is the fundamental unit. We denote $\hat{f}_{k(i)}$ the reduction of the form $f_{k(i)}$. These forms will be in the cycle of reduced principal form. Moreover, experimentally there are well distributed in this cycle, roughly every R_{Δ_K} steps (see Fig. II.1). Some justifications of this can be obtained in terms of Shanks distance: the distance between two non reduced forms in the infrastructure $f_{k(i)}$ and $f_{k(i+1)}$ is the logarithm of the quotient of their respective generators, *i.e*, R_{Δ_K}. The reduction from $f_{k(i)}$ to $\hat{f}_{k(i)}$ usually has small impact on the distances, so we expect that the distance between $\hat{f}_{k(i)}$ and $\hat{f}_{k(i+1)}$ is still close to R_{Δ_K}, which is roughly the number of reduction steps between them.

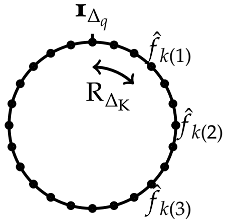


Figure II.1: Repartition of the forms $\hat{f}_{k(i)}$ along the principal cycle

The algorithm to attack real NICE can be sketched as follows (for a detailed description see Fig. B.3 of [CJLN09]). We start form the principal form: the reduced form

$$x^2 + \lfloor\sqrt{\Delta}\rfloor xy + (\lfloor\sqrt{\Delta}\rfloor^2 - \Delta)/4y^2.$$

Then we take a walk around the principal cycle, by applying reduction steps. After each step, we try to represent q^2 with the current form with the Coppersmith homogeneous variant, if we fail, we continue.

We expect to reach the form $\hat{f}_{k(1)}$ after R_{Δ_K} steps which is linear in the security parameter for real NICE (this is our second hint). This is summarize as follows:

Result (B.1 [CJLN09] adapted)**.** *There exits an algorithm that heuristically recovers the secret key of REAL-NICE in polynomial time in the security parameter.*

In practice, for the recommended parameters of [JSW08], our cryptanalysis succeeds in a few tens of seconds on a standard PC.

Interestingly, this attack gives also a deterministic factoring algorithm of numbers of the form pq^2 in $\tilde{\mathrm{O}}(p^{1/2})$ as $\mathrm{R}_p \approx \sqrt{p}$ in the general case.

These cryptanalyses work well in practice but rely on several heuristics on the size of the roots to be found (which correspond to the max norm of the matrix of $\mathrm{SL}_2(\mathbf{Z})$ involved in the reduction) and on the distribution of the reduced form $\hat{f}_{k(i)}$ in the principal cycle in the real case. Bernard and Gama have extended our work in [BG10] in order to remove the heuristics and prove unconditionally the cryptanalyses. For this, they introduce a variant of the reduction algorithm to obtain better bounds on the norm of the reduction matrix. Moreover, they introduce a variant of the algorithm to find small roots of homogeneous polynomials in order to deal with unbalanced solutions.

Chapter III

Primitives

In this chapter, we give a presentation of the CL *linearly homomorphic encryption scheme* introduced in [CL15], which is homomorphic modulo a prime p. We also present some extensions of this work, proposed in [CLT18, CCL+19]. To instantiate this scheme we will use constructively the ideas of the cryptanalysis of NICE seen in the previous chapter.

In the first section we will give some (non exhaustive) background on linearly homomorphic encryption. In Section III.2, we will present the framework of a cyclic group with an easy DL subgroup, introduced in [CL15] and then, in Section III.3, we will expose new linearly homomorphic encryption schemes using this framework.

As seen in the introduction, with a linearly homomorphic encryption scheme, it is possible to compute linear combinations on ciphertexts without knowing the secret key. This property has many applications and we will see some of them in the next chapter, in the areas of advanced encryption schemes and secure two party computation.

III.1. Background on Linearly Homomorphic Encryption

The story of homomorphic encryption begins with the definition of probabilistic encryption and the scheme of Goldwasser and Micali ([GM84]), which is linearly homomorphic over $\mathbf{Z}/2\mathbf{Z}$. This scheme was extended by Benaloh in his PhD thesis [Ben88], then by Naccache and Stern in [NS98] and Okamoto and Uchiyama [OU98]. One of the most achieved system was designed by Paillier [Pai99]. It is linearly homomorphic over $\mathbf{Z}/N\mathbf{Z}$ where N is an RSA integer. Paillier's scheme has then been generalised by Damgård and Jurik [DJ01], allowing to encrypt larger messages. This family of practical linearly homomorphic schemes is still growing with the more recent work of Joye and Libert [JL13]. The security of these schemes is related to the problem of factoring an RSA integer N. More precisely their semantic security is based on distinguishing some powers, *e.g.*, squares of $(\mathbf{Z}/N\mathbf{Z})^\times$ for Goldwasser-Micali with the *Quadratic Residuosity* assumption, N–th powers of $(\mathbf{Z}/\mathrm{N}^2\mathbf{Z})^\times$ for Paillier with the *Decision Composite Residuosity* (DCR) assumption.

In order to design a linearly homomorphic encryption scheme related to the Discrete Logarithm problem (DL), a folklore solution consists in encoding the message in the exponent of an Elgamal encryption, *i.e.*, in encrypting m as $(g^r, h^r g^m)$ where g is a generator of a cyclic group $\mathrm{G} = \langle g \rangle$ and $h = g^x$ is the public key. Unfortunately, to decrypt, one has to recover m from g^m and as the DL problem in G must be intractable, m has to be small enough to ensure a fast decryption. As a result, only a logarithmic number of additions is possible. There have been some attempts to reach a fully linear homomorphy based on the DL problem. For example, a variant of Elgamal modulo p^2 ([CPP06]) or another variant using messages encoded as a small smooth number ([CC07]); but both solutions still have a partial homomorphy. The problem of designing a fully linearly homomorphic scheme based on the sole hardness of the DL problem is stated to be an open problem in [CPP06].

A solution was actually proposed by Bresson, Catalano and Pointcheval in [BCP03]. The idea is to use the same setting as Paillier, using the fact that in $(\mathbf{Z}/\mathrm{N}^2\mathbf{Z})^\times$, the DL problem is easy in the subgroup generated by $f = 1+\mathrm{N}$ (the kernel of the surjection from $(\mathbf{Z}/\mathrm{N}^2\mathbf{Z})^\times$ to $(\mathbf{Z}/\mathrm{N}\mathbf{Z})^\times$). However, their scheme is not only based on the DL problem but also on the factorisation problem. Moreover, like Paillier this scheme is homomorphic modulo N and as seen in the introduction, it will be interesting for application to have a scheme homomorphic modulo p for a prime p smaller than an RSA integer.

The scheme of [BCP03] can be generalised with the framework introduced in the next section.

III.2. Framework of a Cyclic Group with an Easy DL Subgroup

In [CL15, Definition C−1] we have proposed a definition of this framework, focusing on the DDH assumption. We then have refined this definition in [CLT18, Definition E−6] to introduce another assumption which generalises DCR. Some small technical modifications were then made in [CCL+19, Definition F−4] to fit the particularity of the targeted application. We here present informally the most general version of the definition of this framework.

Definition

This framework is defined as a pair of algorithms $(\mathsf{Gen}, \mathsf{Solve})$.

Gen: The Gen algorithm is a group generator which takes as input a security parameter λ and outputs a tuple

$$(p, \tilde{s}, g, f, g_p, \widehat{\mathrm{G}}, \mathrm{G}, \mathrm{F}, \mathrm{G}^p).$$

The elements of this tuple are defined as follows:

- *Groups*: $\widehat{\mathrm{G}}$ is a finite multiplicative abelian group, G is a cyclic subgroup of $\widehat{\mathrm{G}}$, F a subgroup of G and $\mathrm{G}^p = \{x^p, x \in \mathrm{G}\}$;

- *Orders*: p is the order of F. If we denote $s = |\mathrm{G}|/p$, then we require that $\gcd(p,s) = 1$. As a result, G^p has order s and G is the direct product of F and G^p: $\mathrm{G} = \mathrm{F} \times \mathrm{G}^p$;
- *Generators*: g, f, g_p are respective generators of G, F and G^p with $g = f \cdot g_p$;
- *Upper bound*: Only an upper bound $\tilde{s}$ of s is given as output;
- *Additional Properties*: It is required that one can efficiently recognise valid encodings of elements in $\widehat{\mathrm{G}}$.

Solve: The DL problem is required to be easy in F, which means that the Solve algorithm is a deterministic polynomial time algorithm that solves the discrete logarithm problem in F.

Remarks

Let us give some remarks on the integers p and s of this definition. It is crucial in the following that the precise order s of G^p is ***unknown*** and hard to compute from the output of Gen. Otherwise, the algorithmic assumptions that we will consider will be false (see for instance Remark C−1 of [CL15]). Moreover, in practice, we will choose the size of the groups in order that the best known algorithms to compute s (or a multiple of s) from the output of Gen takes 2^λ operations.

The fact that p and s are coprime is extensively used for reductions between hard problems and for cryptographic proofs. In general, in our applications with class groups, p will be a prime ($\mathbf{Z}/p\mathbf{Z}$ will be the plaintext space). But p can also be an RSA integer if we want to encompass constructions based on Paillier, like [BCP03]. For cryptographic proofs, we sometimes need that non zero elements of $\mathbf{Z}/p\mathbf{Z}$ are invertible, so it will hold unconditionally if p is prime and under the factoring assumption if is an RSA integer. Moreover, we often need that p has at least λ bits in order to have $1/p$ negligible, and this would be mandatory for our instantiation based on class groups.

The upper bound $\tilde{s}$ on s will be used for sampling elements of the groups of unknown order. Let us consider the example of sampling in G^p at distance less than $2^{-\lambda}$ to the uniform distribution. This can be done generically by computer g_p^x where x is sampled with a uniform distribution in $\{0, \dots, 2^{\lambda-2}\tilde{s}\}$. A bit more efficiently (in terms of size of the exponents), one can sample x with a Discrete Gaussian distribution over $\mathbf{Z}$ of standard deviation $\sqrt{\lambda}\tilde{s}$. See Lemma E−4 of [CLT18] for details.

The group $\widehat{\mathrm{G}}$ will be mainly used in applications where we consider active adversaries, for example malicious users in 2 party protocols.

Examples

The encryption scheme of [BCP03] which relies on the arithmetic of Paillier is an instance of this definition (and was actually an inspiration for this framework). Let $\mathrm{N} = (2p' + 1)(2q' + 1)$ be an RSA integer, product of two safe primes. The group $\widehat{\mathrm{G}}$ is the subgroup of $(\mathbf{Z}/\mathrm{N}^2\mathbf{Z})^\times$ of elements whose Jacobi symbol is 1 and the group G is the cyclic group of quadratic residues. One has $|G| = Ns$ with $s = p'q'$. The subgroup F of N of G is generated by $1 + N$, and since $(1 + N)^k \equiv 1 + kN \pmod{N^2}$, the DL problem is easy in F. The order of G^N is $s = p'q'$ and an upper bound of s is given by $N/4$.

Our main instantiation will be with *class groups of imaginary quadratic fields*. We sketch in the following the instantiation given in Subsection C.3.1 of [CL15] (see in particular Fig. C.2 there). We introduce some small modifications (mostly rewriting) to fit the evolution of the framework.

p **and** $\widehat{G}$**:** Let p be a λ bit prime, and $q > 4p$ another prime and consider the fundamental discriminant $\Delta_K = -pq$ (we moreover require that $pq \equiv -1 \pmod 4$ and that $(p/q) = -1$ in order to have a 2–Sylow subgroup of $C(\mathcal{O}_{\Delta_K})$ of order 2). The size of q is chosen such that computing the class number $h(\Delta_K)$ (and computing discrete logarithms in $C(\mathcal{O}_{\Delta_K})$) takes 2^λ time. We then consider the suborder of conductor p, $\mathcal{O}_{\Delta_p}$, of discriminant $\Delta_p = -p^2\Delta_K$. Then,

$$\widehat{G} \text{ is the subgroup of squares of } C(\mathcal{O}_{\Delta_p}).$$

One can efficiently check that a given element represents an element of $\widehat{G}$ (cf. end of Subsection C.B.4 of [CL15]). The other (cyclic) groups are constructed by giving a generator.

g_p**:** As p has at least λ bits, following the Cohen-Lenstra heuristics, the probability that p divides $h(\Delta_K)$ is negligeable. Therefore, we can assume that $\gcd(p, h(\Delta_K)) = 1$. To construct g_p we use the setup of the implementations of Elgamal in the class group of a maximal order (cf. Section I.2). We first construct a random square $[\mathfrak{r}]$ of $C(\mathcal{O}_{\Delta_K})$ and we assume that this element will be of order s, an integer of the same order of magnitude than the odd part, $h(\Delta_K)/2$. Then we lift this element with the method used in the cryptanalysis of NICE: we compute

$$g_p = [\mathfrak{r} \cap \mathcal{O}_{\Delta_p}]^p.$$

As $p|\Delta_K$, the order of the kernel of the surjection $\bar{\varphi}_p$ from $C(\mathcal{O}_{\Delta_p})$ to $C(\mathcal{O}_{\Delta_K})$, $\phi_{\Delta_K}(p) = p$ and this map is a well-defined morphism. It is injective as $\gcd(p, h(\Delta_K)) = 1$. So g_p is a p–th power of $\widehat{G}$, of order s.

f**,** F**,** g**,** G **and** G^p**:** We then define $f \in \widehat{G}$ as

$$f = [(p^2, p)].$$

From what we saw in Section I.3 (p. 13 or more precisely in [CL15], Proposition C−1), the discrete logarithm is easy in $F = \langle f \rangle$, which implicitly defines the Solve algorithm. Moreover, $F = \ker \bar{\varphi}_p$ so F as order p. We then set $g = g_p f$ and $G = \langle g \rangle$. Consequently, g_p is a generator of G^p.

$\tilde{s}$: An upper bound $\tilde{s}$ on s is given by an upper bound of $h(\Delta_K)$. For this, one can use the fact that $h(\Delta_K) < \frac{1}{\pi}\log|\Delta_K|\sqrt{|\Delta_K|}$, or obtain a slightly better bound from the analytic class number formula, *cf.* [McC89].

A notable difference between this instantiation and the scheme of [BCP03] is that the prime p is almost generated independently of the rest of the output. So it can be an input of the Gen algorithm and we can generate p to match a targeted application, like the group order of ECDSA elliptic curves in [CCL+19]. Note that the requirement $q > 4p$ is needed in order that the ideals of norm p^2 are reduced. As a result they are found in the Solve algorithm. In [CL15, Subsection C.4.1], it is shown how to drop this requirement and how to have no restriction on the size of q. In this case, the ideals of norm p^2 are non reduced but can be found using constructively the idea of the lifting method of the cryptanalysis of NICE (cf. Section II.1). This enable to use large message spaces, for example a $\mathbf{Z}/p\mathbf{Z}$ with a p of 2048 bits as in Paillier, without using the prime q that will enlarge the size of the discriminant Δ_K while it is already large enough for security. As a result one considers $\Delta_K = -p$ and $\Delta_p = -p^3$. This variant is used for the application for encryption switching protocols in [CIL17].

Another difference of this instantiation in class groups compared to Paillier based schemes, is that the order s is unknown to anyone including Gen where as it was known for Paillier in order to build the RSA integer N. As a result, *we do not need a trusted party* to execute Gen contrary to variants of Paillier.

Assumptions

We give the assumptions on which relies the semantic security of the linearly homomorphic encryption schemes that will be defined in the next section. These assumptions are made in the framework of a cyclic group G with an easy DL subgroup F. We thus consider the output of the group generator Gen.

DDH-f: Originally, we used the traditional DDH assumption in G in [CL15]. However, a more precise analysis was done in [CLT18] where we remark that the IND − CPA security of the original CL scheme is actually equivalent to a weaker assumption named DDH-f. Furthermore, the DDH-f assumption provides clearer proofs.

Denoting $\mathscr{D}$ a distribution statistically close to the uniform distribution modulo ps, and $u \hookleftarrow \mathbf{Z}/p\mathbf{Z}$ a uniform sampling in $\mathbf{Z}/p\mathbf{Z}$, the DDH-f assumption states that it is hard to distinguish the two following distributions

$$\{(g^x, g^y, g^{xy}), x, y \hookleftarrow \mathscr{D}\},$$

i.e., Diffie-Hellman triplets in G, and

$$\{(g^x, g^y, g^{xy} f^u), x, y \hookleftarrow \mathscr{D}, u \hookleftarrow \mathbf{Z}/p\mathbf{Z}\}$$

.

A more precise definition is given in [CLT18], Definition E − 8. A similar assumption of DDH within a subgroup is defined by Hemenway and Ostrovsky in [HO12].

HSM − CL: In [CLT18], we have defined another assumption. In our framework, G is the direct product $F \times G^p$. The HSM − CL assumption (for *hard subgroup membership*) is to distinguish the elements of G^p in G (see Definition E−7 in [CLT18] for a precise statement). With the instantiation of our framework based on Paillier, this gives the DCR assumption.

Moreover, we prove in the general case, in [CLT18, Theorem E−4], that this HSM−CL assumption implies the DDH-f assumption. However in the next section, we will see that this assumption allows to obtain a more efficient scheme, with extra nice properties for applications.

III.3. Linearly Homomorphic Encryption Schemes

Generic Constructions

We use the framework of a cyclic group with an easy DL subgroup of the previous section to design new linearly homomorphic encryption schemes. We thus consider the output of the group generator Gen and the Solve algorithm.

CL: This scheme, proposed in [CL15], is an Elgamal in the exponent, where the plaintext message is encoded in the subgroup F where the DL problem is easy. As a result, during decryption we can recover the result whatever its size, and we get fully linear homomorphy modulo p. The scheme is depicted in Fig. III.1 where $\mathscr{D}$ is a distribution close to the uniform modulo ps (implemented from the upper bound $\tilde{s}$ of s). We ignore the group $\widehat{G}$, the subgroup G^p and its generator which are useless here.

Addition over two ciphertexts $c = (c_1, c_2)$, $c' = (c'_1, c'_2)$ is done by multiplication component wise: $(c_1c'_1, c_2c'_2)$. Multiplication on a ciphertext c by a scalar $a \in \mathbf{Z}/p\mathbf{Z}$, is done by exponentiation: (c_1^a, c_2^a). Re-randomisation of a ciphertext can be done by an addition with a ciphertext of 0.

This scheme is IND − CPA under the DDH-f assumption.

HSM **variant of** CL: This scheme was proposed in [CLT18]. It is IND−CPA under the HSM − CL assumption. The main idea consists in using the subgroup decomposition problem to hide the message in the second component of the ciphertext. This scheme is actually a generalisation of another variant of Paillier, proposed by Camenisch and Shoup in [CS03].

Compared with the original CL, one gains in efficiency, as exponentiation are a bit shorter: we now use a distribution $\mathscr{D}_p$ close to the uniform modulo s. The scheme is depicted in Fig. III.2, still in the framework of a group with an easy DL subgroup where we ignore the useless elements. The scheme is still linearly homomorphic modulo p: addition on ciphertexts is still performed by component wise multiplication and scalar multiplication by exponentiation of both components.

Another interesting property of this variant is that it results from a *smooth Hash Proof System*, *à la* Cramer-Shoup [CS02] as shown in Subsection F.4.2 of [CCL+19].

```
Algorithm Keygen($1^\lambda$)
    1. $(p, \tilde{s}, g, f, G, F) \leftarrow \mathsf{Gen}(1^\lambda)$
    2. Pick $x \hookleftarrow \mathscr{D}$ and set $h = g^x$
    3. Set $pk = (p, \tilde{s}, g, h, f)$ and $sk = x$.
    4. Return $(pk, sk)$
```

```
Algorithm Decrypt($1^\lambda, pk, sk, (c_1, c_2)$)
    1. Compute $M = c_2/c_1^x$
    2. $m \leftarrow \mathsf{Solve}(p, g, f, G, F, M)$
    3. Return $m$
```

```
Algorithm Encrypt($1^\lambda, pk, m$)
    1. Pick $r \hookleftarrow \mathscr{D}$
    2. Compute $c_1 = g^r$
    3. Compute $c_2 = f^m h^r$
    4. Return $(c_1, c_2)$
```

Figure III.1: The CL linearly homomorphic encryption scheme

```
Algorithm Keygen($1^\lambda$)
    1. $(p, \tilde{s}, f, g_p, G, F, G^p) \leftarrow \mathsf{Gen}(1^\lambda)$
    2. Pick $x \hookleftarrow \mathscr{D}_p$ and $h = g_p^x$
    3. Set $pk = (\tilde{s}, g_p, f, p, h)$
    4. Set $sk = x$
    5. Return $(pk, sk)$
```

```
Algorithm Encrypt($pk, m$)
    1. Pick $r \hookleftarrow \mathscr{D}_p$
    2. Return $(g_p^r, f^m h^r)$
```

```
Algorithm Decrypt($sk, (c_1, c_2)$)
    1. Compute $M = c_2/c_1^x$
    2. Return $\mathsf{Solve}(M)$
```

Figure III.2: HSM variant of CL

These constructions allow the simulation in a cryptographic proof to know the secret key, which provides more flexibility. This is the basis of the construction of inner product functional encryption schemes in [CLT18] and of the fact that we can have a simulation-based security proof without resorting to non-standard interactive assumptions in [CCL+19].

It is possible to build a smooth Hash Proof System attached to the subset membership problem of the DDH-f assumption. However, the resulting scheme is less efficient with an extra element in the ciphertext and larger exponents (cf. the modified CL scheme *à la* CS lite of Sub Fig. E.2(b) of [CLT18].

Faster variant of CL**:** In [CL15, Subsection C.4.2], a faster variant of the original CL cryptosystem is proposed. The idea is to perform the exponentiations of the key generation and of the encryption algorithm (the DH triplet (g^x, g^r, h^r)), in the quotient group G/F and to lift h^r in G^p in order to hide f^m. For example, in class groups, this corresponds to the lift used to create g_p. This results in smaller exponentiations in a smaller group: for class groups, G/F is isomorphic to a subgroup of $Cl(\mathcal{O}_{\Delta_K})$. As a matter of fact, this scheme is very similar to the HSM variant of CL, and one can prove that it is also IND − CPA under the HSM − CL assumption (where as it was proven secure under an *ad-hoc* assumption in [CL15]). Note that this last scheme does not result from an hash proof system.

Instantiations

We have seen that with the Gen and Solve algorithms derived from Paillier, one recovers the linearly homomorphic schemes modulo an RSA integer N of [BCP03] and [CS03] as instantiations of the generic constructions.

With the instantiation of Gen and Solve with class groups of imaginary quadratic fields, we get *new linearly homomorphic schemes modulo a prime p*. We give in Fig. III.3 a comparison of the result of an implementation of Paillier and of the HSM − CL scheme. The timings are in ms with an implementation using the Pari C Library ([PAR18]), running on a single core of an Intel(R) Core(TM) i7-7700 CPU @ 3.60GHz. The sizes are in bits. For HSM − CL we use for message space $\mathbf{Z}/p\mathbf{Z}$ where the bit size of p matches the security level.

One can see that in terms of sizes, the HSM − CL scheme is always more compact, at all security levels. This comes from the fact that we can use smaller parameters as problems in class groups are asymptotically more difficult than factoring as we saw in Subsection I.1. However, even if the arithmetic in class groups is efficient, it is still more costly than in quotients of $\mathbf{Z}$. As a result, there is a gain with class groups in terms of timing only at the 192 and 256 bits security levels, keeping in mind that for decryption we use one exponentiation as Paillier, but for encryption we need two (that can be parallelised) instead of one. However, as we will see in the next chapter, we will have additional gains in using CL schemes in applications thanks to the fact that we have homomorphy modulo a prime instead of an RSA integer.

Sec. Param.	N	Expo in $\mathbf{Z}/N^2\mathbf{Z}$	Paillier ciphertext
112	2048	**7**	4096
128	3072	**22**	6144
192	7680	214	15360
256	15360	1196	30720

Sec. Param.	Δ_K	Expo in $Cl(\mathcal{O}_{\Delta_p})$	HSM − CL ciphertext
112	1348	32	**3144**
128	1827	55	**4166**
192	3598	**212**	**7964**
256	5971	**623**	**12966**

Figure III.3: Comparison of Paillier and HSM − CL

Extensions

In [DJS19], Das, Jacobson and Scheidler present some extensions of the instantiation of the CL scheme in class groups of imaginary quadratic fields, allowing to have a message space with a non prime order (multiple prime divisors with or without prime powers). For this, they show how to adapt the Solve algorithm by using Chinese Remaindering. Moreover they report a C/C++ implementation. They conduct a large range of experiments, with several variants to determine the most efficient solution with respect to a security level and a message space size.

Chapter IV

Protocols

In this chapter, we focus on the use of the CL linearly homomorphic encryption scheme as a building block to design advanced cryptographic protocols. Some properties of CL are appealing for applications. For example, the flexibility on the size of the message space of the CL scheme has been used to provide anonymity with the design of mix network in [CDJ+16] and to construct anonymous and secure aggregation schemes in [WWD18].

The fact that CL is homomorphic modulo a prime can be used to build designated-verifier non-interactive zero-knowledge proofs of knowledge over prime-order abelian groups as stated in [CC18]. In [LMS18], the CL scheme is used in the context of homomorphic secret sharing in replacement of Elgamal in the exponent in order to overcome the limitation to small messages.

In the following, we present some other applications of the CL linearly homomorphic encryption scheme which also show the usefulness of the existence of a linearly homomorphic encryption scheme over $\mathbf{Z}/p\mathbf{Z}$ where p is a prime.

We will first describe some results obtained in the context of secure two party computation: in [CIL17] with encryption switching protocols and recently in [CCL+19] for two-party ECDSA signing. We will show that the CL scheme helps in order to simplify protocols, to reduce the number of rounds and the number of bits exchanged by the parties.

We will then show that the CL scheme can be used as a building block to obtain more advanced encryption schemes with results in the area of inner product functional encryption obtained in [CLT18].

IV.1. Secure Two Party Computation

In secure two party computation, two parties can compute the image of a pre-agreed function of their private inputs through interactive cryptographic protocols. At the end of the interaction, anything that a party has learned from the protocol could have been deduced from its public and secret inputs and outputs. This is of course a special case of secure multi-party computation (MPC).

This area of research has emerged in the 80's with the work of Yao. Initially considered as a theoretical subject due to overly inefficient protocols, MPC has nowadays reached a reasonable complexity and has become relevant for practical purposes, especially in the two party case. MPC protocols are built from various tools (*e.g.*, oblivious transfer, secret sharing, garbled circuits...) and linearly homomorphic encryption is one of them.

Encryption Switching Protocols

Suppose that Alice has a private input a and Bob a private input b. Then both of them encrypt its private input with a linearly homomorphic scheme with a common public key, and send the ciphertext to the other party. Then Alice and Bob can compute a linear combination of a and b over ciphertexts, without interaction. At the end of the protocol, they can decrypt the result with a two party decryption protocol.

They have been proposal to extend this protocol to manage multiplications of private inputs ([CDN01]) but this requires interaction for each multiplication. Couteau, Peters and Pointcheval, in [CPP16], have introduced the notion *encryption switching protocols* (ESP). This mechanism uses a pair of encryption schemes, one is additively homomorphic and the other one multiplicatively homomorphic with a common plaintext space. Furthermore, there exists switching two party protocols to securely convert ciphertexts between the two schemes. Combined with the idea above, it allows to evaluate a circuit in two party efficiently gathering the additive and multiplicative gates separately.

Couteau *et al.* have proposed an instantiation using Paillier for the additively homomorphic scheme, which is homomorphic over $\mathbf{Z}/N\mathbf{Z}$, with N an RSA integer. For the multiplicative scheme they had to design a clever variant of Elgamal over $(\mathbf{Z}/N\mathbf{Z})^*$. Indeed as Elgamal is secure only in the subgroup of $(\mathbf{Z}/N\mathbf{Z})^\times$ of elements of Jacobi symbol +1, they need a careful encoding of the group $(\mathbf{Z}/N\mathbf{Z})^*$ in order to enlarge the message space, taking care of elements that could leak the factorisation of N. The downside is that their variant of Elgamal does not support a simple 2-party decryption (a Paillier layer has to be added to Elgamal in order to be able to simulate this protocol). As a result, their switching protocols are intricate and specific to their construction.

In [CIL17], we revisit this work together with L. Imbert and F. Laguillaumie. We propose a generic approach of an ESP, inspired by Couteau *et al.*'s solid basis. We suppose that both homomorphic schemes support a 2-party decryption protocol in one round. This is a common property for encryption schemes whose decryption algorithms consist in the computation of an exponentiation. This gives a conceptually simpler ESP (see Section D.4 of [CIL17]).

We then provide an efficient instantiation of our generic protocol over the field $\mathbf{Z}/p\mathbf{Z}$. For the additively homomorphic scheme, we use the CL scheme of Fig. III.1, using the variant of the $(\mathsf{Gen}, \mathsf{Solve})$ algorithms suited for large message space discussed in Section III.2 (see also Section C.4 of [CL15]): we work with a class group of an order of discriminant $-p^3$. Indeed, we use Elgamal for the multiplicatively homomorphic scheme, so we need that the discrete logarithm problem is hard in $\mathbf{Z}/p\mathbf{Z}$, so p will have at least 2048 bits. We devise a 2-party decryption protocol for CL in one round. In CL we only need to share one exponentiation, and the secret exponent has to be shared over the integers, as the group order is unknown (see Fig. D.9 of [CIL17]).

Then, for the multiplicative scheme, we need to extend Elgamal to $(\mathbf{Z}/p\mathbf{Z})^*$. We choose p to be a safe prime, so we can safely use Elgamal in the subgroup of quadratic residues modulo p. To extend Elgamal, we thus need to only encrypt an extra bit with the Goldwasser-Micali encryption scheme (cf. Fig. D.10 of [CIL17]). The situation is simpler than the extension of Elgamal to $(\mathbf{Z}/N\mathbf{Z})^*$ as we do not have to hide the factorisation of N. As a result, we can obtain a 2-party decryption in one round for this multiplicative scheme (Fig. D.11 of [CIL17]).

The instantiation of our generic construction with these two schemes reduces the number of rounds as well as the number of bits exchanged by the parties (cf. Table D.1 of [CIL17]). In this application, the fact that CL is homomorphic modulo a prime p has been crucial to simplify proofs and protocols compared to using Paillier which is homomorphic modulo an RSA integer N.

Two party ECDSA signing

With D. Catalano, F. Laguillaumie, F. Savasta and I. Tucker we recently obtained in [CCL+19] another result in 2-party computation where we revisit a protocol of Lindell [Lin17] for 2 party ECDSA signing. In this protocol, a particular case of threshold cryptography, two players P_1 and P_2 have shares of the signing key x and must actively participate in order to generate a signature. A single player can do nothing.

There have been intense research efforts in threshold cryptography in the 90's and recent years have seen renewed interest in the field for several reasons. First a number of start-up companies are using this technology to protect keys in real life applications [Ser, Unb, Sep]. Moreover, Bitcoin and other cryptocurrencies – for which security breaches can result in concrete financial losses – use ECDSA as underlying digital signature scheme. While multisignature-based countermeasures are built-in to Bitcoin, they offer less flexibility and introduce anonymity and scalability issues (see [GGN16]).

They have been recent proposals to construct 2 party variants of ECDSA signatures (*e.g.*, [GGN16, Lin17, DKLs18]) and constructing efficient protocols proved to be much harder than for other signature schemes. The main reason comes from the following computation in the ECDSA protocol:

$$s \leftarrow k^{-1}(\mathrm{H}(m) + rx) \bmod q,$$

Starting from [MR04a] two party ECDSA signature protocols started adopting a multiplicative sharing both for x (the secret key) and k (the randomness used in the signing phase). Players start holding shares x_1, x_2 such that $x = x_1 x_2$ and k_1, k_2 such that $k = k_1 k_2$. This immediately allows to get shares of the inverse of k as clearly

$(k_1)^{-1}(k_2)^{-1} = k^{-1} \bmod q$. As a final ingredient, the parties use Paillier's homomorphic encryption to get the linear part: For instance, player P_1 computes the ciphertexts $\mathsf{Enc}(k_1^{-1}H(m))$ and $\mathsf{Enc}(k_1^{-1}rx_1)$. P_2 can then complete the computation, using the homomorphic properties of the scheme to get the linear combination $(k_2^{-1}, k_2^{-1}x_2)$ over ciphertexts, which gives an encryption of

$$k_2^{-1}(k_1^{-1}H(m)) + k_2^{-1}x_2(k_1^{-1}x_1r) = k^{-1}(H(m) + rx) \bmod q$$

However, there are some technicalities induced from the fact that Paillier's plaintexts space is $\mathbf{Z}/N\mathbf{Z}$ whereas ECDSA signatures live in $\mathbf{Z}/q\mathbf{Z}$ (q is prime). Thus to avoid inconsistencies one needs to make sure that N is taken large enough so that no wraps around occur during the whole signature generation process. This also means that to handle malicious users, the players must prove in zero knowledge that encryptions are in the right range which is in general inefficient. Another issue with Paillier encryption is that to prove the indistinguishability of an adversary's view in real and simulated executions, one wants to use the semantic security of Paillier scheme, which means that the simulator must not know the secret key. Lindell [Lin17] proposes two alternative proofs to overcome this. That results in a loss in the proof of security or the use of a new (interactive) non standard assumption.

We have revisited the protocol of Lindell. We first show that one can have a generic construction for two-party ECDSA signing that can be seen as a generalisation of Lindell's scheme to the setting of homomorphic hash proof systems (HPS) (see Fig. F.5 of [CCL+19]). This protocol resolves an issue of previous schemes as it allows for a proof of security that is both tight and does not require artificial interactive assumptions when proving simulation security (cf. Theorem F−1 of [CCL+19]). This comes from the fact that with an HPS, it is possible for the simulation of the proof to know the secret key. Then, the indistinguishability of an adversary's view in real and simulated executions comes from properties of the HPS (smoothness and hardness of the attached hard subset membership problem).

Secondly, we have instantiated this generic construction with the HSM variant of CL described in Fig. III.2. As seen in the previous chapter, this linearly encryption scheme results from an HPS. Moreover, the scheme is homomorphic modulo a prime that can be chosen equal to the prime q of the ECDSA standard independently of the security parameter. This provides a cleaner approach than with Paillier and we do not need costly zero knowledge range proofs. However difficulties arise from the fact that the order of the class group used in CL is unknown and that we cannot assume that a ciphertext is valid (where as for Paillier, every element of $(\mathbf{Z}/N^2\mathbf{Z})^\times$ is a valid ciphertext). These issues occur when proving that a ciphertext encrypts x_1, where $x_1 \in \mathbf{Z}/q\mathbf{Z}$ is implicitly defined from an elliptic curve point $Q_1 = x_1G$ where G is the generator of the group of points of order q of the elliptic curve used in ECDSA. This is addressed by using a Schnorr-like proof with binary challenge. This proof has to be repeated to get negligible soundness error, but this is done only once for key generation.

A C implementation of our protocols shows that for high security levels, our solution becomes faster (in terms of key generation from a 192-bits security level and for both key generation and signing for a 256-bits security level) than the solution of Lindell. In terms of communications, our solution significantly reduces the number of bits exchanged, and the number of rounds for key generation (cf. Fig. F.9 of [CCL⁺19]).

IV.2. Inner Product Functional Encryption

Functional encryption (FE) allows to finely control the information revealed to recipients from a given ciphertext. Specifically, FE allows for a receiver to recover a function $f(y)$ of the encrypted message y, without learning anything else about y. Though constructions for general FE have been put forth, these schemes are far from practical, or rely on non-standard assumptions. The problem thus arose of building efficient FE schemes for restricted classes of functions.

Inner product functional encryption (IPFE) restricts the class to linear functions. In an IPFE, a plaintext is a vector y and secret keys allow to recover linear combinations x of its components, *i.e.*, $\langle x, y\rangle$, given a ciphertext for y. Among other uses, linear functions allow for the computation of weighted averages and sums which are of use for statistical analysis on encrypted data.

Moreover IPFE allows us to construct other cryptographic primitives. Agrawal, Bhattacherjee, Phan, Stehlé and Yamada provide a generic transformation from FE for linear functions to trace-and-revoke systems in [ABP⁺17]. Katsumata and Yamada recently show that one can build Non-Zero Inner Product Encryption from IPFE (cf. [KY19]).

We consider in the following $\mathsf{IND-CPA}$ security for IPFE: it corresponds to the same notion than for traditional encryption but with an additional access to an oracle for key derivation. A limitation is that the adversary can not obtain key for queries x such that $\langle x, y_0\rangle \neq \langle x, y_1\rangle$ where y_0, y_1 are the plaintexts of its choice.

This specific line of research on IPFE was initiated by Abdalla, Bourse, De Caro and Pointcheval in 2015 [ABDP15]. They provided the first IPFE schemes which rely on standard assumptions such as learning with errors (LWE) and decision Diffie Hellman (DDH). Their construction from DDH is built upon the so-called Elgamal in the exponent which restricts the size of the plaintext space. Moreover their schemes are only secure in the selective setting, *i.e.*, the adversary must commit to challenge plaintexts y_0, y_1 before having access to the schemes' public parameters.

Soon after, Agrawal, Libert and Stehlé have shown in [ALS16] how to enhance security with schemes under the LWE, DDH and DCR.

In a nutshell, several of these constructions follow the same pattern. One starts from a linearly homomorphic scheme with an Elgamal structure. Then one encrypts a vector y, coordinate per coordinate using a different public key for each coordinate but the same randomness. Key derivation for a vector x is done by computing the inner product $\langle s, x\rangle$ where s is the vector of secret keys. Then, using the homomorphic properties, from a ciphertext of y and $\langle s, x\rangle$ one can retrieve $\langle x, y\rangle$.

To reach full security, [ALS16] uses for the construction based on DDH and DCR linearly homomorphic schemes derived from HPS. As a result in the security proofs, the simulation knows the master secret key s and can answer key queries. It remains to show some enhanced form of smoothness: knowing the public key and answers to key queries of the form $\langle s, x\rangle$, s is not completely known to the adversary and the uncertainty allows to prove that the adversary can not get information on the plaintext from the challenge ciphertext using the hardness of the attached hard subset membership problem. This relation with HPS was further investigated in [BBL17] to get $\mathsf{IND}-\mathsf{CCA}$ secure IPFE.

In collaboration with F. Laguillaumie and I. Tucker [CLT18], we have revisited the approach of [ALS16] using the framework of a cyclic group with an easy DL subgroup presented in Section III.2. We have given two generic constructions of IPFE one using a modified CL scheme *à la* CS lite (cf. Fig. E.3 of [CLT18]) and the other one with the HSM variant of CL (cf. Fig. E.5 of [CLT18]). Following the approach of [ALS16] the two constructions are based on hash proof systems and we use a similar proof methodology to reach full security.

Compared with previous work, we do not have the limitation of DDH based constructions that encode the plaintext in the exponent and therefore limits the size of the inner product as a discrete logarithm must be computed during decryption. Moreover using our instantiation with class groups, we get an IPFE modulo a prime p with much better performance that the LWE scheme of [ALS16]. Compared with the construction based on Paillier's DCR which gives an IPFE modulo N where N is an RSA integer, the flexibility on the choice of size of the prime p allows to significantly reduce the size of keys and ciphertexts. See Table E.1 of [CLT18] for a detailed comparison. Concerning timings we have an encryption time of the same order of magnitude, but thanks to shorter keys, we significantly reduce decryption time.

Conclusion and Perspectives

This manuscript has presented many cryptographic applications of class groups of non maximal imaginary quadratic orders. We have seen how the ideas of the cryptanalysis of the NICE family of cryptosystems can be used constructively to instantiate the framework of a cyclic group with an easy DL subgroup. From this framework, we have shown how to build new linear homomorphic schemes modulo a prime p, the CL cryptosystem and its variants. With the protocols of the last chapter, we have seen that some properties of CL are appealing for applications. Compared to Elgamal in the exponent, CL has no limitation on the size of the messages, and compared with Paillier, CL gives flexibility on the size of the message space, that can be chosen independently of the security parameter. Moreover, working with a prime instead of an RSA integer can simplify protocols and proofs. Last but not least, the fact that the underlying algorithmic problems of class groups are harder than factoring and computing discrete logarithms in finite fields allows to have shorter key sizes in CL. As a result using CL in protocols significantly reduces the number of bits exchanged.

In the following, we discuss some perspectives on interesting related research topics.

Faster Implementation

While class groups of imaginary quadratic fields have been introduced in cryptography more than 20 years ago, they have been little effort on implementation and optimisation of the group law and the exponentiation, compared to elliptic curves for instance.

The fact that we can use lower parameters with class groups than with finite fields or quotients of $\mathbf{Z}$ brings a gain in terms of bit sizes in all situation. However the picture is less favorable in terms of timings.

We have already compared in Fig. III.3 the timings of exponentiation in class groups for our most versatile encryption scheme, $\mathsf{HSM}-\mathsf{CL}$, and the exponentiation in $\mathbf{Z}/N^2\mathbf{Z}$ used in Paillier, at the same security level. We have seen that exponentiation in class groups is less efficient at the 112 and 128 bits levels which are used in practice. Fortunately, when building protocols they are many more advantages of using $\mathsf{HSM}-\mathsf{CL}$ instead of Paillier that make the resulting protocols more efficient. However, they are situations where there is no advantage at the 112 and 128 bits levels. For example, in 2-party ECDSA signing, the protocol of Lindell uses only 2 exponentiations, while our solution needs 3 (2 can be done in parallel, though) so there is a loss in term of efficiency for this part at the 112 and 128 bits levels.

Our timings measurements where done with the Pari C library ([PAR18]) that uses a sliding window method combined with the Shanks's NUCOMP and NUDUPL methods for multiplication and squaring (see [JvdP02] for details on these algorithms). There exist other implementations; for example [DJS19] that implements the CL encryption scheme, uses libqform [Say]. Moreover, the recent proposals of verifiable delay functions (VDF) based on class groups ([BBBF18, Wes19]) have motivate research in this area. The Chia Network [Chi] has opened a competition for implementation for evaluation of those VDF that involve repeated squaring in class groups. The first round of this competition is over and a second round allowing SIMD and GPU optimizations is in progress as I write these lines.

Improvements could also come from advances in ideal arithmetic in class groups of quadratic fields. For instance Imbert, Jacobson, and Schmidt [IJS10] propose a method for computing the cube of an ideal class. Combined with double base chains using binary and ternary exponents, this leads to faster exponentiation.

More Cryptographic-Friendly Properties

Another direction of research is to find arithmetic properties that can lead to a wider class of cryptographic applications. A dream for applications will be to equip our framework of a cyclic group with an easy DL subgroup with a cryptographic bilinear map. This has been explored in the case of elliptic curve. An adaptation of the Paillier setting has been done by Galbraith in [Gal02], which shows the existence of an easy DL subgroup in the group of points of an elliptic curve over $\mathbf{Z}/N^2\mathbf{Z}$ for an RSA integer N. However, according to Galbraith and McKee in [GM05] it seems unlikely to use a pairing in this setting while keeping the factorisation of N unknown, which is necessary in order to hide the order of the groups of points of the elliptic curve, a requirement for our framework.

In class groups of quadratic fields it is also very unlikely to define such objects. However, other arithmetic properties might be useful for cryptography. For example the connection with isogeny based cryptography is a direction that is worth exploring. The structure of class groups is central to certain isogeny based cryptosystem (see for example [BKV19]). Moreover our representation of the kernel of the surjection between the class group of a non maximal order of conductor q and the class group of the maximal order has also been proven useful to obtain q^2-isogeny cycles (see [BLS12]).

Another direction of research is the adaptation of our work in the infrastructure of real quadratic fields. A direct adaptation might not be useful: it will be interesting from the mathematical point of view but less for cryptographic applications. The arithmetic will be less efficient and will not be compensate by smaller security parameters. However, this might be an interesting research area if we can exploit the infrastructure to obtain additional properties useful for cryptography.

An even broader research direction is adaptation in other number fields. Generalisations of class groups of quadratic fields based cryptography to arbitrary number fields is discussed in [BTV04] and there have been some proposal in small degree fields (*e.g.*, [BBHM02, MNP01]). The direction is to consider number fields with small regulators and large discriminants in order to have large class groups and be able to decide equality. However, arithmetic in the class group is very inefficient for fields of large degree and right choice of parameters is an open problem. Moreover, they have been advances on class group computations for large number fields (see [Gé18] for instance). Note that number fields are also used to design Fully Homomorphic Encryption scheme, using in particular the principal ideal problem.

Others Cryptographic Applications

Our framework of a cyclic group with an easy DL subgroup and the instantiation with class groups that leads to linear homomorphic schemes modulo a prime p have proven very useful for cryptographic applications. However, there are still many possible directions to explore.

In [CLT18, CCL+19] we have begun to use our constructions with the point of view of Hash Proof Systems (HPS). HPS have found many applications since their introduction by Cramer and Shoup in [CS02]. A logical follow-up to this work would be to enrich the properties of these HPS, in order to get $\mathsf{IND}-\mathsf{CCA}$ encryption schemes from our framework. Another interesting future work is to devise practical $\mathsf{IND}-\mathsf{CCA}$ inner product functional encryption schemes from the protocols of [CLT18] following the work of Benhamouda, Bourse, Lipmaa [BBL17].

Another cryptographic object that has many applications is the concept of Lossy Trapdoor Functions (LTDF) which was introduced by Peikert and Waters in [PW08]. LTDF can be built from smooth homomorphic HPS has shown in [HO09] and direct constructions from Paillier where proposed (see *e.g.*, [FGK+13]). As a result, there are few doubts that it will be possible to obtain a LTDF from our assumptions. It would thus be interesting to investigate if there will be a gain in term of efficiency or applications.

A natural follow up of our work on 2 party computation is to consider multiparty computation with the general threshold case. They have been a lot of recent activities in the area for threshold ECDSA signatures [GGN16, GG18, LN18, DKLs19]. A first point to address to move to this setting in the efficiency of zero knowledge proofs in our context, as we use a cyclic subgroup of a group of unknown order and that we can not check that elements belong to the subgroup. These zero knowledge proofs are necessary to deal with malicious adversaries that may corrupt participants of the protocol. In [CCL+19], the zero knowledge proof was not a crucial point as it was performed only once in the key generation phase. As a result, we used a Schnorr-type proof with binary challenges. However, we need more efficient solutions for the multiparty setting where numerous zero knowledge proofs might be needed for each signature generation.

There have been many proposals to deal with generalised Schnorr proofs in groups of unknown order (see for instance the framework of [CKY09] using safeguard groups, or [TW12]). For the case of subgroups of $(\mathbf{Z}/N\mathbf{Z})^\times$, efficient solutions for this type of proofs enlarge the challenge space, and rely on variants of the strong RSA assumption. For class groups, there have been informal proposals (see [DF02] for instance) using the root problem or the hardness of finding low order elements. However, computing square roots or finding elements of order 2 can be done efficiently in class groups knowing the factorisation of the discriminant (which is public in our case). Moreover, as suggested in [BBF18], in the context of verifiable delay functions, there might be other approaches to find low order elements in class groups. Advances in these subjects would lead to substantial efficiency improvements for cryptography based on class groups.

A related subject is the development of a threshold variant of the CL encryption scheme that can also be useful for applications in multiparty computation (for example, [GGN16] uses a threshold variant of Paillier for threshold ECDSA signatures). Again, the fact that we work with groups of unknown order will make delicate the design of efficient zero knowledge proofs. We also need a threshold secret sharing of the secret key over the integers, for which there have been some proposals (*e.g.*, [DT06]). While this unknown order seems to complicate things, it is also a positive point as this order is actually unknown to anyone during key generation (as computing the class number is an hard problem). As a result, one can hope to design an efficient protocol without trusted dealer. This feature of class groups was also used in [Lip12] for accumulators and [Wes19] for verifiable delay functions without trusted setup.

Appendices: Articles

Chapter A

On the Security of Cryptosystems with Quadratic Decryption: The Nicest Cryptanalysis

Joint work with Fabien Laguillaumie
[CL09]

Abstract. We describe the first *polynomial time chosen-plaintext total break* of the NICE family of cryptosystems based on ideal arithmetic in imaginary quadratic orders, introduced in the late 90's by Hartmann, Paulus and Takagi [HPT99]. The singular interest of these encryption schemes is their natural quadratic decryption time procedure that consists essentially in applying Euclid's algorithm. The only current specific cryptanalysis of these schemes is Jaulmes and Joux's chosen-ciphertext attack to recover the secret key [JJ00]. Originally, Hartmann *et al.* claimed that the security against a total break attack relies *only* on the difficulty of factoring the public discriminant $\Delta_q = -pq^2$, although the public key was also composed of a specific element of the class group of the order of discriminant Δ_q, which is crucial to reach the quadratic decryption complexity. In this article, we propose a drastic cryptanalysis which factors Δ_q (and hence recovers the secret key), only given this element, in cubic time in the security parameter. As a result, performing our cryptanalysis on a cryptographic example takes less than a second on a standard PC.

A.1. Introduction

We propose an original and radical cryptanalysis of a large class of schemes designed within imaginary quadratic fields, based on the NICE cryptosystem (cf. [HPT99, PT98, PT00]) which recovers the secret key from the sole public key. These systems have been intensively developed and studied in the late 90's, since they offer a very efficient secret operation (decryption or signature), compared to cryptosystems based on traditional number theory. The one-wayness of these schemes rely on the difficulty of the *Smallest Kernel-Equivalent Problem* (SKEP) and their security against a total break was believed to rely on the difficulty of the factorisation of numbers of the form pq^r. The first and only cryptanalysis of the NICE encryption scheme, proposed by Jaulmes and Joux's at Eurocrypt'00 [JJ00], recovers the secret key with an access to a decryption oracle[1]. In the setting of the NICE cryptosystems, the public key contains a discriminant $\Delta_q = -pq^2$ *and* the representation of a reduced ideal $\mathfrak{h}$ whose class belongs to the kernel of the surjection from the class group of the quadratic order of (public) discriminant $\Delta_q = -pq^2$ to the class group of the maximal order of (secret) discriminant $\Delta_K = -p$. We will show that with this knowledge of $\mathfrak{h}$ we can actually factor the public discriminant in cubic time in the security parameter.

A.1.1. Imaginary Quadratic Field-based Cryptography

The first use of class groups of imaginary quadratic fields allowed to achieve a Diffie-Hellman key exchange. This paper by Buchmann and Williams [BW88] was the first of several attempts to design imaginary quadratic field-based cryptosystems. Key exchange was also discussed by McCurley in [McC89]. Ten years after, a new encryption scheme appeared in the literature, in the work of Hühnlein, Jacobson, Paulus and Takagi [HJPT98]. The goal of this paper was also to improve the efficiency of the seminal cryptosystems. In fact, the key point of these Elgamal-like encryption schemes is the switching between the class group of the maximal order and the class group of a non-maximal order, which can be done with quadratic complexity (as already mentioned). Unfortunately, Hühnlein *et al.*'s scheme, although using this efficient switching, did not benefit from a quadratic time decryption since the decryption of this scheme really needed a final exponentiation (like in Elgamal).

Soon after, quadratic decryption time was eventually reached with a new encryption scheme, called *NICE*, for *New Ideal Coset Encryption*, described in [HPT99, PT98, PT00]. In [HPT99], it is shown that the decryption time of NICE is comparably as fast as the encryption time of RSA with public exponent $e = 2^{16} + 1$ and an even better implementation is described by Hühnlein in [Hüh99]. The key idea of NICE is not to mask the message by a power of the public key (which leads to a cubic decryption like in Elgamal), but by an element which belongs to the kernel of the map which switches between the class group of a non-maximal order to the maximal order. This hiding element is added to the public key and naturally disappears from the ciphertext when applying the map.

[1] This attack can actually be deflected by adding a suitable padding.

As the semantic security of NICE holds only under a chosen-plaintext attack, Buchmann, Sakurai and Takagi patched the scheme by adapting classical techniques to obtain a chosen-ciphertext security in the random oracle model [BST02]. This enhanced scheme, based on REACT [OP01] is called NICE-X, and of course resists Jaulmes and Joux's attack [JJ00]. Hühnlein, Meyer and Takagi also built in [HMT98] Rabin and RSA analogues based on non-maximal imaginary quadratic orders, but the only advantages over the original systems is their seemingly natural immunity against low exponent attacks and some chosen-ciphertext attacks.

The design of signature schemes has also been addressed in [HM00b, Hüh01] with an adaptation of Schnorr signatures (cf. [Sch90]). Again an element of the kernel of the switching between two class groups is published: this element is crucial for the efficiency of the signature generation. An undeniable signature scheme has been designed in [BPT04], and again, the public element of the kernel is needed for the design of an efficient scheme.

A.1.2. Related Work on Security Issues of Cryptography based on Quadratic Fields

All the NICE schemes share the same public information: a discriminant of the form $\Delta_q = -pq^2$ and the representation of a reduced ideal $\mathfrak{h}$ whose class belongs to the kernel of the surjection from the class group of the quadratic order of (public) discriminant $\Delta_q = -pq^2$ to the class group of the maximal order of (secret) discriminant $\Delta_K = -p$. Of course, a factorisation of the discriminant obviously totally breaks the scheme. Therefore, the security parameters are set such that the factorisation of numbers of the form pq^r is difficult. This particular factorisation has been addressed by Boneh, Durfee and Howgrave-Graham in [BDH99], but for small r (such as 2), their method is not better than Lenstra's ECM method [Len87] or the Number Field Sieve [LL93]. In [BST02], the authors also mention the *Quadratic Order Discrete Logarithm Problem* (QODLP). The fastest algorithm to solve the QODLP is the Hafner-McCurley algorithm [HM89], but its running time has a worse subexponential complexity than the fastest factoring algorithm. In [PT00], Paulus and Takagi argue that "the knowledge of $\mathfrak{h}$ does not substantially help to factor Δ_q using currently known fast algorithms". They also mention the possibility to find a power of the class $[\mathfrak{h}]$ of order 2, but computing the order of the class $[\mathfrak{h}]$ in the class group of the order of discriminant Δ_q is essentially equivalent to factor this discriminant. The problem of factoring the discriminant Δ_q given $[\mathfrak{h}]$ is called the Kernel Problem in [BPT04] and again is assumed to be "intractable".

Up to now, the sole specific cryptanalysis of this family of encryption schemes is the *chosen-ciphertext* nice cryptanalysis from [JJ00]. This attack uses the fact that the decryption fails (*i.e.*, does not recover the plain message) if the norm of the ideal representing the message is greater than $\sqrt{|\Delta_K|/3}$, so that the decoded message will expectedly be one step from being reduced. The relation between two pairs original message/decoded message leads to a Diophantine equation of the form $k = XY$ for a known "random" integer k of the size of the secret primes. The authors suggest to factor this integer to find out X and Y and then factor Δ_q. This attack is feasible for the parameters proposed in [HPT99], but can be defeated by enlarging the key size by a factor of 3. No complexity analysis is given for this attack, and the scheme can also be repaired by adding redundancy to the message as suggested in [JJ00] and [BST02]. Note that, contrary to ours, Jaulmes and Joux's attack also applies to [HJPT98].

A.1.3. Our contributions

We propose the first definitive cryptanalysis of cryptosystems based on NICE, which have been resisting for almost 10 years. All these schemes contain in the public key the representation of the reduced ideal $\mathfrak{h}$ whose class belongs to the kernel of the surjection from the class group of the quadratic order of discriminant $\Delta_q = -pq^2$ to the class group of the maximal order of discriminant $\Delta_K = -p$. The key point of our attack is the fact that this ideal $\mathfrak{h}$ is indeed always equivalent to a non-reduced ideal of norm q^2, as we will show in Theorem A−2. The core of our attack then consists of lifting the class of $\mathfrak{h}$ in the class group of the order of discriminant $\Delta_q r^2$, where r is chosen to make the ideals of norm q^2 reduced. This operation will reveal an ideal of norm q^2 and thus the factorisation of Δ_q, leading to a total break of the scheme.

Note that the public ideal $\mathfrak{h}$ is crucial in the design of NICE: Random powers of this element are used to hide the message. As it is in the kernel of a surjective map, this randomness can be removed from the ciphertext and the message recovered by applying this map which leads to a decryption algorithm with quadratic complexity (the computation is done with Euclid's algorithm).

The attack described in this paper thus uses this extra piece of information given in the public key to factor the public discriminant. Therefore, this setting is insecure in order to build a cryptosystem with quadratic decryption time. Note that such a scheme with quadratic decryption is a very rare object in group theory based cryptography. Although some schemes built from lattices or coding theory problems have this property, to our knowledge, very few schemes built from the integer factorisation or the discrete logarithm problems have it (*e.g.*, variants of the cryptosystems of Okamoto-Uchiyama and Paillier, cf. [CNP99, Pai99]). As a matter of fact, the encryption schemes built on NICE from [HPT99, PT98, PT00, BST02, Hüh99], the signature schemes [Hüh01, HM00b] and the undeniable signature scheme [BPT04] totally succumb to our attack.

The rest of the paper is organised as follows: The next section gives a background on orders of imaginary quadratic fields to understand the NICE cryptosystem, and then Section A.3 is the core of the paper. We describe the cryptanalysis by first discussing the (im-)possibility of reversing the reduction process applied on the reduced ideal $\mathfrak{h}$ in Subsection 3.1. Then, in Subsection 3.2, we describe our attack (Algorithm A.3) whose correctness is then proved with Theorem A−3 and Corollary A−1. Finally, we illustrate the attack with an example.

A.2. Background

The next subsection widely follows the description from [Cox99].

A.2.1. Computations in Quadratic Orders.

A *quadratic field* K is a subfield of the field of complex numbers $\mathbf{C}$ which has degree 2 over $\mathbf{Q}$. Such a field can be uniquely written as $\mathbf{Q}(\sqrt{n})$ where n is a square-free integer, different from 1 and 0. Its (fundamental) *discriminant* Δ_K is defined as n if $n \equiv 1 \pmod 4$ and $4n$ otherwise. We will then consider K in terms of its discriminant : $K = \mathbf{Q}(\sqrt{\Delta_K})$ with $\Delta_K \equiv 0, 1 \pmod 4$. An *order* $\mathcal{O}$ in K is a subset of K such that $\mathcal{O}$ is a subring of K containing 1 and $\mathcal{O}$ is a free $\mathbf{Z}$-module of rank 2. The ring $\mathcal{O}_{\Delta_K}$ of integers[2] in K is the *maximal* order of K in the sense that it contains all the other orders of K. It can be written as $\mathbf{Z} + \frac{1}{2}(\Delta_K + \sqrt{\Delta_K})\mathbf{Z}$. If we set $f = [\mathcal{O}_{\Delta_K} : \mathcal{O}]$ the *finite* index of any order $\mathcal{O}$ in $\mathcal{O}_{\Delta_K}$, then $\mathcal{O} = \mathbf{Z} + f\frac{1}{2}(\Delta_K + \sqrt{\Delta_K})\mathbf{Z} = \mathbf{Z} + f\mathcal{O}_{\Delta_K}$. The integer f is called the *conductor* of $\mathcal{O}$. The discriminant of $\mathcal{O}$ is then $\Delta_f = f^2\Delta_K$. We will then use the notation $\mathcal{O}_{\Delta_f}$ for such an order.

Now we discuss the ideals of an order $\mathcal{O}_\Delta$ of discriminant Δ. If $\mathfrak{a}$ is a nonzero ideal of $\mathcal{O}_\Delta$, its norm is defined as $N(\mathfrak{a}) = |\mathcal{O}_\Delta/\mathfrak{a}|$. An ideal $\mathfrak{a}$ is said to be *proper* if $\{\beta \in K : \beta\mathfrak{a} \subset \mathfrak{a}\} = \mathcal{O}_\Delta$. This definition can be extended to *fractional* ideals, which are of the form $\alpha\mathfrak{a}$ where $\alpha \in K^\times$ and $\mathfrak{a}$ is an ideal of $\mathcal{O}_\Delta$. If we denote by $I(\mathcal{O}_\Delta)$ the set of proper fractional ideals of $\mathcal{O}_\Delta$ and its subgroup $P(\mathcal{O}_\Delta)$ consisting of principal ideals, the *ideal class group* of $\mathcal{O}_\Delta$ is defined as $C(\mathcal{O}_\Delta) = I(\mathcal{O}_\Delta)/P(\mathcal{O}_\Delta)$. Its cardinality is the *class number* of $\mathcal{O}_\Delta$ denoted as $h(\mathcal{O}_\Delta)$.

Every ideal $\mathfrak{a}$ of $\mathcal{O}_\Delta$ can be written as

$$\mathfrak{a} = m\left(a\mathbf{Z} + \frac{-b + \sqrt{\Delta}}{2}\mathbf{Z}\right)$$

with $m \in \mathbf{Z}$, $a \in \mathbf{N}$ and $b \in \mathbf{Z}$ such that $b^2 \equiv \Delta \pmod{4a}$. In the sequel, we will only consider *primitive* ideals, which are those with $m = 1$. This expression is unique if $-a < b \leq a$ and we will now denote a primitive ideal by (a, b). The norm of such an ideal is then a.

This notation represents also the positive definite binary quadratic form $ax^2 + bxy + cy^2$ with $b^2 - 4ac = \Delta$. Theorem 7.7 from [Cox99] shows that, up to equivalence relations, it is essentially equivalent to work with ideals and positive definite quadratic forms. An ideal (a, b) of $\mathcal{O}_\Delta$ is said to be *reduced* if the corresponding quadratic form is reduced, which means that $|b| \leq a \leq c$ and $b \geq 0$ if one of the inequalities is not strict. Note that in every class of $\mathcal{O}_\Delta$-ideals there exists exactly one reduced ideal. From the theory of quadratic forms, we can efficiently compute a reduced equivalent ideal. The algorithm, which is due to Gauss, is described in [Coh00, Algorithm 5.4.2 p. 243] and is called Red in the rest of the paper. In general, instead of working with classes, we will work with reduced ideals. The product of ideals is also efficiently computable with the composition of quadratic forms algorithm, see [Coh00, Algorithm 5.4.7 p. 243]. These two algorithms have *quadratic* complexity. A crucial fact for our purpose is described

[2] *i.e.*, the set of all $\alpha \in K$ which are roots of a monic polynomial in $\mathbf{Z}[X]$

in Lemma 5.3.4 from [Coh00]: If an ideal $\mathfrak{a} = (a, b)$ is reduced, then $a \leq \sqrt{\Delta/3}$ and conversely, if $a < \sqrt{\Delta/4}$ and $-a < b \leq a$, then $\mathfrak{a}$ is reduced.

Let $\left(\frac{a}{b}\right)$ be the Kronecker symbol of a and b. The formula for the class number is given by the following theorem.

Theorem A−1 ([Cox99, Theorem 7.24]). *Let $\mathcal{O}_{\Delta_f}$ be the order of conductor f in an imaginary quadratic field* K *(i. e., $\Delta_f = f^2\Delta_{\mathrm{K}}$). Then*

$$h(\mathcal{O}_{\Delta_f}) = \frac{h(\mathcal{O}_{\Delta_{\mathrm{K}}})f}{[\mathcal{O}_{\Delta_{\mathrm{K}}}^{\times} : \mathcal{O}_{\Delta_f}^{\times}]} \prod_{p|f}\left(1 - \left(\frac{\Delta_{\mathrm{K}}}{p}\right)\frac{1}{p}\right).$$

Given an order $\mathcal{O}_{\Delta_f}$ of conductor f, a nonzero $\mathcal{O}_{\Delta_f}$-ideal $\mathfrak{a}$ is said to be *prime to f* if $\mathfrak{a} + f\mathcal{O}_{\Delta_f} = \mathcal{O}_{\Delta_f}$ (it is equivalent to say that its norm $\mathrm{N}(\mathfrak{a})$ is prime to f − see Lemma 7.18 from [Cox99]). We denote by $\mathrm{I}(\mathcal{O}_{\Delta_f}, f)$ the subgroup of $\mathrm{I}(\mathcal{O}_{\Delta_f})$ generated by ideals prime to f. $\mathrm{P}(\mathcal{O}_{\Delta_f}, f)$ is the subgroup generated by the principal ideals $\alpha\,\mathcal{O}_{\Delta_f}$ where $\alpha \in \mathcal{O}_{\Delta_f}$ has a norm prime to f. Note that in every ideal class, there exists an ideal prime to f (cf. [Cox99, Corollary 7.17]). To establish Theorem A−1, Cox has studied the links between the class group of the maximal order of an imaginary quadratic field and the class groups of any of its orders. The following propositions throw a light on such fundamental links.

Proposition A−1 ([Cox99, Proposition 7.19]). *The inclusion $\mathrm{I}(\mathcal{O}_{\Delta_f}, f) \subset \mathrm{I}(\mathcal{O}_{\Delta_f})$ induces an isomorphism*

$$\mathrm{I}(\mathcal{O}_{\Delta_f}, f)/\mathrm{P}(\mathcal{O}_{\Delta_f}, f) \simeq \mathrm{I}(\mathcal{O}_{\Delta_f})/\mathrm{P}(\mathcal{O}_{\Delta_f}) = \mathrm{C}(\mathcal{O}_{\Delta_f}).$$

Proposition A−2 ([Cox99, Proposition 7.20]). *Let $\mathcal{O}_{\Delta_f}$ be an order of conductor f in an imaginary quadratic field* K.

1. *If $\mathfrak{A}$ is an $\mathcal{O}_{\Delta_{\mathrm{K}}}$-ideal prime to f, then $\mathfrak{A} \cap \mathcal{O}_{\Delta_f}$ is an $\mathcal{O}_{\Delta_f}$-ideal prime to f of the same norm.*
2. *If $\mathfrak{a}$ is an $\mathcal{O}_{\Delta_f}$-ideal prime to f, then $\mathfrak{a}\mathcal{O}_{\Delta_{\mathrm{K}}}$ is an $\mathcal{O}_{\Delta_{\mathrm{K}}}$-ideal prime to f of the same norm.*
3. *The map $\varphi_f : \mathrm{I}(\mathcal{O}_{\Delta_f}, f) \longrightarrow \mathrm{I}(\mathcal{O}_{\Delta_{\mathrm{K}}}, f)$ such that $\mathfrak{a} \mapsto \mathfrak{a}\mathcal{O}_{\Delta_{\mathrm{K}}}$ is an isomorphism.*

Consequently, the map φ_f from Proposition A−2 induces a surjection

$$\bar{\varphi}_f : \mathrm{C}(\mathcal{O}_{\Delta_f}) \twoheadrightarrow \mathrm{C}(\mathcal{O}_{\Delta_{\mathrm{K}}})$$

that can be computed as follows: given a class $[\mathfrak{a}] \in \mathrm{C}(\mathcal{O}_{\Delta_f})$, one finds $\mathfrak{b} \in [\mathfrak{a}]$ such that $\mathfrak{b} \in \mathrm{I}(\mathcal{O}_{\Delta_f}, f)$ (see standard techniques [HJPT98, Algorithm 1]) and $\bar{\varphi}_f([\mathfrak{a}]) = [\varphi_f(\mathfrak{b})] = [\mathfrak{b}\mathcal{O}_{\Delta_{\mathrm{K}}}]$. The next two algorithms compute φ_f and its inverse (cf. [PT00]).

```
Input: $\mathfrak{A} = (A, B) \in I(\mathcal{O}_{\Delta_K}, f)$
Output: $\mathfrak{A} \cap \mathcal{O}_{\Delta_f} = (a, b) \in I(\mathcal{O}_{\Delta_f}, f)$
  1. $a \leftarrow A$
  2. $b \leftarrow Bf \mod_c 2a$ $(|b| < a)$        [centered euclidean division]
  3. Return $(a, b)$
```

Algorithm A.1: Algorithm to compute φ_f^{-1}

```
Input: $\mathfrak{a} = (a, b) \in I(\mathcal{O}_{\Delta_f}, f), \Delta_f$
Output: $\mathfrak{a}\mathcal{O}_{\Delta_K} = (A, B) \in I(\mathcal{O}_{\Delta_K}, f)$
  1. $A \leftarrow a$
  2. $\delta \leftarrow \Delta_f \mod 2$
  3. Compute $u$ and $v \in \mathbf{Z}$ such that $1 = uf + a\delta v$        [extended Euclidean
     algorithm]
  4. $B \leftarrow bu + a\delta v \mod_c 2a$ $(|B| < a)$        [centered euclidean division]
  5. Return $(A, B)$
```

Algorithm A.2: Algorithm to compute φ_f

Takagi and Paulus showed, in Section 4.2 from [PT00], that in the NICE setting the computation of this homomorphism φ_f cannot be done without the knowledge of the prime secret conductor.

The following effective lemma was used in [Cox99] to prove the formula of Theorem A−1 by computing the order of $\ker \bar{\varphi}_f$ and by Hühnlein, for example in [Hüh99,Hüh01] to efficiently compute in $\ker \bar{\varphi}_f$. It also proves the correctness of our attack. Indeed with a well-known system of representatives of $\left(\mathcal{O}_{\Delta_K}/f\mathcal{O}_{\Delta_K}\right)^\times / \left(\mathbf{Z}/f\mathbf{Z}\right)^\times$, we will derive a suitable system of representatives for $\ker \bar{\varphi}_f$, which is essential for the proofs of Theorem A−2 and Lemma A−2.

Lemma A − 1. *Let Δ_K be a fundamental negative discriminant, different from -3 and -4, and f a conductor. Then there exists an effective isomorphism*

$$\psi_f\colon \left(\mathcal{O}_{\Delta_K}/f\mathcal{O}_{\Delta_K}\right)^\times / \left(\mathbf{Z}/f\mathbf{Z}\right)^\times \xrightarrow{\sim} \ker \bar{\varphi}_f.$$

We will denote by $\phi_{\Delta_K}(f) := f \prod_{p|f} \left(1 - \left(\frac{\Delta_K}{p}\right)\frac{1}{p}\right)$ *the order of* $\ker \bar{\varphi}_f$.

Proof. The proof follows the line of the proof of [Cox99, Proposition 7.22 and Theorem 7.24]. □

Remark A – 1. To effectively map a class from $\left(\mathcal{O}_{\Delta_K}/f\mathcal{O}_{\Delta_K}\right)^\times / \left(\mathbf{Z}/f\mathbf{Z}\right)^\times$ to $\ker\bar{\varphi}_f$, one takes a representative $\alpha \in \mathcal{O}_{\Delta_K}$, $\alpha := x + y\frac{\Delta_K + \sqrt{\Delta_K}}{2}$ where $x, y \in \mathbf{Z}$ and $\gcd(N(\alpha), f) = 1$ (to ensure that α is invertible modulo $f\mathcal{O}_{\Delta_K}$), and computes

$$\psi_f\left([\alpha]\right) = \left[\varphi_f^{-1}(\alpha\mathcal{O}_{\Delta_K})\right],$$

which is in $\ker\bar{\varphi}_f$. In this computation, the representation of $\alpha\mathcal{O}_{\Delta_K}$ can be obtained with [BTW95, Proposition 2.9] and the evaluation of φ_f^{-1} with Algorithm A.1.

Conversely, given a class of $\ker\bar{\varphi}_f$ usually represented by its reduced ideal, one finds a representative ideal $\mathfrak{h} \in I(\mathcal{O}_{\Delta_f}, f)$ (with [HJPT98, Algorithm 1]) and computes $\alpha \in \mathcal{O}_{\Delta_K}$ such that $\alpha\mathcal{O}_{\Delta_K} = \varphi_f(\mathfrak{h})$ (φ_f is evaluated with Algorithm A.2 and α can be found with [HJW03, Algorithm 1]). Eventually, $\psi_f^{-1}([\mathfrak{h}]) = [\alpha] \in \left(\mathcal{O}_{\Delta_K}/f\mathcal{O}_{\Delta_K}\right)^\times / \left(\mathbf{Z}/f\mathbf{Z}\right)^\times$.

A.2.2. The NICE family

We will now describe in Fig. A.1 the original NICE cryptosystem as it is presented in [PT00]. For our purpose, it is only important to concentrate on the key generation which outputs an element $[\mathfrak{h}]$ of $\ker\bar{\varphi}_q$ as a part of the public key. Other encryption schemes which share this key generation can be found in [HPT99, PT00, BST02, Hüh99, PT98], and signature schemes in [Hüh01, HM00b, BPT04]. As already mentioned, all these cryptosystems succumb to our attack.

Underlying Algorithmic Assumptions

The security against a total break (resp. of the one-wayness) of the NICE cryptosystem is proved to rely on the hardness of the following problems:

Definition A – 1 (Kernel Problem [BPT04])**.** Let λ be an integer, p and q be two λ-bit primes with $p \equiv 3 \pmod 4$. Fix a non-fundamental discriminant $\Delta_q = -pq^2$. Given an element $[\mathfrak{h}]$ of $\ker\bar{\varphi}_q$, factor the discriminant Δ_q.

Definition A – 2 (Smallest Kernel-Equivalent Problem [BST02, BPT04] (SKEP))**.** Let λ be an integer, p and q be two λ-bit primes with $p \equiv 3 \pmod 4$. Fix a non-fundamental discriminant $\Delta_q = -pq^2$. Given an element $[\mathfrak{h}]$ of $\ker\bar{\varphi}_q$ and an element $[\mathfrak{m}] \in C(\mathcal{O}_{\Delta_q})$, compute the ideal with the smallest norm in the equivalence class, modulo the subgroup generated by $[\mathfrak{h}]$, of $[\mathfrak{m}]$.

It is clear that an algorithm which solves the Kernel Problem also solves the Smallest Kernel-Equivalent Problem. The insecurity of the Kernel Problem will be discussed in the next section.

KeyGen(1^λ)**:**

- Let p be a λ-bit prime such that $p \equiv 3 \pmod 4$ and let q be a prime such that $q > \sqrt{p/3}$.
- Set
$$\begin{cases} \Delta_K = -p \\ \Delta_q = \Delta_K q^2 = -pq^2 \end{cases}$$
- Let k and l be the bit lengths of $\lfloor\sqrt{|\Delta_K|/4}\rfloor$ and $q - \left(\frac{\Delta_K}{q}\right)$ respectively.
- Let $[\mathfrak{h}]$ be an element of $\ker\bar{\varphi}_q$, where $\mathfrak{h}$ is a reduced $\mathcal{O}_{\Delta_q}$- ideal.

The public key pk consists of the quadruple $(\Delta_q, \mathfrak{h}, k, l)$, and the secret key sk consists of the pair (p, q).

Encrypt$(1^\lambda, pk, m)$**:**

- A message m is embedded into a reduced $\mathcal{O}_{\Delta_q}$-ideal $\mathfrak{m}$ with $\log_2(\mathrm{N}(\mathfrak{m})) < k$.
- Pick randomly $r \in [\![1, 2^{l-1}]\!]$ and compute $\mathfrak{c} = \mathsf{Red}(\mathfrak{m} \times \mathfrak{h}^r)$.

Decrypt$(1^\lambda, sk, \mathfrak{c})$**:** Compute $\varphi_q^{-1}(\mathsf{Red}(\varphi_q(\mathfrak{c}))) = \mathfrak{m}$.

Figure A.1: Description of NICE

A.3. The Cryptanalysis

A.3.1. Intuition

In the NICE setting, $\Delta_K = -p$, $\Delta_q = \Delta_K q^2$ where p and q are two large primes, and the schemes are totally broken if one can recover p and q from Δ_q. (Un-)fortunately, another piece of information is given in the public key: an ideal $\mathfrak{h}$ whose class belongs to the kernel of $\bar{\varphi}_q$, the surjection from $C(\mathcal{O}_{\Delta_q})$ to $C(\mathcal{O}_{\Delta_K})$. In [PT00] (for example), the authors suppose that no ideal whose class is in $\ker\bar{\varphi}_q$ leaks a factor of the public discriminant Δ_q, except if this element has order 2, but then a subexponential computation is required to find it.

While investigating this assumption, we experimentally found non-reduced ideals of the form (q^2, kq), with k odd and $|k| < q$ whose classes belong to the kernel of $\bar{\varphi}_q$, and which obviously give the factorisation of Δ_q. By using the effective isomorphism of Lemma A−1, we actually prove in the next theorem that one can build a representative set of this kernel with ideals of norm q^2.

Theorem A − 2. *Let* Δ_K *be a fundamental negative discriminant, different from* -3 *and* -4 *and* q *an odd prime conductor. There exists an ideal of norm* q^2 *in each nontrivial class of* $\ker\bar{\varphi}_q$.

Proof. Let us recall the effective isomorphism from Lemma A − 1:

$$\psi_q\colon \left(\mathcal{O}_{\Delta_K}/q\mathcal{O}_{\Delta_K}\right)^{\times}/\left(\mathbf{Z}/q\mathbf{Z}\right)^{\times} \xrightarrow{\ \sim\ } \ker\bar{\varphi}_q.$$

We are going to build a set of representatives of $\left(\mathcal{O}_{\Delta_K}/q\mathcal{O}_{\Delta_K}\right)^{\times}/\left(\mathbf{Z}/q\mathbf{Z}\right)^{\times}$ and apply ψ_q (which can be computed according to Remark A − 1) to obtain ideals of norm q^2 which are a set of representatives of $\ker\bar{\varphi}_q$.

Let us set $\alpha_k = k + \frac{\Delta_K+\sqrt{\Delta_K}}{2}$ with $k \in \{0,\dots,q-1\}$. Clearly,

$$\mathrm{N}(\alpha_k) = \left(k + \frac{\Delta_K}{2}\right)^2 - \frac{\Delta_K}{4} = k^2 + \Delta_K k + \frac{\Delta_K(\Delta_K-1)}{4}.$$

Consider the following set of representatives of $\left(\mathcal{O}_{\Delta_K}/q\mathcal{O}_{\Delta_K}\right)^{\times}/\left(\mathbf{Z}/q\mathbf{Z}\right)^{\times}$:

$$\left\{1\right\} \cup \left\{\alpha_k \text{ with } k \in \{0,\dots,q-1\},\ \mathrm{N}(\alpha_k) \not\equiv 0 \pmod q\right\},$$

indeed, it is easy to check that all the α_k belong to different classes and that they are in sufficient number: If $\left(\frac{\Delta_K}{q}\right)$ equals 1 (resp. equals 0, resp. equals -1) then the order of the quotient $\left(\mathcal{O}_{\Delta_K}/q\mathcal{O}_{\Delta_K}\right)^{\times}/\left(\mathbf{Z}/q\mathbf{Z}\right)^{\times}$ is $1+(q-2)$ (resp. $1+(q-1)$, resp. $1+(q+1)$). We are now going to compute the image of this set by ψ_q in $\ker\bar{\varphi}_q$.

Following the proof of [BTW95, Proposition 2.9], we detail here the computation of $\mathfrak{A}_k = \alpha_k\mathcal{O}_{\Delta_K}$. The representation of $\mathfrak{A}_k$ is (a_k, b_k), with $a_k = \mathrm{N}(\alpha_k)$. Let us now find b_k. The representation of $\mathcal{O}_{\Delta_K}$ is $\left(1, \frac{\Delta_K+\sqrt{\Delta_K}}{2}\right)$. A simple calculation gives

$$\alpha_k\mathcal{O}_{\Delta_K} = \alpha_k\mathbf{Z} + \left(\frac{k\Delta_K}{2} + \frac{\Delta_K(\Delta_K+1)}{4} + (k+\Delta_K)\frac{\sqrt{\Delta_K}}{2}\right)\mathbf{Z}$$

which must be equal to $m_k\left(a_k\mathbf{Z} + \frac{-b_k+\sqrt{\Delta_K}}{2}\mathbf{Z}\right)$. As mentioned in the proof of [BTW95, Proposition 2.9], m_k is the smallest positive coefficient of $\sqrt{\Delta_K}/2$ in $\mathfrak{A}_k$: in our case $m_k = \gcd(1, k+\Delta_K)$ and therefore $m_k = 1$.

Since $\alpha_k \in \alpha_k\mathcal{O}_{\Delta_K}$, there exists μ_k and ν_k such that $\alpha_k = a_k\mu_k + \frac{-b_k+\sqrt{\Delta_K}}{2}\nu_k$. By identification in the basis $(1,\sqrt{\Delta_K})$, $\nu_k = 1$ and by a multiplication by 2, we obtain $2k + \Delta_K = 2a_k\mu_k - b_k$. As the value of b_k is defined modulo $2a_k$, we can take

$$b_k = -2k - \Delta_K.$$

We now need to compute $\varphi_q^{-1}(\mathfrak{A}_k)$. From Algorithm A.1, it is equal to $(a_k, b_k q \mod 2a_k)$. Eventually, in every nontrivial class of $\ker \bar{\varphi}_q$, there exists an ideal $(a_k, b_k q)$. This ideal corresponds to the quadratic form $a_k x^2 + b_k qxy + c_k y^2$ with

$$c_k = \frac{q^2\left((2k+\Delta_K)^2 - \Delta_K\right)}{4(k^2+\Delta_K k) + \Delta_K(\Delta_K - 1)} = q^2,$$

which is then equivalent to the form $q^2x^2 - b_k qxy + a_k y^2$ corresponding to the ideal $(q^2, -b_k q)$ whose norm is q^2. □

A first attempt: inverting the reduction process.

From this theorem, the reduced ideal $\mathfrak{h}$ published in the NICE cryptosystems is equivalent to an ideal of norm q^2. A first attack is thus to try to do a brute force ascent of the reduction algorithm, *i. e.*, the Gauss algorithm, from $\mathfrak{h}$. To “invert” a step of this algorithm (see Algorithms 1.3.14 and 5.4.2 of [Coh00]), one has to consider all the possible quotients of the Euclidean division. The number of possible quotients is heuristically low (say ten), and the complexity of the attack grows exponentially with the number of reduction steps. If this number is very low, the attack will be feasible. In particular, if $q < \sqrt{p/4}$, all ideals of the form (q^2, kq) are already reduced, so the norm of $\mathfrak{h}$ is q^2 and the schemes are insecure. If the parameters for NICE are chosen as proposed in [PT00] (*i. e.*, $\sqrt{p/3} < q$) the number of reduction steps can still be too low. In the given implementation and later papers (*e. g.*, [BST02]), p and q are actually chosen of same size λ, the security parameter. Let us analyse more generally the numbers of reduction steps needed to reduce ideals of the form (q^2, kq) in $C(\mathscr{O}_{\Delta_q})$.

If we translate the problem in terms of quadratic forms, the quadratic form $q^2x^2 + kqxy + c(k)^2y^2$, with $c(k) := \frac{1}{4}(k^2+p)$, can be represented by the matrix

$$\begin{pmatrix} q^2 & kq/2 \\ kq/2 & c(k) \end{pmatrix},$$

which defines (up to an isometry) two vectors u and v of $\mathbf{C}$ such that $|u|^2 = q^2$, $|v|^2 = c(k)$ and $\langle u, v\rangle = kq/2$, where $\langle\cdot,\cdot\rangle$ denotes the usual scalar product in $\mathbf{C}$. If we consider the complex number $z = \frac{v}{u}$ (we suppose here that u is larger than v, *i. e.*, $q^2 > \frac{1}{4}(k^2+p)$), then

$$z = \frac{\langle u, v\rangle}{|u|^2} + i\frac{\det(u,v)}{|u|^2} = \frac{kq}{2q^2} + i\frac{\sqrt{|\Delta_q|}}{q^2} = \frac{k}{2q} + i\frac{\sqrt{p}}{q}\cdot$$

The mean number of iteration A_h of the Gauss algorithm when the complex number z belongs to the strip $\{|\Im(z)| \le 1/h\}$ is heuristically

$$A_h \sim \frac{1}{2}\log h \left[\frac{1}{\log(1+\sqrt{2})} - \frac{1}{\log\left(\frac{\pi^2}{6\log\phi}\right)}\right],$$

where ϕ is the golden ratio.

Inside this horizontal strip, the complex numbers z for which the number of iterations is of order $\Omega(\log L)$ are those for which their real part $\Re(z)$ is close to a rational number whose continued fraction expansion is of order $\Omega(\log L)$.

Then, since our complex number z is of the form $z = \frac{k}{2q} + i\frac{\sqrt{p}}{q}$, the number of iterations of the Gauss Algorithm on the input z will be (with a high probability) of order $\Omega(\log qp^{-\frac{1}{2}})$ provided that the height of the continued fraction expansion of the rational number k/q is of order $\Omega(\log q)$ (which is always the case, with a high probability). See [VV07] for a precise analysis of Gauss algorithm. If we set $q = p^{\alpha}$ these theoretical results give a behaviour in $\Omega\Big(\big(\alpha - \frac{1}{2}\big)\log p\Big)$, and therefore if we set $\alpha = 1$ as suggested in [BST02], we have a number of steps proportional to $\log p/2 = \lambda/2$ so the going up is infeasible. Note that our experiments confirm this complexity. Therefore we have to establish another strategy to recover these non-reduced ideal of norm q^2.

A.3.2. An Algorithm to Solve the Kernel Problem

Description.

In this subsection, we describe an algorithm (A.3) which totally breaks the NICE family of cryptosystems by solving the Kernel Problem in polynomial time in the security parameter. More precisely, given $\Delta_q = -pq^2$ where p and q are two λ-bit primes and $\mathfrak{h}$ a reduced ideal whose class is in the kernel of the surjection from $C(\mathcal{O}_{\Delta_q})$ to $C(\mathcal{O}_{\Delta_K})$, this algorithm outputs p and q in cubic time. The next subsection is dedicated to the analysis of the correctness and the complexity of this algorithm. The main result is given in Corollary A – 1.

The strategy of the attack, detailed in the next algorithm, is as follows. First, in an initialisation phase (steps 1–3), we generate a power r of a small odd prime. This integer r is chosen large enough to make the ideals of norm q^2 reduced in $C(\mathcal{O}_{\Delta_q r^2})$. Then, the core of the algorithm consists in lifting $[\mathfrak{h}']$ (where $\mathfrak{h}'$ is equivalent to $\mathfrak{h}$ and prime to r) in this class group. In step 5, we compute $\mathfrak{g} = \mathfrak{h}' \cap \mathcal{O}_{\Delta_q r^2}$, which is an $\mathcal{O}_{\Delta_q r^2}$-ideal, with Algorithm A.1 (this algorithm still works between two non-maximal orders).

Then, in step 6, we compute the reduced element $\mathfrak{k}$ of the class of $\mathfrak{g}$ raised to the power $\phi_{\Delta_K}(r)$. In the next subsection, we will prove that this lift (steps 5 and 6) maps almost all the elements of $\ker\bar{\varphi}_q$, including $[\mathfrak{h}]$, to elements of $\ker\bar{\varphi}_{qr}$ whose reduced ideal has norm q^2. As a consequence, the ideal $\mathfrak{k}$ computed in step 6 has norm q^2 and eventually step 7 extracts p and q.

Remark A – 2. We omit elements of small order in the input of our algorithm, because they are useless for the NICE cryptosystems. As we will see in the proof of Corollary A – 1, this restriction ensures that the incrementation of step 3 will be done at most once. For completeness, if the order of $[\mathfrak{h}]$ is 3, only few iterations will be done to obtain a suitable r such that the order of $[\mathfrak{h}]$ does not divide $\phi_{\Delta_K}(r) = r'^{\delta_{r'}-1}\left(r' - \left(\frac{\Delta_K}{r'}\right)\right)$, and

```
Input: $\lambda \in \mathbf{Z}, \Delta_q = -pq^2 \in \mathbf{Z}, \mathfrak{h} = (a, b) \in \mathrm{I}(\mathcal{O}_{\Delta_q}, q)$ with $[\mathfrak{h}] \in \ker \bar{\varphi}_q$ of order $> 6$
Output: $p, q$
Initialisation:
    1. Set $r' = 3$
    2. Set $\delta_{r'} = \lceil \frac{\lambda+3}{2} \frac{\log 2}{\log r'} \rceil$ and $r = r'^{\delta_{r'}}$
    3. If the order of $[\mathfrak{h}]$ divides $\phi_{\Delta_K}(r)$ then set $r'$ to the next prime and goto 2.
    4. Find $\mathfrak{h}' \in [\mathfrak{h}]$ such that $\mathfrak{h}' \in \mathrm{I}(\mathcal{O}_{\Delta_q}, r')$        [HJPT98, Algorithm 1]
Core Algorithm:
    5. Compute $\mathfrak{g} = \mathfrak{h}' \cap \mathcal{O}_{\Delta_q r^2}$        [Algorithm A.1]
    6. Compute $\mathfrak{f} = \mathsf{Red}(\mathfrak{g}^{\phi_{\Delta_K}(r)})$
    7. Return $p = \Delta_q / \mathrm{N}(\mathfrak{f}), q = \sqrt{\mathrm{N}(\mathfrak{f})}$
```

Algorithm A.3: Solving the Kernel Problem

for an order of 5, $r' = 3$ suits. Note also that elements of order 2 (4 and 6) leads to ambiguous ideals which give the factorisation of the discriminant (see [Sch82]).

Correctness.

Again, the proof of correctness of Algorithm A.3 will be done by using the effective isomorphisms between the groups $\ker \bar{\varphi}_q$ and $\left(\mathcal{O}_{\Delta_K}/q\mathcal{O}_{\Delta_K}\right)^\times / \left(\mathbf{Z}/q\mathbf{Z}\right)^\times$ and between $\ker \bar{\varphi}_{qr}$ and $\left(\mathcal{O}_{\Delta_K}/qr\mathcal{O}_{\Delta_K}\right)^\times / \left(\mathbf{Z}/qr\mathbf{Z}\right)^\times$. The integer r is an odd integer prime to q and Δ_K such that $r > 2q/\sqrt{|\Delta_K|}$, *i. e.*, such that ideals of norm q^2 are reduced in $\mathrm{C}(\mathcal{O}_{\Delta_q r^2})$.

First in Lemma A−2, we prove that nontrivial elements of a certain subgroup of $\left(\mathcal{O}_{\Delta_K}/qr\mathcal{O}_{\Delta_K}\right)^\times / \left(\mathbf{Z}/qr\mathbf{Z}\right)^\times$ map to classes of $\ker \bar{\varphi}_{qr}$ whose reduced element has norm q^2. Actually, this subgroup contains the image of a particular lift of $\left(\mathcal{O}_{\Delta_K}/q\mathcal{O}_{\Delta_K}\right)^\times / \left(\mathbf{Z}/q\mathbf{Z}\right)^\times$ following the Chinese remainder theorem: A class $[\alpha]$ modulo q is lifted to a class $[\beta]$ modulo qr such that $[\beta] \equiv 1 \pmod{r}$ and $[\beta] \equiv [\alpha]^{\phi_{\Delta_K}(r)} \pmod{q}$.

Then, in Theorem A−3, we prove that the lift computed in steps 4 and 6 of Algorithm A.3 corresponds to the lift previously mentioned on the quotients of $\mathcal{O}_{\Delta_K}$. As a result, this lift evaluated on an element of $\ker \bar{\varphi}_q$ either gives the trivial class or a class corresponding to the nontrivial elements of the subgroup of Lemma A−2, *i. e.*, a class whose reduced element has norm q^2.

Finally, in Corollary A−1, we prove that Algorithm A.3 is polynomial and correct, *i. e.*, that the choice of r done in the initialisation of the algorithm ensures that the lift will produce a nontrivial class and hence an ideal of norm q^2.

Lemma A − 2. *Let Δ_K be a fundamental negative discriminant, different from -3 and -4 and q an odd prime conductor and r be an odd integer prime to q and Δ_K such that $r > 2q/\sqrt{|\Delta_K|}$. The isomorphism ψ_{qr} of Lemma A−1 maps the nontrivial elements of the kernel of this natural surjection*

$$\pi : \left(\mathcal{O}_{\Delta_K}/qr\mathcal{O}_{\Delta_K}\right)^\times / \left(\mathbf{Z}/qr\mathbf{Z}\right)^\times \longrightarrow\!\!\!\!\!\rightarrow \left(\mathcal{O}_{\Delta_K}/r\mathcal{O}_{\Delta_K}\right)^\times / (\mathbf{Z}/r\mathbf{Z})^\times$$

to classes of $\ker \bar{\varphi}_{qr} \subset C(\mathcal{O}_{\Delta_K q^2 r^2})$*, whose reduced element has norm* q^2.

Proof. This proof is similar to the proof of Theorem A−2, but relative to r (more precisely, specialising $r = 1$ in this lemma yields Theorem A−2).

Let us set $\alpha_k = k + r\frac{\Delta_K+\sqrt{\Delta_K}}{2}$ where $k \in \mathbf{Z}$ takes $\phi_{\Delta_K}(q)$ values s.t.

$$\begin{cases} k & \not\equiv & 0 \pmod{r}, \\ k & \equiv & 0, \dots, q-1 \pmod{q}, \\ k^2 & \not\equiv & r^2\Delta_K \pmod{q}, \end{cases}$$

and denote $\mathcal{S} = \{1\} \cup \{\alpha_k\}_k$. For each k, the norm $N(\alpha_k)$ is equal to $\left(k + r\frac{\Delta_K}{2}\right)^2 - \Delta_K\frac{r^2}{4}$.

Since r is prime to q, the Chinese remainder theorem gives the isomorphism between the quotient $\left(\mathcal{O}_{\Delta_K}/qr\mathcal{O}_{\Delta_K}\right)^\times / \left(\mathbf{Z}/qr\mathbf{Z}\right)^\times$ and

$$\left(\left(\mathcal{O}_{\Delta_K}/q\mathcal{O}_{\Delta_K}\right)^\times / \left(\mathbf{Z}/q\mathbf{Z}\right)^\times\right) \times \left(\left(\mathcal{O}_{\Delta_K}/r\mathcal{O}_{\Delta_K}\right)^\times / (\mathbf{Z}/r\mathbf{Z})^\times\right).$$

As all the elements of $\mathcal{S}$ map to the neutral element in $\left(\mathcal{O}_{\Delta_K}/r\mathcal{O}_{\Delta_K}\right)^\times / (\mathbf{Z}/r\mathbf{Z})^\times$ and gives all the elements of $\left(\mathcal{O}_{\Delta_K}/q\mathcal{O}_{\Delta_K}\right)^\times / \left(\mathbf{Z}/q\mathbf{Z}\right)^\times$, $\mathcal{S}$ is actually a set of representatives of $\ker \pi$.

Let us now compute $\mathfrak{A}_k = \alpha_k\mathcal{O}_{\Delta_K}$. Its representation is (a_k, b_k), with $a_k = N(\alpha_k)$ and then

$$\alpha_k\mathcal{O}_{\Delta_K} = \alpha_k\mathbf{Z} + \left(k + r\frac{\Delta_K + \sqrt{\Delta_K}}{2}\right)\left(\frac{\Delta_K + \sqrt{\Delta_K}}{2}\right)\mathbf{Z},$$

which must be equal to $m_k\left(a_k\mathbf{Z} + \frac{-b_k+\sqrt{\Delta_K}}{2}\mathbf{Z}\right)$. Then, $m_k = \gcd(r, r\Delta_K + k)$ which is equal to 1 since $\gcd(k, r) = 1$.

As $\alpha_k \in \alpha_k\mathcal{O}_{\Delta_K}$, there exists μ_k and ν_k such that $\alpha_k = a_k\mu_k + \frac{-b_k+\sqrt{\Delta_K}}{2}\nu_k$. By identification in the basis $(1, \sqrt{\Delta_K})$, $\nu_k = r$ and by multiplying by 2, we obtain $2k+r\Delta_K = 2a_k\mu_k - rb_k$ and again we can take

$$b_k = \frac{-2k}{r} - \Delta_K.$$

Then $\varphi_{qr}^{-1}(\mathfrak{A}_k)$ is equal to (a_k, B_k) where $B_k = b_k qr$. This ideal corresponds to the quadratic form $a_k x^2 + B_k xy + c_k y^2$ with

$$c_k = \frac{B_k^2 - q^2r^2\Delta_K}{4a_k} = q^2,$$

which is then equivalent to the form $q^2x^2 - B_k xy + a_k y^2$ corresponding to the ideal $(q^2, -B_k) = (q^2, -B_k \mod_c 2q^2)$, where the subscript c designates the centered euclidean division. Finally, this ideal is reduced because $|-B_k \mod_c 2q^2| < q^2 < \sqrt{\Delta_K q^2 r^2/4}$.

□

Theorem A − 3. *Let* Δ_K *be a fundamental negative discriminant, different from* -3 *and* -4 *and* q *be an odd prime conductor. Let* r *be an odd integer, prime to both* q *and* Δ_K *such that* $r > 2q/\sqrt{|\Delta_K|}$. *Given a class of* $\ker\bar{\varphi}_q$ *and* $\mathfrak{h}$ *a representative in* $I(\mathcal{O}_{\Delta_q}, qr)$, *then the class*

$$[\mathfrak{h} \cap \mathcal{O}_{\Delta_q r^2}]^{\phi_{\Delta_K}(r)}$$

is trivial if the order of $[\mathfrak{h}]$ *divides* $\phi_{\Delta_K}(r)$ *and has a reduced element of norm* q^2 *otherwise.*

Proof. Let $\mathfrak{h} \in I(\mathcal{O}_{\Delta_q}, q)$ be a representative of a class of $\ker\bar{\varphi}_q$. Let $\alpha \in \mathcal{O}_{\Delta_K}$ such that $\mathfrak{h}\mathcal{O}_{\Delta_K} = \alpha\,\mathcal{O}_{\Delta_K}$. Let us remark first that $\mathfrak{h} \cap \mathcal{O}_{\Delta_q r^2}$, which is an $\mathcal{O}_{\Delta_q r^2}$-ideal, is equal to $\alpha\mathcal{O}_{\Delta_K} \cap \mathcal{O}_{\Delta_q r^2}$. Therefore $[\mathfrak{h} \cap \mathcal{O}_{\Delta_q r^2}]$ is in $\ker\bar{\varphi}_{qr}$. By the isomorphisms of Lemma A − 1, $[\mathfrak{h}] \in \ker\bar{\varphi}_q$ corresponds to $\big([\alpha] \pmod{q}\big) \in \big(\mathcal{O}_{\Delta_K}/q\mathcal{O}_{\Delta_K}\big)^\times / \big(\mathbf{Z}/q\mathbf{Z}\big)^\times$ and $[\mathfrak{h} \cap \mathcal{O}_{\Delta_q r^2}]$ corresponds to $\big([\alpha] \pmod{qr}\big)$ in $\big(\mathcal{O}_{\Delta_K}/qr\mathcal{O}_{\Delta_K}\big)^\times / \big(\mathbf{Z}/qr\mathbf{Z}\big)^\times$.

Once again, we are going to use properties of quotients of $\mathcal{O}_{\Delta_K}$ to obtain some information on the kernel of $\bar{\varphi}_q$ and $\bar{\varphi}_{qr}$. Let

$$\begin{array}{cccc} s: & \big(\mathcal{O}_{\Delta_K}/q\mathcal{O}_{\Delta_K}\big)^\times / \big(\mathbf{Z}/q\mathbf{Z}\big)^\times & \longrightarrow & \big(\mathcal{O}_{\Delta_K}/qr\mathcal{O}_{\Delta_K}\big)^\times / \big(\mathbf{Z}/qr\mathbf{Z}\big)^\times \\ & [\alpha] & \longmapsto & [\alpha]^{\phi_{\Delta_K}(r)}. \end{array}$$

The map s is a well-defined morphism. Indeed, if α and β are two elements of $\mathcal{O}_{\Delta_K}$ such that $[\alpha] = [\beta]$ in $\big(\mathcal{O}_{\Delta_K}/q\mathcal{O}_{\Delta_K}\big)^\times / \big(\mathbf{Z}/q\mathbf{Z}\big)^\times$, then, in the Chinese remainder isomorphism, $[\alpha]^{\phi_{\Delta_K}(r)}$ maps to $([\alpha]^{\phi_{\Delta_K}(r)} \pmod{q}, [1] \pmod{r})$. On the other hand, the element $[\beta]^{\phi_{\Delta_K}(r)}$ maps to $([\beta]^{\phi_{\Delta_K}(r)} \pmod{q}, [1] \pmod{r})$ and therefore $s([\alpha]) = s([\beta])$. Note that the kernel of s is the subgroup of $\phi_{\Delta_K}(r)$-th roots of unity of the quotient $\big(\mathcal{O}_{\Delta_K}/q\mathcal{O}_{\Delta_K}\big)^\times / \big(\mathbf{Z}/q\mathbf{Z}\big)^\times$.

Let us define the morphism $\hat{s}$ between $\ker\bar{\varphi}_q$ and $\ker\bar{\varphi}_{qr}$ such that the following diagram commutes:

$$\begin{tikzcd}
\ker\bar{\varphi}_q \arrow[r, "\hat{s}"] & \ker\bar{\varphi}_{qr} \\
\left(\mathcal{O}_{\Delta_K}/q\mathcal{O}_{\Delta_K}\right)^\times / \left(\mathbf{Z}/q\mathbf{Z}\right)^\times \arrow[u, "\psi_q", "\wr"'] \arrow[r, "s"] \arrow[ur, phantom, "\circlearrowleft"] & \left(\mathcal{O}_{\Delta_K}/qr\mathcal{O}_{\Delta_K}\right)^\times / \left(\mathbf{Z}/qr\mathbf{Z}\right)^\times \arrow[u, "\psi_{qr}", "\wr"']
\end{tikzcd}$$

Now, we prove that $\hat{s}([\mathfrak{h}]) = [\mathfrak{h} \cap \mathcal{O}_{\Delta_q r^2}]^{\phi_{\Delta_K}(r)}$. Indeed, $\hat{s}([\mathfrak{h}]) = \hat{s} \circ \psi_q([\alpha] \pmod q)$ and by commutativity of the diagram

$$\begin{aligned}
\hat{s} \circ \psi_q([\alpha] \pmod q) &= \psi_{qr} \circ s([\alpha] \pmod q) \\
&= \psi_{qr}\Big(\big([\alpha] \pmod q\big)^{\phi_{\Delta_K}(r)}\Big) \\
&= \psi_{qr}\Big(\big([\alpha] \pmod{qr}\big)^{\phi_{\Delta_K}(r)}\Big) \\
&= \psi_{qr}\Big([\alpha] \pmod{qr}\Big)^{\phi_{\Delta_K}(r)} = [\mathfrak{h} \cap \mathcal{O}_{\Delta_q r^2}]^{\phi_{\Delta_K}(r)}.
\end{aligned}$$

By construction, $\ker\hat{s}$ is the subgroup of $\phi_{\Delta_K}(r)$-th roots of unity of $\ker\bar{\varphi}_q$ and therefore, if the order of $[\mathfrak{h}]$ divides $\phi_{\Delta_K}(r)$, then $\hat{s}([\mathfrak{h}]) = [\mathcal{O}_{\Delta_q r^2}]$. Otherwise, as the image of s is a subset of the kernel of the surjection π of Lemma A − 2, the reduced ideal of the class $\hat{s}([\mathfrak{h}])$ has norm q^2. □

Corollary A − 1. *Algorithm A.3 solves the Kernel Problem and totally breaks the NICE family of cryptosystems in cubic time in the security parameter.*

Proof. The correctness of Algorithm A.3 follows from the previous theorem: All the assumptions are verified. In particular, $r > 2q/\sqrt{|\Delta_K|}$ and $\mathfrak{h}'$ is a representative of $[\mathfrak{h}]$ in $I(\mathcal{O}_{\Delta_q}, qr)$: The ideal $\mathfrak{h}'$ is chosen prime to r' and will be also prime to q, otherwise the factorisation of Δ_q is already recovered. Now, $[\mathfrak{k}]$ is trivial if the order of $[\mathfrak{h}]$ divides $\phi_{\Delta_K}(r) = r'^{\,\delta_{r'}-1}\left(r' - \left(\frac{\Delta_K}{r'}\right)\right)$. As we suppose that the order of $[\mathfrak{h}]$ is greater than 6 (see Remark. A − 2), at most one iteration of step 3 will be done, otherwise the order of $[\mathfrak{h}]$ divides both $\phi_{\Delta_K}(3^{\delta_3})$ and $\phi_{\Delta_K}(5^{\delta_5})$, which is impossible (since their gcd is 2, 4 or 6, according to the value of the Kronecker symbols). Eventually, $\mathfrak{k}$ has norm q^2 and therefore Algorithm A.3 outputs a nontrivial factorisation of Δ_q.

The cost of the initialisation phase is essentially cubic in the security parameter. The core of the algorithm consists in applying Algorithm A.1 whose complexity is quadratic in λ, and an exponentiation whose complexity is cubic. □

Corollary A−1 implies that all the schemes for which a public element of the kernel of $\bar{\varphi}_q$ is needed are broken in polynomial time. This includes the NICE encryption scheme and its variants, notably the enhanced IND-CCA2 version (cf. [PT98, HPT99, PT00, Hüh99, BST02]), the derived signature scheme (cf. [HM00b, Hüh01]) and the undeniable signature scheme (cf. [BPT04]). Note that this result does not affect the security of the adaptation of seminal cryptosystems in imaginary quadratic fields, *i. e.*, the Diffie-Hellman key exchange of [BW88, McC89], the Rabin and RSA analogues of [HMT98] and the adaptation of Elgamal of [HJPT98].

Example.

We apply our cryptanalysis on the example of the NICE encryption scheme mentioned in [JJ00], described as follows:

$$\begin{array}{rcl} \Delta_q & = & -10011336194028467500739190370826191745653724259466 7\\ & & 4915149340539464219927955168182167600836407521987097\\ & & 2619973270184386441185324964453536572880202249818566\\ & & 5592983708546453282107912775914256762913490132215200\\ & & 222246716212360016561209 23 \end{array}$$

$$\begin{array}{rcl} a & = & 5702268770894258318168588438117558871300783180769995 1\\ & & 95092715895755173700399141486895731384747 \end{array}$$

$$\begin{array}{rcl} b & = & 3361236040582754784958586298017949110648731745605930 1\\ & & 64666819569606755029773074415823039847007 \end{array}$$

The public key consists in Δ_q and $\mathfrak{h} = (a, b)$.

The ideal $\mathfrak{h} = (a, b)$ is equivalent to the ideal $\mathfrak{h}' = (a', b')$ with norm prime to 3 with $b' = -b$ and $a' = (b^2 - \Delta_q)/4a$:

$$\begin{array}{rl} a' = & 43891898980317792308326285455049173482378605867\\ & 42403785190862097985269408138288879224220052968\\ & 10150815323915182343893632698778887397967669 \end{array}$$

$$\begin{array}{rl} b' = & -336123604058275478495858629801794911064873174 5\\ & 60593016466681956960675502977307441582303984700 7 \end{array}$$

We used the following power of 3:

$$r = 3^{83} = 3990838394187339929534246675572349035227$$

Then, in 20ms, we have computed the lift of (a', b') of norm q^2:

$$\begin{aligned}\mathfrak{f} = (\ & 53631231719770388398296099992823384509917463 2823\\ & 695735108942457748870561203659790025346332338302\\ & 277214655139356149715939077126809522499818706407\\ & 36401120729,\\ & 507261151958947963506445391580733286545183991 70\\ & 010324260439808053865626730478159167292645232706\\ & 489579615441563764090965623987919889655079184915\\ & 879970067243)\end{aligned}$$

The experiments have been done on a standard laptop running Linux with PARI/GP.

A.4. Conclusion

We totally break a large class of cryptosystems based on imaginary quadratic field arithmetic, whose main interest was the quadratic complexity of the secret operation. This polynomial time attack shows that SKEP and the kernel problem are not suited to build cryptosystems and lessen the number of public-key cryptosystems with quadratic decryption time. The adaptation of NICE recently proposed in [JSW08] in the very different setting of real quadratic fields, seems to resist to our attack.

Acknowledgement.

We warmly thank Denis Simon with whom we had helpful discussions on quadratic forms, Brigitte Vallée for her huge contribution to the analysis of the complexity of our first attack, and last not but least Andreas Enge for his conscientious reviewing of a preliminary version of our paper and for his precious comments.

Chapter B

Factoring pq^2 with Quadratic Forms: Nice Cryptanalyses

Joint work with Antoine Joux, Fabien Laguillaumie and
Phong Q. Nguyen
[CJLN09]

Abstract. We present a new algorithm based on binary quadratic forms to factor integers of the form $N = pq^2$. Its heuristic running time is exponential in the general case, but becomes polynomial when special (arithmetic) hints are available, which is exactly the case for the so-called NICE family of public-key cryptosystems based on quadratic fields introduced in the late 90s. Such cryptosystems come in two flavours, depending on whether the quadratic field is imaginary or real. Our factoring algorithm yields a general key-recovery polynomial-time attack on NICE, which works for both versions: Castagnos and Laguillaumie recently obtained a total break of *imaginary*-NICE, but their attack could not apply to *real*-NICE. Our algorithm is rather different from classical factoring algorithms: it combines Lagrange's reduction of quadratic forms with a provable variant of Coppersmith's lattice-based root finding algorithm for homogeneous polynomials. It is very efficient given either of the following arithmetic hints: the public key of *imaginary*-NICE, which provides an alternative to the CL attack; or the knowledge that the regulator of the quadratic field $\mathbf{Q}(\sqrt{p})$ is unusually small, just like in *real*-NICE.

B.1. Introduction

Many public-key cryptosystems require the hardness of factoring large integers of the special form $N = pq^2$, such as Okamoto's Esign [Oka90], Okamoto and Uchiyama's encryption [OU98], Takagi's fast RSA variants [Tak98], and the large family (surveyed in [BTV04]) of cryptosystems based on quadratic fields, which was initiated by Buchmann and Williams' key exchange [BW88], and which includes NICE[1] cryptosystems [HPT99, PT98, PT00, JSW08] (whose main feature is a quadratic decryption). These moduli are popular because they can lead to special functionalities (like homomorphic encryption) or improved efficiency (compared to RSA). And no significant weakness has been found compared to standard RSA moduli of the form $N = pq$: to the best of our knowledge, the only results on pq^2 factorisation are [PO96, Per01, BDH99]. More precisely, [PO96, Per01] obtained a linear speed-up of Lenstra's ECM, and [BDH99, Sect. 6] can factor in time $\tilde{O}(N^{1/9})$ when p and q are balanced. Furthermore, computing the "squarefree part" of an integer (that is, given $N \in \mathbf{N}$ as input, compute $(r,s) \in \mathbf{N}^2$ such that $N = r^2 s$ with s squarefree) is a classical problem in algorithmic number theory (cf. [AM94]), because it is polynomial-time equivalent to determining the ring of integers of a number field [Chi89].

However, some of these cryptosystems actually provide additional information in the public key, which may render factorisation easy. For instance, Howgrave-Graham [How01] showed that the public key of [Oka86] disclosed the secret factorisation in polynomial time, using the gcd extension of the root finding method of Coppersmith [Cop97]. Very recently, Castagnos and Laguillaumie [CL09] showed that the public key in the imaginary version [HPT99, PT98, PT00] of NICE allowed to retrieve the secret factorisation in polynomial time. And this additional information in the public key was crucial to make the complexity of decryption quadratic in *imaginary*-NICE, which was the main claimed benefit of NICE. But surprisingly, the attack of [CL09] does not work against REAL-NICE [JSW08], which is the version of NICE with real (rather than imaginary) quadratic fields, and which also offers quadratic decryption. In particular, the public key of REAL-NICE only consists of $N = pq^2$, but the prime p has special arithmetic properties.

Our Results. We present a new algorithm to factor integers of the form $N = pq^2$, based on binary quadratic forms (or equivalently, ideals of orders of quadratic number fields). In the worst case, its heuristic running time is exponential, namely $\tilde{O}(p^{1/2})$. But in the presence of special hints, it becomes heuristically polynomial. These hints are different from the usual ones of lattice-based factoring methods [Cop97, BDH99, How01] where they are a fraction of the bits of the secret prime factors. Instead, our hints are arithmetic, and correspond exactly to the situation of NICE, including both the imaginary [HPT99, PT98, PT00] and real versions [JSW08]. This gives rise to the first general key-recovery polynomial-time attack on NICE, using only the public key.

More precisely, our arithmetic hints can be either of the following two:

[1] for *New Ideal Coset Encryption*

1. The hint is an ideal equivalent to a secret ideal of norm q^2 in an order of an imaginary quadratic field of discriminant $-pq^2$: in NICE, such an ideal is disclosed by the public key. This gives an alternative attack of NICE, different from [CL09].

2. The hint is the knowledge that the *regulator* of the quadratic field $\mathbf{Q}(\sqrt{p})$ is unusually small, just like in REAL-NICE. Roughly speaking, the regulator is a real number which determines how "dense" the units of the ring of integers of the number field $\mathbf{Q}(\sqrt{p})$ are. This number is known to lie in the large interval

$$\left[\log\left(\frac{1}{2}(\sqrt{p-4}+\sqrt{p})\right), \sqrt{\frac{1}{2}p}\left(\frac{1}{2}\log p+1\right)\right].$$

But for infinitely many p (including square-free numbers of the form $p = k^2 + r$, where $p > 5$, $r|4k$ and $-k < r \leq k$, see [Deg58]), the regulator is at most polynomial in $\log p$. For these unusually small regulators, our algorithm heuristically runs in time polynomial in the bit-length of $\mathrm{N} = pq^2$, which gives the first total break of REAL-NICE [JSW08]. We stress that although such p's are easy to construct, their density is believed to be arbitrary small.

Interestingly, our algorithm is rather different from classical factoring algorithms. It is a combination of Lagrange's reduction of quadratic forms with a provable variant of Coppersmith's lattice-based root finding algorithm [Cop97] for homogeneous polynomials. In a nutshell, our factoring method first looks for a reduced binary quadratic form $f(x,y) = ax^2 + bxy + cy^2$ representing properly q^2 with small coefficients, *i.e.* there exist small coprime integers x_0 and y_0 such that $q^2 = f(x_0, y_0)$. In case i., such a quadratic form is already given. In case ii., such a quadratic form is found by a walk along the principal cycle of the class group of discriminant pq^2, using Lagrange's reduction of (indefinite) quadratic forms. Finally, the algorithm finds such small coprime integers x_0 and y_0 such that $q^2 = f(x_0, y_0)$, by using the fact that $\gcd(f(x_0, y_0), pq^2)$ is large. This discloses q^2 and therefore the factorisation of N. In both cases, the search for x_0 and y_0 is done with a new *rigorous* homogeneous bivariate variant of Coppersmith's method, which might be of independent interest: by the way, it was pointed out to us that Bernstein [Ber11] independently used a similar method in the different context of Goppa codes decoding.

Our algorithm requires "natural" bounds on the roots of reduced quadratic forms of a special shape. We are unable to prove rigorously all these bounds, which makes our algorithm heuristic (like many factoring algorithms). But we have performed many experiments supporting such bounds, and the algorithm works very well in practice.

Factorisation and Quadratic Forms. Our algorithm is based on quadratic forms, which share a long history with factoring (see [CP01]). Fermat's factoring method represents N in two intrinsically different ways by the quadratic form $x^2 + y^2$. It has been improved by Shanks with SQUFOF, whose complexity is $\tilde{O}(\mathrm{N}^{1/4})$ (see [GW08] for a detailed analysis). Like ours, this method works with the infrastructure of a class group of positive discriminant, but is different in spirit since it searches for an *ambiguous* form (after having found a square form), and does not focus on discriminants of a special shape. Schoof's factoring algorithms [Sch82] are also essentially looking for ambiguous forms. One is based on computation in class groups of complex quadratic orders and the other is close to SQUFOF since it works with real quadratic orders by computing a good approximation of the regulator to find an ambiguous form. Like SQUFOF, this algorithm does not takes advantage of working in a non-maximal order and is rather different from our algorithm. Both algorithms of [Sch82] runs in $\tilde{O}(N^{1/5})$ under the generalised Riemann hypothesis. McKee's method [McK99] is a speedup of Fermat's algorithm (and was presented as an alternative to SQUFOF) with a heuristic complexity of $\tilde{O}(N^{1/4})$ instead of $\tilde{O}(N^{1/2})$.

SQUFOF and other exponential methods are often used to factor small numbers (say 50 to 100 bits), for instance in the post-sieving phase of the Number Field Sieve algorithm. Some interesting experimental comparisons can be found in [Mil07]. Note that the currently fastest rigorous *deterministic* algorithm actually has exponential complexity: it is based on a polynomial evaluation method (for a polynomial of the form $x(x-1)\cdots(x-B+1)$ for some bound B) and its best variant is described in [BGS07]. Finally, all sieve factoring algorithms are somewhat related to quadratic forms, since their goal is to find random pairs (x, y) of integers such that $x^2 \equiv y^2 \mod N$. However, these algorithms factor generic numbers and have a subexponential complexity.

Road Map. The rest of the paper is organised as follows. The first section recalls facts on quadratic fields and quadratic forms, and present our heuristic supported by experiments. The next section describes the homogeneous Coppersmith method and the following exhibits our main result: the factoring algorithm. The last section consists of the two cryptanalyses of cryptosystems based on real quadratic fields (REAL-NICE) and on imaginary quadratic fields (NICE).

B.2. Background on Quadratic Fields and Quadratic Forms

B.2.1. Quadratic Fields

Let $D \neq 0, 1$ be a squarefree integer and consider the quadratic number field $K = \mathbf{Q}(\sqrt{D})$. If $D < 0$ (resp. $D > 0$), K is called an *imaginary* (resp. a *real*) quadratic field. The *fundamental discriminant* Δ_K of K is defined as $\Delta_K = D$ if $D \equiv 1 \pmod 4$ and $\Delta_K = 4D$ otherwise. An *order* $\mathcal{O}$ in K is a subset of K such that $\mathcal{O}$ is a subring of K containing 1 and $\mathcal{O}$ is a free $\mathbf{Z}$-module of rank 2. The ring $\mathcal{O}_{\Delta_K}$ of algebraic integers in K is the *maximal* order of K. It can be written as $\mathbf{Z} + \omega_K \mathbf{Z}$, where $\omega_K = \frac{1}{2}(\Delta_K + \sqrt{\Delta_K})$. If we set $f = [\mathcal{O}_{\Delta_K} : \mathcal{O}]$ the *finite* index of any order $\mathcal{O}$ in $\mathcal{O}_{\Delta_K}$, then $\mathcal{O} = \mathbf{Z} + f\omega_K\mathbf{Z}$. The integer f is called the *conductor* of $\mathcal{O}$. The discriminant of $\mathcal{O}$ is then $\Delta_f = f^2\Delta_K$. Now, let $\mathcal{O}_\Delta$ be an order of discriminant Δ and $\mathfrak{a}$ be a nonzero ideal of $\mathcal{O}_\Delta$, its norm is $N(\mathfrak{a}) = |\mathcal{O}_\Delta/\mathfrak{a}|$. A *fractional* ideal is a subset $\mathfrak{a} \subset K$ such that $d\mathfrak{a}$ is an ideal of $\mathcal{O}_\Delta$ for $d \in \mathbf{N}$. A fractional ideal $\mathfrak{a}$ is said to be *invertible* if there exists an another fractional ideal $\mathfrak{b}$ such that $\mathfrak{a}\mathfrak{b} = \mathcal{O}_\Delta$. The *ideal class group* of $\mathcal{O}_\Delta$ is $C(\mathcal{O}_\Delta) = I(\mathcal{O}_\Delta)/P(\mathcal{O}_\Delta)$, where $I(\mathcal{O}_\Delta)$ is the group of invertible fractional ideals of $\mathcal{O}_\Delta$ and $P(\mathcal{O}_\Delta)$ the subgroup consisting of principal ideals.

Its cardinality is the *class number* of $\mathcal{O}_\Delta$ denoted by $h(\mathcal{O}_\Delta)$. A nonzero ideal $\mathfrak{a}$ of $\mathcal{O}_\Delta$, $\mathfrak{a}$ is said to be *prime to* f if $\mathfrak{a} + f\mathcal{O}_\Delta = \mathcal{O}_\Delta$. We denote by $\mathrm{I}(\mathcal{O}_\Delta, f)$ the subgroup of $\mathrm{I}(\mathcal{O}_\Delta)$ of ideals prime to f. The group $\mathcal{O}_\Delta^\star$ of units in $\mathcal{O}_\Delta$ is equal to $\{\pm 1\}$ for all $\Delta < 0$, except when Δ is equal to -3 and -4 ($\mathcal{O}_{-3}^\star$ and $\mathcal{O}_{-4}^\star$ are respectively the group of sixth and fourth roots of unity). When $\Delta > 0$, then $\mathcal{O}_\Delta^\star = \langle -1, \varepsilon_\Delta \rangle$ where $\varepsilon_\Delta > 0$ is called the *fundamental unit*. The real number $\mathrm{R}_\Delta = \log(\varepsilon_\Delta)$ is the *regulator* of $\mathcal{O}_\Delta$. The following important bounds on the regulator of a real quadratic field can be found in [JLW95]:

$$\log\left(\frac{1}{2}(\sqrt{\Delta - 4} + \sqrt{\Delta})\right) \leq \mathrm{R}_\Delta < \sqrt{\frac{1}{2}\Delta}\left(\frac{1}{2}\log\Delta + 1\right). \tag{B.1}$$

The lower bound is reached *infinitely often*, for instance with $\Delta = x^2 + 4$ with $2 \nmid x$. Finally, this last proposition is the heart of both NICE and REAL-NICE.

Proposition B – 1 ([Cox99, Proposition 7.20] [Wei04, Theorem 2.16]). *Let $\mathcal{O}_{\Delta_f}$ be an order of conductor f in a quadratic field* K.

1. *If $\mathfrak{A}$ is an $\mathcal{O}_{\Delta_\mathrm{K}}$-ideal prime to f, then $\mathfrak{A} \cap \mathcal{O}_{\Delta_f}$ is an $\mathcal{O}_{\Delta_f}$-ideal prime to f of the same norm.*
2. *If $\mathfrak{a}$ is an $\mathcal{O}_{\Delta_f}$-ideal prime to f, then $\mathfrak{a}\mathcal{O}_{\Delta_\mathrm{K}}$ is an $\mathcal{O}_{\Delta_\mathrm{K}}$-ideal prime to f of the same norm.*
3. *The map $\varphi_f : \mathrm{I}(\mathcal{O}_{\Delta_f}, f) \to \mathrm{I}(\mathcal{O}_{\Delta_\mathrm{K}}, f)$, $\mathfrak{a} \mapsto \mathfrak{a}\mathcal{O}_{\Delta_\mathrm{K}}$ is an isomorphism.*

The map φ_f from Proposition B – 1 induces a surjection $\bar{\varphi}_f : \mathrm{C}(\mathcal{O}_{\Delta_f}) \twoheadrightarrow \mathrm{C}(\mathcal{O}_{\Delta_\mathrm{K}})$ which can be efficiently computed (see [PT00]). In our settings, we will use a prime conductor $f = q$ and consider $\Delta_q = q^2\Delta_\mathrm{K}$, for a fundamental discriminant Δ_K. In that case, the order of the kernel of $\bar{\varphi}_q$ is given by the classical *analytic class number formula* (see for instance [BV07])

$$\frac{h(\mathcal{O}_{\Delta_q})}{h(\mathcal{O}_{\Delta_\mathrm{K}})} = \begin{cases} q - (\Delta_\mathrm{K}/q) & \text{if } \Delta_k < -4, \\ (q - (\Delta_\mathrm{K}/q))\mathrm{R}_{\Delta_\mathrm{K}}/\mathrm{R}_{\Delta_q} & \text{if } \Delta_k > 0. \end{cases} \tag{B.2}$$

Note that in the case of real quadratic fields, $\epsilon_{\Delta_q} = \epsilon_{\Delta_\mathrm{K}}^t$ for a positive integer t, hence $\mathrm{R}_{\Delta_q}/\mathrm{R}_{\Delta_\mathrm{K}} = t$ and $t \mid (q - (\Delta_\mathrm{K}/q))$.

B.2.2. Representation of the Classes

Working with ideals modulo the equivalence relation of the class group is essentially equivalent to work with binary quadratic forms modulo $\mathrm{SL}_2(\mathbf{Z})$ (cf. [Coh00, Section 5.2]). Moreover, quadratic forms are more suited to an algorithmic point of view. Every ideal $\mathfrak{a}$ of $\mathcal{O}_\Delta$ can be written as $\mathfrak{a} = m\left(a\mathbf{Z} + \frac{-b+\sqrt{\Delta}}{2}\mathbf{Z}\right)$ with $m \in \mathbf{Z}$, $a \in \mathbf{N}$ and $b \in \mathbf{Z}$ such that $b^2 \equiv \Delta \pmod{4a}$. In the remainder, we will only consider *primitive* integral ideals, which are those with $m = 1$. This notation also represents the binary quadratic form $ax^2 + bxy + cy^2$ with $b^2 - 4ac = \Delta$. This representation of the ideal is unique if the form is normal (see below). We recall here some facts about binary quadratic forms.

Definition B – 1. A *binary quadratic form* f is a degree 2 homogeneous polynomial $f(x, y) = ax^2 + bxy + cy^2$ where a, b and c are integers, and is denoted by $[a, b, c]$. The discriminant of the form is $\Delta = b^2 - 4ac$. If $a > 0$ and $\Delta < 0$, the form is called *definite positive* and *indefinite* if $\Delta > 0$.

Let $\mathrm{M} \in \mathsf{SL}_2(\mathbf{Z})$ with $\mathrm{M} = \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix}$, and $f = [a, b, c]$, a binary quadratic form, then $f.\mathrm{M}$ is the equivalent binary quadratic form $f(\alpha x + \beta y, \gamma x + \delta y)$.

Definite Positive Forms.

Let us first define the crucial notion of *reduction*.

Definition B – 2. The form $f = [a, b, c]$ is called *normal* if $-a < b \leq a$. It is called *reduced* if it is normal, $a \leq c$, and if $b \geq 0$ for $a = c$.

The procedure which transforms a form $f = [a, b, c]$ into a normal one consists in setting s such that $b + 2sa$ belongs to the right interval (see [BV07, (5.4)]) and producing the form $[a, b + 2sa, as^2 + bs + c]$. Once a form $f = [a, b, c]$ is normalised, a *reduction step* consists in normalising the form $[c, -b, a]$. We denote this form by $\rho(f)$ and by Rho a corresponding algorithm. The reduction then consists in normalising f, and then iteratively replacing f by $\rho(f)$ until f is reduced. The time complexity of this (Lagrange-Gauß) algorithm is quadratic (see [BV07]). It returns a reduced form g which is equivalent to f modulo $\mathsf{SL}_2(\mathbf{Z})$. We will call *matrix of the reduction*, the matrix M such that $g = f.\mathrm{M}$. The reduction procedure yields a *uniquely* determined reduced form in the class modulo $\mathsf{SL}_2(\mathbf{Z})$.

Indefinite Forms.

Our main result will deal with forms of positive discriminant. Here is the definition of a reduced indefinite form.

Definition B – 3. The form $f = [a, b, c]$ of positive discriminant Δ is called *reduced* if $\left|\sqrt{\Delta} - 2|a|\right| < b < \sqrt{\Delta}$ and *normal* if $-|a| < b \leq |a|$ for $|a| \geq \sqrt{\Delta}$, and $\sqrt{\Delta} - 2|a| < b < \sqrt{\Delta}$ for $|a| < \sqrt{\Delta}$.

The reduction process is similar to the definite positive case. The time complexity of the algorithm is still quadratic (see [BV07, Theorem 6.6.4]). It returns a reduced form g which is equivalent to f modulo $\mathsf{SL}_2(\mathbf{Z})$. The main difference with forms of negative discriminant is that there will in general not exist a unique reduced form per class, but several organised in a cycle structure *i. e.*, when f has been reduced then subsequent applications of ρ give other reduced forms.

Definition B – 4. Let f be an indefinite binary quadratic form, the *cycle* of f is the sequence $(\rho^i(g))_{i\in\mathbf{Z}}$ where g is a reduced form which is equivalent to f.

From Theorem 6.10.3 from [BV07], the cycle of f consists of all reduced forms in the equivalence class of f. Actually, the complete cycle is obtained by a finite number of application of ρ as the process is periodic. It has been shown in [BTW95] that the period length ℓ of the sequence of reduced forms in each class of a class group of discriminant Δ satisfies $\frac{R_\Delta}{\log\Delta} \leq \ell \leq \frac{2R_\Delta}{\log 2}+1$.

Our factoring algorithm will actually take place in the principal equivalence class. The following definition exhibits the *principal form* of discriminant Δ.

Definition B − 5. The reduced form $[1, \lfloor\sqrt{\Delta}\rfloor, (\lfloor\sqrt{\Delta}\rfloor^2-\Delta)/4]$ of discriminant Δ is called the *principal form* of discriminant Δ, and will be denoted $\mathbf{1}_\Delta$.

B.2.3. Reduction of the Forms $[q^2, kq, (k^2 \pm p)/4]$ and Heuristics

In this subsection, p and q are two distinct primes of the same bit-size λ and $p \equiv 1 \mod 4$ (resp. $p \equiv 3 \mod 4$) when we deal with positive (resp. negative) discriminant. Our goal is to factor the numbers pq^2 with the special normalised quadratic forms $[q^2, kq, (k^2+p)/4]$ or $[q^2, kq, (k^2-p)/4]$, depending whether we work with a negative discriminant $\Delta_q = -pq^2$ or with a positive one $\Delta_q = pq^2$. If p and q have the same size, these forms are clearly not reduced neither in the imaginary setting nor in real one. But as we shall see, we can find the reduced forms which correspond to the output of the reduction algorithm applied on these forms.

Suppose that we know a form $\hat{f}_k$, either definite positive or indefinite, which is the reduction of a form $f_k = [q^2, kq, (k^2 \pm p)/4]$ where k is an integer. Then $\hat{f}_k$ represents the number q^2. More precisely, if $M_k = \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} \in \mathsf{SL}_2(\mathbf{Z})$ is the matrix such that $\hat{f}_k = f_k.M_k$, then $\hat{f}_k.M_k^{-1} = f_k$ and $q^2 = f_k(1,0) = \hat{f}_k(\delta, -\gamma)$. In Section B.3, we will see that provided they are relatively small compared to Δ_q, the values δ and $-\gamma$ can be found in polynomial time with a new variant of Coppersmith method. Our factoring algorithm can be sketched as follows: find such a form $\hat{f}_k$ and if the coefficients of M_k are sufficiently small, retrieve δ and $-\gamma$ and the non-trivial factor q^2 of Δ_q. In this paragraph, we give some heuristics on the size of such a matrix M_k and discuss their relevance. If M is a matrix we denote by $|M|$ the max norm, *i. e.*, the maximal coefficient of M in absolute value.

In the imaginary case, it is showed in the proof of Theorem A − 2 of [CL09] that the forms f_k belong to different classes of the kernel of the map $\bar{\varphi}_q$, depending on k, so the reduced equivalent forms $\hat{f}_k$ are the unique reduced elements of the classes of the kernel. To prove the correctness of our attack on NICE, we need the following heuristic (indeed, the root finding algorithm of Section B.3 recovers roots up to $|\Delta_q|^{1/9}$):

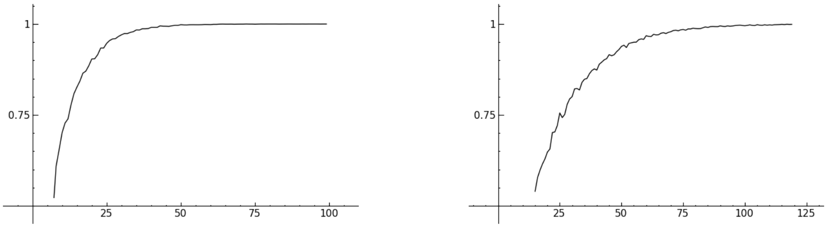
Heuristic B.1 (Imaginary case)**.** Given a reduced element $\hat{f}_k$ of a nontrivial class of $\ker\bar{\varphi}_q$, the matrix of reduction M_k is such that $|M_k| < |\Delta_q|^{1/9}$ with probability asymptotically close to 1.

From Lemma 5.6.1 of [BV07], $|\mathrm{M}_k| < 2\max\{q^2, (k^2+p)/4\}/\sqrt{pq^2}$. As f_k is normalised, $|k| \leq q$ and $|\mathrm{M}_k| < 2q/\sqrt{p} \approx |\Delta_q|^{1/6}$. Note that we cannot reach such a bound with our root finding algorithm. Experimentally, for random k, $|\mathrm{M}_k|$ can be much smaller. For example, if the bit-size λ of p and q equals 100, the mean value of $|\mathrm{M}_k|$ is around $|\Delta_q|^{1/11.7}$. Our heuristic can be explained as follows. A well-known heuristic in the reduction of positive definite quadratic forms (or equivalently, two-dimensional lattices) is that if $[a, b, c]$ is a reduced quadratic form of discriminant Δ, then a and c should be close to $\sqrt{\Delta}$. This cannot hold for all reduced forms, but it can be proved to hold for an overwhelming majority of reduced forms. Applied to $\hat{f}_k = [a, b, c]$, this means that we expect a and c to be close to $|\Delta_q|^{1/2}$. Now, recall that $q^2 = \hat{f}_k(\delta, -\gamma) = a\delta^2 - b\delta\gamma + c\gamma^2$, which leads to δ and γ close to $\sqrt{q^2/a} = q/\sqrt{a} \approx q/|\Delta_q|^{1/4} \approx |\Delta_q|^{1/12}$. Thus, we expect that $|\mathrm{M}_k| \leq |\Delta_q|^{1/12}$. And this explains why we obtained experimentally the bound $|\Delta_q|^{1/11.7}$. Figure B.1(a) shows a curve obtained by experimentation, which gives the probability that $|\mathrm{M}_k| < |\Delta_q|^{1/9}$ for random k, in function of λ. This curve also supports our heuristic.

In the real case, we prove in the following theorem that $\mathrm{R}_{\Delta_q}/\mathrm{R}_{\Delta_\mathrm{K}}$ forms f_k are principal and we exhibit the generators of the corresponding primitive ideals.

Theorem B − 1. *Let Δ_K be a fundamental positive discriminant, $\Delta_q = \Delta_\mathrm{K} q^2$ where q is an odd prime conductor. Let $\varepsilon_{\Delta_\mathrm{K}}$ (resp. ε_{Δ_q}) be the fundamental unit of $\mathcal{O}_{\Delta_\mathrm{K}}$ (resp. $\mathcal{O}_{\Delta_q}$) and t such that $\varepsilon_{\Delta_\mathrm{K}}^t = \varepsilon_{\Delta_q}$. Then the principal ideals of $\mathcal{O}_{\Delta_q}$ generated by $q\varepsilon_{\Delta_\mathrm{K}}^i$ correspond to quadratic forms $f_{k(i)} = [q^2, k(i)q, (k(i)^2 - p)/4]$ with $i \in \{1, \dots, t-1\}$ and $k(i)$ is an integer defined modulo $2q$ computable from $\varepsilon_{\Delta_\mathrm{K}}^i \mod q$.*

Proof. Let $\alpha_i = q\varepsilon_{\Delta_\mathrm{K}}^i$ with $i \in \{1, \dots, t-1\}$. Following the proof of [BTW95, Proposition 2.9], we detail here the computation of $\mathfrak{a}_i = \alpha_i\mathcal{O}_{\Delta_q}$. Let x_i and y_i be two integers such that $\varepsilon_{\Delta_\mathrm{K}}^i = x_i + y_i\omega_\mathrm{K}$. Then $\alpha_i = qx_i + y_iq\Delta_\mathrm{K}(1-q)/2 + y_i\frac{1}{2}(\Delta_q + \sqrt{\Delta_q})$, and α_i is an element of $\mathcal{O}_{\Delta_q}$. Let m_i, a_i and b_i be three integers such that $\mathfrak{a}_i = m_i\left(a_i\mathbf{Z} + \frac{-b_i+\sqrt{\Delta_q}}{2}\right)$. As mentioned in the proof of [BTW95, Proposition 2.9], m_i is the smallest positive coefficient of $\sqrt{\Delta_q}/2$ in $\mathfrak{a}_i$. As $\mathcal{O}_{\Delta_q}$ is equal to $\mathbf{Z} + (\Delta_q + \sqrt{\Delta_q})/2\mathbf{Z}$, $\alpha_i\mathcal{O}_{\Delta_q}$ is generated by α_i and $\alpha_i(\Delta_q + \sqrt{\Delta_q})/2$ as a $\mathbf{Z}$-module. So a simple calculation gives that $m_i = \gcd(y_i, q(x_i + y_i\Delta_\mathrm{K}/2))$. As $\varepsilon_{\Delta_\mathrm{K}}^i$ is not an element of $\mathcal{O}_{\Delta_q}$, we have $\gcd(y_i, q) = 1$ so $m_i = \gcd(y_i, x_i + y_i\Delta_\mathrm{K}/2)$. The same calculation to find m_i' for the ideal $\varepsilon_{\Delta_\mathrm{K}}^i\mathcal{O}_{\Delta_\mathrm{K}}$ reveals that $m_i = m_i'$. As $\varepsilon_{\Delta_\mathrm{K}}^i\mathcal{O}_{\Delta_\mathrm{K}} = \mathcal{O}_{\Delta_\mathrm{K}}$ we must have $m_i' = 1$. Now, $\mathrm{N}(\mathfrak{a}_i) = |\mathrm{N}(\alpha_i)| = q^2$ and $\mathrm{N}(\mathfrak{a}_i) = m_i^2 a_i = a_i$ and therefore $a_i = q^2$. Let us now find b_i. Note that b_i is defined modulo $2a_i$. Since $\alpha_i \in \alpha_i\mathcal{O}_{\Delta_q}$, there exist μ_i and ν_i such that $\alpha_i = a_i\mu_i + (-b_i + \sqrt{\Delta_q})/2\nu_i$. By identification in the basis $(1, \sqrt{\Delta_q})$, $\nu_k = 1$ and by a multiplication by 2, we obtain $2qx_i + qy_i\Delta_\mathrm{K} \equiv -b_iy_i \pmod{2a_i}$. As $b_i \equiv \Delta_q \pmod 2$, we only have to determine b_i modulo q^2. As y_i is prime to q, we have $b_i \equiv k(i)q \pmod{q^2}$ with $k(i) \equiv -2x_i/y_i - \Delta_\mathrm{K} \pmod q$. Finally, as we must have $-a_i < b \leq a_i$ if $a_i > \sqrt{\Delta_q}$ and else $\sqrt{\Delta_q} - 2a_i < b < \sqrt{\Delta_q}$, $k(i)$ is



(a) Imaginary case

(b) Real case

Figure B.1: Probability that $|M_k| < |\Delta_q|^{1/9}$ in function of the bit-size λ of p and q

the unique integer with $k(i) \equiv \Delta_q \pmod 2$ and $k(i) \equiv -2x_i/y_i - \Delta_K \pmod q$, such that $b = k(i)q$ satisfies that inequalities. Eventually, the principal ideal of $\mathcal{O}_{\Delta_q}$ generated by $q\varepsilon^i_{\Delta_K}$ corresponds to the form $[q^2, k(i)q, c_i]$ with $c_i = (b_i^2 - \Delta_q)/(4a_i) = (k(i)^2 - \Delta_K)/4$. □

From this theorem, we see that if we go across the cycle of principal forms, then we will find reduced forms $\hat{f}_k$. To analyse the complexity of our factoring algorithm, we have to know the distribution of these forms on the cycle. An appropriate tool is the Shanks distance d (see [BV07, Definition 10.1.4]) which is close to the number of iterations of Rho between two forms. One has $d(\mathbf{1}_{\Delta_q}, f_{k(i)}) = i\mathrm{R}_{\Delta_K}$. From Lemma 10.1.8 of [BV07], $|d(\hat{f}_{k(i)}, f_{k(i)})| < \log q$, for all $i = 1, 2, \ldots, t-1$. Let j be the smallest integer such that $0 < j\mathrm{R}_{\Delta_K} - 2\log q$, then as $j\mathrm{R}_{\Delta_K} = d(f_{k(i)}, f_{k(i+j)}) = d(f_{k(i)}, \hat{f}_{k(i)}) + d(\hat{f}_{k(i)}, \hat{f}_{k(i+j)}) + d(\hat{f}_{k(i+j)}, f_{k(i+j)})$, from the triangle inequality, one has $j\mathrm{R}_{\Delta_K} < 2\log(q) + |d(\hat{f}_{k(i)}, \hat{f}_{k(i+j)})|$. So, $|d(\hat{f}_{k(i)}, \hat{f}_{k(i+j)})| > j\mathrm{R}_{\Delta_K} - 2\log q > 0$. This inequality proves that $f_{k(i)}$ and $f_{k(i+j)}$ do not reduce to the same form. Experiments actually show that asymptotically, $|d(\hat{f}_{k(i)}, f_{k(i)})|$ is very small on average (smaller than 1). As a consequence, as pictured in figure B.2, $d(\mathbf{1}_{\Delta_q}, \hat{f}_{k(i)}) \approx i\mathrm{R}_{\Delta_K}$.

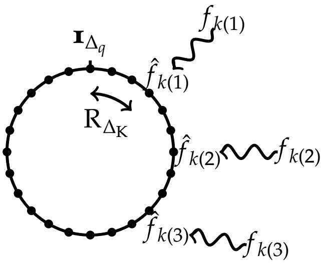


Figure B.2: Repartition of the forms $\hat{f}_{k(i)}$ along the principal cycle

Moreover, as in the imaginary case, experiments show that asymptotically the probability that the norm of the matrices of reduction, $|\mathrm{M}_k|$ is smaller than $\Delta_q^{1/9}$ is close to 1 (see figure B.1(b)). This leads to the following heuristic.

Heuristic B.2 (Real case)**.** From the principal form $\mathbf{1}_{\Delta_q}$, a reduced form $\hat{f}_k$ such that the matrix of the reduction, M_k, satisfy $|\mathrm{M}_k| < \Delta_q^{1/9}$, can be found in $\mathcal{O}(\mathrm{R}_{\Delta_\mathrm{K}})$ successive applications of Rho.

We did also some experiments to investigate the case where the bit-sizes of p and q are unbalanced. In particular when the size of q grows, the norm of the matrix of reduction becomes larger. For example, for a 100-bit p and a 200-bit q (resp. a 300-bit q), more than 95% (resp. 90%) of the $\hat{f}_k$ have a matrix M_k with $|\mathrm{M}_k| < \Delta_q^{1/6.25}$ (resp. $|\mathrm{M}_k| < \Delta_q^{1/5.44}$).

B.3. A Rigorous Homogeneous Variant of Coppersmith's Root Finding Method

Our factoring algorithm searches many times for small modular roots of degree two homogeneous polynomials and the most popular technique to find them is based on Coppersmith's method (see [Cop97] or May's survey [May10]). Our problem is the following: Given $f(x,y) = x^2 + bxy + cy^2$ a (monic) binary quadratic form and $\mathrm{N} = pq^2$ an integer of unknown factorisation, find $(x_0, y_0) \in \mathbf{Z}^2$ such that $f(x_0, y_0) \equiv 0 \pmod{q^2}$, while $|x_0|, |y_0| \leq \mathrm{M}$, where $\mathrm{M} \in \mathbf{N}$. The usual technique for this kind of problems is only heuristic, since it is the gcd extension of bivariate congruences. Moreover, precise bounds cannot be found in the litterature. Fortunately, because our polynomial is homogeneous, we will actually be able to prove the method. This homogenous variant is quite similar to the one-variable standard Coppersmith method, but is indeed even simpler to describe and more efficient since there is no need to balance coefficients. We denote as $\|\cdot\|$ the usual Euclidean norm for polynomials. The main tool to solve this problem is given by the following variant of the widespread elementary Howgrave-Graham's lemma [How97].

Lemma B − 1. *Let* $\mathrm{B} \in \mathbf{N}$ *and* $g(x,y) \in \mathbf{Z}[x,y]$ *be a homogeneous polynomial of total degree* δ. *Let* $\mathrm{M} > 0$ *be a real number and suppose that* $\|g(x,y)\| < \frac{\mathrm{B}}{\sqrt{\delta+1}\mathrm{M}^\delta}$ *then for all* $x_0, y_0 \in \mathbf{Z}$ *such that* $g(x_0,y_0) \equiv 0 \pmod{\mathrm{B}}$ *and* $|x_0|, |y_0| \leq \mathrm{M}$, $g(x_0, y_0) = 0$.

Proof. Let $g(x,y) = \sum_{i=0}^{\delta} g_i x^i y^{\delta-i}$ where some g_is might be zero. We have

$$\begin{aligned} |g(x_0,y_0)| &\leq \textstyle\sum_{i=0}^{\delta} |g_i||x_0^i y_0^{\delta-i}| \leq \mathrm{M}^\delta \sum_{i=0}^{\delta} |g_i| \\ &\leq \mathrm{M}^\delta \sqrt{\delta+1}\|g(x,y)\| < \mathrm{B} \end{aligned}$$

and therefore $g(x_0, y_0) = 0$. □

The trick is then to find only one small enough bivariate homogeneous polynomial satisfying the conditions of this lemma and to extract the rational root of the corresponding univariate polynomial with standard techniques. On the contrary, the original Howgrave-Graham's lemma suggests to look for two polynomials of small norm having (x_0, y_0) as integral root, and to recover it *via* elimination theory. The usual way to obtain these polynomials is to form a lattice spanned by a special family of polynomials, and to use the LLL algorithm (cf. [LLL82]) to obtain the two "small" polynomials. Unfortunately, this reduction does not guarantee that these polynomials will be algebraically independent, and the elimination can then lead to a trivial relation. Consequently, this bivariate approach is heuristic. Fortunately, for homogeneous polynomials, we can take another approach by using Lemma B−1 and then considering a univariate polynomial with a rational root. This makes the method rigorous and slightly simpler since we need a bound on $\|g(x,y)\|$ and not on $\|g(x\mathrm{X}, y\mathrm{Y})\|$ if X and Y are bounds on the roots and therefore the resulting lattice has smaller determinant than in the classical bivariate approach.

To evaluate the maximum of the bound we can obtain, we need the size of the first vector provided by LLL which is given by:

Lemma B−2 (LLL). *Let* L *be a full-rank lattice in* $\mathbf{Z}^d$ *spanned by an integer basis* $\mathscr{B} = \{b_1, \ldots, b_d\}$. *The LLL algorithm, given* $\mathscr{B}$ *as input, will output a non-zero vector* $u \in \mathrm{L}$ *satisfying* $\|u\| \leq 2^{(d-1)/4} \det(\mathrm{L})^{1/d}$ *in time* $\mathrm{O}(d^6 \log^3(\max \|b_i\|))$.

We will now prove the following general result regarding the modular roots of bivariate homogeneous polynomials which can be of independent interest.

Theorem B−2. *Let* $f(x,y) \in \mathbf{Z}[x,y]$ *be a homogeneous polynomial of degree* δ *satisfying* $f(x,0) = x^\delta$, N *be a non-zero integer and* α *be a rational number in* $[0,1]$, *then one can retrieve in polynomial time in* $\log \mathrm{N}$, δ *and the bit-size of* α, *all the rationals* x_0/y_0, *where* x_0 *and* y_0 *are integers such that* $\gcd(f(x_0,y_0),\mathrm{N}) \geq \mathrm{N}^\alpha$ *and* $|x_0|, |y_0| \leq \mathrm{N}^{\alpha^2/(2\delta)}$.

Proof. Let b be a divisor of N for which their exists $(x_0, y_0) \in \mathbf{Z}^2$ such that

$$b = \gcd(f(x_0,y_0),\mathrm{N}) \geq \mathrm{N}^\alpha.$$

We define some integral parameters (to be specified later) m, t and t' with $t = m + t'$ and construct a family of $\delta t + 1$ homogeneous polynomials g and h of degree δt such that (x_0, y_0) is a common root modulo b^m. More precisely, we consider the following polynomials

$$\begin{cases} g_{i,j}(x,y) &= x^j y^{\delta(t-i)-j} f^i \mathrm{N}^{m-i} \quad \text{for } i = 0, \ldots, m-1, j = 0, \ldots, \delta - 1 \\ h_i(x,y) &= x^i y^{\delta t' - i} f^m \quad \text{for } i = 0, \ldots, \delta t'. \end{cases}$$

We build the triangular matrix L of dimension $\delta t + 1$, containing the coefficients of the polynomials $g_{i,j}$ and h_i. We will apply LLL to the lattice spanned by the rows of L. The columns correspond to the coefficients of the monomials $y^{\delta t}, xy^{\delta t-1}, \ldots, x^{\delta t-1}y, x^{\delta t}$. Let $\beta \in [0,1]$ such that $\mathrm{M} = \mathrm{N}^\beta$. The product of the diagonal elements gives $\det(\mathrm{L}) =$

$N^{\delta m(m+1)/2}$. If we omit the quantities that do not depend on N, to satisfy the inequality of Lemma B−1 with the root bound M, the LLL bound from Lemma B−2 implies that we must have

$$\delta m(m+1)/2 \leq (\delta t+1)(\alpha m - \delta t \beta) \tag{B.3}$$

and if we set λ such that $t = \lambda m$, this gives asymptotically $\beta \leq \frac{\alpha}{\delta\lambda} - \frac{1}{2\delta\lambda^2}$, which is maximal when $\lambda = \frac{1}{\alpha}$, and in this case, $\beta_{\max} = \alpha^2/(2\delta)$. The vector output by LLL gives a homogeneous polynomial $\tilde{f}(x,y)$ such that $\tilde{f}(x_0,y_0) = 0$ thanks to Lemma B−1. Let $r = x/y$, any rational root of the form x_0/y_0 can be found by extracting the rational roots of $\tilde{f}'(r) = 1/y^{\delta t}\tilde{f}(x,y)$ with classical methods. □

For the case we are most interested in, $\delta = 2$, $N = pq^2$ with p and q of the same size, *i. e.*, $\alpha = 2/3$ then $\lambda = 3/2$ and we can asymptotically get roots up to N^β with $\beta = \frac{1}{9}$. If we take $m = 4$ and $t = 6$, *i. e.*, we work with a lattice of dimension 13, we get from (B.3) that $\beta \approx \frac{1}{10.63}$ and with a 31-dimensional lattice ($m = 10$ and $t = 15$), $\beta \approx \frac{1}{9.62}$. If the size of q grows compared to p, *i. e.*, α increases towards 1, then β increases towards 1/4. For example, if q is two times larger than p, *i. e.*, $\alpha = 4/5$ then $\beta = 1/6.25$. For $\alpha = 6/7$, we get $\beta \approx 1/5.44$.

In the following, we will call HomogeneousCoppersmith the algorithm which implements this method. It takes as input an integer $N = pq^2$ and a binary quadratic form $[a,b,c]$, from which we deduce the unitary polynomial $x^2 + b'xy + c'y^2$, by dividing both b and c by a modulo N, and the parameters m and t. In fact, this method will only disclose proper representations of q^2, those for which x and y are coprime, but we note that f_k properly represents q^2, and therefore so does our form $[a,b,c]$.

The case $\alpha = 1$ of Theorem 2 can already be found in Joux's book [Jou09] and we mention that a similar technique has already been independently investigated by Bernstein in [Ber11].

B.4. A $\tilde{O}(p^{1/2})$-Deterministic Algorithm for factoring pq^2

We detail our new quadratic form-based factoring algorithm for numbers of the form pq^2. In this section, p and q will be of same bit-size, and $p \equiv 1 \pmod 4$.

B.4.1. The Algorithm

Roughly speaking, if $\Delta_q = N = pq^2$, our factoring algorithm, depicted in Fig. B.3, exploits the fact that the non-reduced forms $f_k = [q^2, kq, -]$ reduce to forms $\hat{f}_k$ for which there exists a small pair (x_0, y_0) such that $q^2 \mid \hat{f}_k(x_0,y_0)$ while $q^2 \mid N$. From Theorem B−1, we know that these reduced forms appear on the principal cycle of the class group of discriminant Δ_q. To detect them, we start a walk in the principal cycle from the principal form $\mathbf{1}_N$, and apply Rho until the Coppersmith-like method finds these small solutions.

```
Input: $N = pq^2, m, t$
Output: $p, q$
$h \leftarrow \mathbf{1}_N$
while $(x_0, y_0)$ not found do
    $h \leftarrow \mathsf{Rho}(h)$
    $x_0/y_0 \leftarrow \mathsf{HomogeneousCoppersmith}(h, N, m, t)$
$q \leftarrow \mathsf{Sqrt}(\mathsf{Gcd}(h(x_0, y_0), N))$
return $(N/q^2, q)$
```

Figure B.3: Factoring $N = pq^2$

B.4.2. Heuristic Correctness and Analysis of Our Algorithm

Assuming Heuristic B.2, starting from $\mathbf{1}_N$, after $O(R_p)$ iterations, the algorithm will stop on a reduced form whose roots will be found with our Coppersmith-like method (for suitable values of m and t) since they will satisfy the expected $N^{1/9}$ bound. The computation of $\gcd(h(x_0, y_0), N)$ will therefore expose q^2 and factor N. The time complexity of our algorithm is then heuristically $O(R_p \mathsf{Poly}(\log N))$, whereas the space complexity is $O(\log N)$. The worst-case complexity is $O(p^{1/2} \log p \, \mathsf{Poly}(\log N))$. For small regulators, such as in REAL-NICE cryptosystem (see. Subsection 5.1), the time complexity is polynomial.

This algorithm can be generalised with a few modifications to primes p such that $p \equiv 3 \pmod 4$, by considering $\Delta_q = 4pq^2$. Moreover if the bit-sizes of p and q are unbalanced, our experiments suggest that the size of the roots will be small enough (see end of Subsection 2.3 and Section 3), so the factoring algorithm will also work in this case, with the same complexity.

Comparison with other Deterministic Factorisation Methods.

Boneh, Durfee and Howgrave-Graham presented in [BDH99] an algorithm for factoring integers $N = p^r q$. Their main result is the following:

Lemma B − 3 ([BDH99]). *Let $N = p^r q$ be given, and assume $q < p^c$ for some c. Furthermore, assume that P is an integer satisfying $|P - p| < p^{1-\frac{c}{r+c}-2\frac{r}{d}}$. Then the factor p may be computed from N, r, c and P by an algorithm whose running time is dominated by the time it takes to run LLL on a lattice of dimension d.*

For $r = 2$ and $c = 1$, this leads to a deterministic factoring algorithm which consists in exhaustively search for an approximation P of p and to solve the polynomial equation $(P + X)^2 \equiv 0 \pmod{p^2}$ with a method *à la* Coppersmith. The approximation will be found after $O(p^{1/3}) = O(N^{1/9})$ iterations.

The fastest deterministic generic integer factorisation algorithm is actually a version of Strassen's algorithm [Str76] from Bostan, Gaudry and Schost [BGS07], who ameliorates a work of Chudnovsky and Chudnovsky [CC87] and proves a complexity of

$O(\mathsf{M}_{\mathsf{int}}(\sqrt[4]{N}\log N))$ where $\mathsf{M}_{\mathsf{int}}$ is a function such that integers of bit-size d can me multiplied in $\mathsf{M}_{\mathsf{int}}(d)$ bit operations. More precisely, for numbers of our interest, Lemma 13 from [BGS07] gives the precise complexity:

Lemma B − 4 ([BGS07]). *Let b, N be two integers with $2 \leq b < N$. One can compute a prime divisor of N bounded by b, or prove that no such divisor exists in space $O(\sqrt{b}\log N)$ bits and $O\left(M_{int}(\sqrt{b}\log N) + \log b M_{int}(\log N)\log\log N\right)$ bit operations.*

In particular, for $b = N^{1/3}$, the complexity is $\tilde{O}(N^{1/6})$, with a very large space complexity compared to our algorithm. Moreover, none of these two last of algorithms can actually factor an integer of cryptographic size. The fact that a prime divisor has a small regulator does not help in these algorithms, whereas it makes the factorisation polynomial in our method.

B.5. Cryptanalysis of the NICE Cryptosystems

Hartmann, Paulus and Takagi proposed the elegant *NICE* encryption scheme ([PT98, HPT99, PT00]), based on imaginary quadratic fields and whose main feature was a quadratic decryption time. Later on, several other schemes, including (special) signature schemes relying on this framework have been proposed. The public key of these NICE cryptosystems contains a discriminant $\Delta_q = -pq^2$ together with a reduced ideal $\mathfrak{h}$ whose class belongs to the kernel of $\bar{\varphi}_q$. The idea underlying the NICE cryptosystem is to hide the message behind a random element $[\mathfrak{h}]^r$ of the kernel. Applying $\bar{\varphi}_q$ will make this random element disappear, and the message will then be recovered.

Then, in [JSW08], Jacobson, Scheidler and Weimer embedded the original NICE cryptosystem in *real* quadratic fields. Whereas the idea remains essentially the same as the original, the implementation is very different. The discriminant is now $\Delta_q = pq^2$, but because of the differences between imaginary and real setting, these discriminant will have to be chosen carefully. Among these differences, the class numbers are expected to be small with very high probability (see the Cohen-Lenstra heuristics [CL84]). Moreover, an equivalence class does not contain a *unique* reduced element anymore, but a multitude of them, whose number is governed by the size of the fundamental unit. The rough ideas to understand these systems and our new attacks are given in the following. The full description of the systems is omitted for lack of space but can be found in [PT98,JSW08].

B.5.1. Polynomial-Time Key Recovery in the Real Setting

The core of the design of the REAL-NICE encryption scheme is the very particular choice of the secret prime numbers p and q such that $\Delta_K = p$ and $\Delta_q = pq^2$. They are chosen such that the ratio $R_{\Delta_q}/R_{\Delta_K}$ is of order of magnitude of q and that R_{Δ_K} is bounded by a polynomial in $\log(\Delta_K)$. To ensure the first property, it is sufficient to choose q such that $q - \left(\frac{\Delta_K}{q}\right)$ is a small multiple of a large prime. If the second property is very unlikely to naturally happen since the regulator of p is generally of the order of magnitude of $\sqrt{p}$, it is indeed quite easy to construct fundamental primes with small regulator. The authors of [JSW08] suggest to produce a prime p as a so-called *Schinzel sleeper*, which is a positive squarefree integer of the form $p = a^2x^2 + 2bx + c$ with a, b, c, x in $\mathbf{Z}$, $a \neq 0$ and $b^2 - 4ac$ dividing $4\gcd(a^2, b)^2$. Schinzel sleepers are known to have a regulator of the order $\log(p)$ (see [CW05]). Some care must be taken when setting the (secret) a, b, c, x values, otherwise the resulting $\Delta_q = pq^2$ is subject to factorisation attacks described in [Wei04]. We do not provide here more details on these choices since the crucial property for our attack is the fact that the regulator is actually of the order $\log(p)$. The public key consists of the sole discriminant Δ_q. The message is carefully embedded (and padded) into a primitive $\mathscr{O}_{\Delta_q}$-ideal so that it will be recognised during decryption. Instead of moving the message ideal $\mathfrak{m}$ to a different equivalence class (like in the imaginary case), the encryption actually hides the message in the cycle of reduced ideal of its own equivalent class by multiplication of a random principal $\mathscr{O}_{\Delta_q}$-ideal $\mathfrak{h}$ (computed during encryption). The decryption process consists then in applying the (secret) map $\bar{\varphi}_q$ and perform an exhaustive search for the padded message in the *small* cycle of $\bar{\varphi}_q([\mathfrak{m}\mathfrak{h}])$. This exhaustive search is actually possible thanks to the choice of p which has a very small regulator. Like in the imaginary case, the decryption procedure has a quadratic complexity and significantly outperforms an RSA decryption for any given security level (see Table 3 from [JSW08]). Unfortunately, due to the particular but necessary choice of the secret prime p, the following result states the total insecurity of the REAL-NICE system.

Result B.1. Algorithm B.3 recovers the secret key of REAL-NICE in polynomial time in the security parameter under Heuristic B.2 since the secret fundamental discriminant p is chosen to have a regulator bounded by a polynomial in $\log p$.

We apply the cryptanalysis on the following example. The Schinzel polynomial $\mathrm{S}(\mathrm{X}) = 2725^2\mathrm{X}^2 + 2 \cdot 3815\mathrm{X} + 2$ gives a suitable 256-bit prime p for the value $\mathrm{X}_0 = 103042745825387139695432123167592199$. This prime has a regulator $\mathrm{R}_{\Delta_\mathrm{K}} \simeq 90.83$. The second 256-bit prime q is chosen following the recommendations from [Wei04]. This leads to a the discriminant

$$\begin{aligned}\Delta_q \quad = \quad & 2873693882331004487338071614228207339618684390675746327479263873414406060283051 0\\ & 807386691634892735925990545294422710538698324853636823418921245006784003227198 42\\ & 63278692833860326257638544601057379571931906787755152745236263303465093\end{aligned}$$

Our algorithm recovers the prime

$$q = 60372105471499634417192859173853663456123015267207769653235558092781188395563$$

from Δ_q after 45 iterations in 42.42 seconds on a standard laptop. The rational root is $\frac{x_0}{y_0}$ equal to $-\frac{21555116117109964456 23}{35448742771347786589 48}$, where x_0 and y_0 satisfy $\frac{\log(\Delta_q)}{\log(|x_0|)} \simeq 10.8$ and $\frac{\log(\Delta_q)}{\log(|y_0|)} \simeq 10.7$.

B.5.2. Polynomial-Time Key Recovery of the Original NICE

As mentioned above, the public key of the original NICE cryptosystem contains the representation of a reduced ideal $\mathfrak{h}$ whose class belongs to the kernel of the surjection $\bar{\varphi}_q$. The total-break of the NICE cryptosystem is equivalent to solving the following *kernel problem*.

Definition B – 6 (Kernel Problem [BPT04]). Let λ be an integer, p and q be two λ-bit primes with $p \equiv 3 \pmod 4$. Fix a non-fundamental discriminant $\Delta_q = -pq^2$. Given an element $[\mathfrak{h}]$ of $\ker \bar{\varphi}_q$, factor the discriminant Δ_q.

Castagnos and Laguillaumie proposed in [CL09] a polynomial-time algorithm to solve this problem. We propose here a completely different solution within the spirit of our factorisation method and whose complexity is also polynomial-time. As discuss in Subsection 2.3, the idea is to benefit from the fact that the public ideal $\mathfrak{h}$ corresponds to a reduced quadratic form, $\hat{f}_k$, which represents q^2. We thus find these x_0 and y_0 such that $\gcd(\hat{f}_k(x_0, y_0), \Delta_q) = q^2$ with the Coppersmith method of Section B.3.

Result B.2. The Homogeneous Coppersmith method from Section B.3 solves the Kernel Problem in polynomial time in the security parameter under Heuristic B.1.

We apply our key recovery on the example of NICE proposed in [JJ00, CL09]:

$$\begin{aligned}
\Delta_q &= -100113361940284675007391903708261917456537242594667491514934053946421992795516 8\\
&\quad 182167600836407521987097261997327018438644118532496445353657288020224981856655 92\\
&\quad 98370854645328210791277591425676291349013221520022224671621236001656120923
\end{aligned}$$

$$\begin{aligned}
a &= 570226877089425831816858843811755887130078318076999519509271589575517370039 9\\
&\quad 141486895731384747
\end{aligned}$$

$$\begin{aligned}
b &= 336123604058275478495858629801794911064873174560593016466681956960675502977 3\\
&\quad 074415823039847007
\end{aligned}$$

The public key consists in Δ_q and $\mathfrak{h} = (a, b)$. Our Coppersmith method finds in less that half a second the root $u_0 = \dfrac{-103023911}{349555951} = \dfrac{x_0}{y_0}$ and

$$\begin{aligned}
h(x_0, y_0) &= 536312317197703883982960999928233845099174632823695735108 9\\
&\quad 424577488705612036597900253463323383022772146551393561497 1\\
&\quad 5939077126809522499818706407364011207 29 = q^2.
\end{aligned}$$

All our experiments have been run on a standard laptop under Linux with software Sage. The lattice reduction have been performed with Stehlé's fplll [FPL16].

Acknowledgements. We warmly thank Denis Simon and Brigitte Vallée for helpful discussions and the reviewers for their useful comments. Part of this work was supported by the Commission of the European Communities through the ICT program under contract ICT-2007-216676 ECRYPT II.

Chapter C

Linearly Homomorphic Encryption from DDH

Joint work with Fabien Laguillaumie
[CL15]

Abstract. We design a linearly homomorphic encryption scheme whose security relies on the hardness of the decisional Diffie-Hellman problem. Our approach requires some special features of the underlying group. In particular, its order is unknown and it contains a subgroup in which the discrete logarithm problem is tractable. Therefore, our instantiation holds in the class group of a non maximal order of an imaginary quadratic field. Its algebraic structure makes it possible to obtain such a linearly homomorphic scheme whose message space is the whole set of integers modulo a prime p and which supports an unbounded number of additions modulo p from the ciphertexts. A notable difference with previous works is that, for the first time, the security does not depend on the hardness of the factorization of integers. As a consequence, under some conditions, the prime p can be scaled to fit the application needs.

C.1. Introduction

Encryption protocols insure confidentiality during information transmission. They are the heart of any communication architecture. Their security has been formally defined for long, and many efficient encryption schemes fulfill the strongest security requirement, namely the indistinguishability of ciphertexts under an adaptive chosen message attack. Roughly speaking, it means that an attacker will learn not even a single bit of a message, given its encryption, even if he has access to a decryption oracle.

Paradoxically, a widely deployed kind of encryption scheme has an "algebraic" property which precludes it to reach this highest level of security. It is called ***homomorphic***, because an operation on the ciphertexts translates into an operation on the underlying plaintexts. It is well-known that such protocols cannot reach the highest level of security, even though, this homomorphic property is actually very important for many applications, like e-voting for instance. Indeed, an *additively* homomorphic encryption makes it possible to obtain an encryption of the sum of all the ballots (which consists in 0 or 1 in the case of a 2-choice referendum for instance) from their encryption, so that a single decryption will reveal the result of the election, saving a lot of computational resources which would have been necessary to decrypt all the ciphertexts one by one. Linearly homomorphic encryption schemes have attracted a lot of attention because of their potential applications. A tremendous breakthrough related to homomorphic encryption was Gentry's theoretical construction of a *fully* homomorphic encryption scheme [Gen09], which actually allows to evaluate any function on messages given their ciphertexts.

Currently, no linearly homomorphic encryption scheme is secure under a discrete logarithm related assumption. This theoretical question has been open for thirty years. In this paper, we provide the first construction of such a scheme.

Related Work. The story of homomorphic encryption begins with the first probabilistic encryption scheme, which was also homomorphic, by Goldwasser and Micali from [GM84], improved by Benaloh in his thesis [Ben88], then by Naccache and Stern in [NS98] and Okamoto and Uchiyama [OU98]. One of the most achieved system was actually designed by Paillier [Pai99]. Its semantic security relies on the decisional composite residuosity assumption. Paillier's scheme has then been generalized by Damgård and Jurik [DJ01], allowing to encrypt larger messages. This family of practical linearly homomorphic schemes is still growing with the recent work of Joye and Libert [JL13]. The security of these schemes is based on the problem of factoring RSA integers (including the elliptic curve variant of Paillier [Gal02]).

To design a linearly homomorphic encryption based on the Discrete Logarithm problem (DL), a folklore solution consists in encoding the message in the exponent of an Elgamal encryption, *i.e.*, in encrypting m as $(g^r, h^r g^m)$ where g is a generator of a cyclic group $\mathbb{G} = \langle g \rangle$ and $h = g^x$ is the public key. Unfortunately, to decrypt, one has to recover m from g^m and as the DL problem in $\mathbb{G}$ must be intractable, m has to be small enough to ensure a fast decryption. As a result, only a logarithmic number of additions is possible. There have been some attempts to reach a fully additive homomorphy based on the DL problem, with a variant of Elgamal modulo p^2 ([CPP06]) or with messages encoded as a small smooth number ([CC07]); both solutions still have a partial homomorphy. In [WWP+11], the map $m \mapsto g_0^m \mod p_0$ is used with the plain Elgamal, where p_0 is a prime such that $p_0 - 1$ is smooth and g_0 is a primitive root modulo p_0. Unfortunately, although not clearly stated, this scheme only supports a limited number of additions, and it is not semantically secure as the set of encoded messages does not belong to a proper subgroup of $(\mathbf{Z}/p\mathbf{Z})^\times$ where the Decisional Diffie-Hellman assumption (DDH) holds.

A full solution has been proposed by Bresson *et al.* in [BCP03]. However, their scheme is not only based on the DL problem but also on the factorization problem. It is less efficient than [Pai99] but has an additional property: it has a double trapdoor. The idea is to use the same setting as Paillier: In $(\mathbf{Z}/\mathrm{N}^2\mathbf{Z})^\times$, the DL problem in basis $f = 1+\mathrm{N}$ is easy. Bresson *et al.* use an Elgamal encryption of the message m as $(g^r, f^m \cdot h^r)$ modulo N^2, where N is an RSA integer.

To our knowledge, designing a linearly homomorphic scheme based on the sole hardness of the DL problem is an open problem, as stated in [CPP06]. Some other schemes allow more homomorphic operations, like [BGN05] or [CL12]. As already mentioned, a fully homomorphic encryption (FHE) scheme appeared in 2009 [Gen09]. Its security relies on hard problems related to lattices. The latest developments of FHE [BV14] are getting more and more efficient and might become operational soon for applications that need a complex treatment over ciphertexts. Meanwhile, for applications that need only to add ciphertexts, protocols that rely on "classical" algorithmic assumptions are still more competitive, in particular in terms of compactness.

Our Contributions. Our contribution has both a theoretical and a practical impact. On one hand, we propose a linearly homomorphic encryption scheme whose security relies on the hardness of the decisional Diffie-Hellman problem. In particular it is the first time that the security of such a scheme does not depend on the hardness of the factorization of integers. On the other hand, we provide an efficient implementation within some specific group, namely the class group of orders in imaginary quadratic fields.

The design of our scheme is somehow similar to the one of [BCP03]. We use a group $\mathrm{G} = \langle g \rangle$ such that the DDH assumption holds in G and such that there exists a subgroup $\langle f \rangle$ of G where the DL problem is easy (called a DDH group with an easy DL subgroup). Then the core of the protocol is an Elgamal encryption of the message m as $(g^r, f^m \cdot h^r)$ for a random r. In our case, the message space will be $(\mathbf{Z}/p\mathbf{Z})^*$, where p is a prime. Compared to some other linearly homomorphic schemes, ours allows some flexibility as p can be chosen (with some restrictions) independently from the security parameter.

To reach this unnatural feature without involving the factorization problem, we had to use the particular algebraic structure of class groups of imaginary quadratic fields, which have some specificities which seem hard to find in other groups. We designed a method to compute a group of unknown[1] order (to insure the hardness of a partial discrete logarithm assumption) which contains an easy DL subgroup (of known order). The interest of class group of orders in imaginary (or real) quadratic fields in cryptography decreased after critical attacks by Castagnos *et al.* [CL09, CJLN09] on some specific cryptosystems such as NICE [HPT99, PT00] and its real variant [JSW08]. These attacks will not apply in our setting. Indeed, these attacks recover the secret key by expos-

[1] Using groups of unknown order in cryptography has already been done [Bre00, CHN99, DF02]

ing the factorization of the discriminant of the field, thanks to the structure of the kernel of the surjection between the class group of a non maximal order to the class group of the maximal order. In our case, the factorization of the discriminant will be public and we will use constructively the ideas of [CL09]: the subgroup with an easy DL will be precisely the kernel of this surjection. The security of our scheme is proved to rely only on the hardness of the DDH problem in the class group of a non maximal order and on the hardness of computing class numbers. Several systems that adapt either Diffie-Hellman or Elgamal in class groups are already based on the DL problem and the DDH assumption in class groups of maximal order ([BW88, BDW90, SP05, BH01, BV07]) of discriminant Δ_K. The current best known algorithms to solve these problems have a sub-exponential complexity of complexity $L_{|\Delta_K|}(1/2, o(1))$ (cf. [BJS10]). It means that the factorization problem (or the discrete logarithm problem in a finite field) can be solved asymptotically *faster* than the discrete logarithm in the class group.[2] Moreover, arithmetic operations in class groups are very efficient, since the reduction and composition of quadratic forms have a quadratic time complexity (and even quasi linear using fast arithmetic).

As a result, our scheme is very competitive. With a straightforward implementation and using an underlying arithmetics very favorable to [Pai99, BCP03], it compares very well with these linearly homomorphic protocols. With a similar level of security, it is faster than [BCP03] with a 2048 bits modulus, and the decryption process is faster than Paillier's for a 3072 bits modulus.

A very nice application of our protocol is that it can be used directly in Catalano and Fiore's linearly homomorphic encryption transformation to evaluate degree-2 computations on ciphertexts [CF15]. Their technique requires the message space to be a public ring in which it is possible to sample elements uniformly at random. Our scheme has this feature naturally, contrary to some of the other additively homomorphic schemes. It is therefore a very competitive candidate in 2-server delegation of computation over encrypted data (see [CF15] for more details).

The rest of the paper is organized as follows. In Section C.2, we formalize the notion of DDH *Group with an Easy* DL *Subgroup*, give reductions between related problems and propose a generic construction of a linearly homomorphic encryption scheme which relies on such group, and prove its security. Sections C.3 and C.4 present our instantiation in class groups. We give benchmarks and comparisons before concluding. Background on linearly homomorphic encryption can be found in Appendix C.A. Background on class groups of imaginary quadratic fields and their use for DL based cryptography are given in Appendix C.B.

C.2. DDH Group with an Easy DL Subgroup

In this section, we introduce and formalize the concept of a group in which the decisional Diffie-Hellman problem is hard, whereas it contains a subgroup in which the

[2] Note that it is well known (see [HM00a] for instance) that computing the class number of a quadratic field of discriminant Δ allows to factor Δ . However for our scheme, the factorization of the discriminant Δ will be public or Δ will be a prime, so we will not rely on the hardness of the factorization problem.

discrete logarithm problem is easy. This problem has already been used to design cryptosystems, including, for instance, Bresson *et al.*'s encryption scheme [BCP03]. It will be adjusted to build our new encryption protocol.

C.2.1. Definitions and Reductions

Definition C – 1. We define a DDH *group with an easy* DL *subgroup* as a pair of algorithms (Gen, Solve). The Gen algorithm is a group generator which takes as input two parameters λ and μ and outputs a tuple (B, n, p, s, g, f, G, F). The integers B, n, p and s are such that s is a λ-bit integer, p is a μ-bit integer, $\gcd(p, s) = 1$, $n = p \cdot s$ and B is an upper bound for s. The set $(G, \cdot)$ is a cyclic group of order n generated by g, and $F \subset G$ is the subgroup of G of order p and f is a generator of F. The upper bound B is chosen such that the distribution induced by $\{g^r, r \overset{\$}{\leftarrow} \{0, \dots, Bp-1\}\}$ is statistically indistinguishable from the uniform distribution on G. We assume that the canonical surjection $\pi : G \to G/F$ is efficiently computable from the description of G, H and p and that given an element $h \in G/F$ one can efficiently lift h in G, *i.e.*, compute an element $h_\ell \in \pi^{-1}(h)$.

We suppose moreover that:

1. The DL problem is easy in F. The Solve algorithm is a deterministic polynomial time algorithm that solves the discrete logarithm problem in F:

$$\Pr\Big[x = x^\star : (B, n, p, s, g, f, G, F) \overset{\$}{\leftarrow} \mathsf{Gen}(1^\lambda, 1^\mu), x \overset{\$}{\leftarrow} \mathbf{Z}/p\mathbf{Z}, X = f^x, \\ x^\star \leftarrow \mathsf{Solve}(B, p, g, f, G, F, X)\Big] = 1$$

2. The DDH problem is hard in G even with access to the Solve algorithm:

$$\Bigg|\Pr\Big[b = b^\star : (B, n, p, s, g, f, G, F) \overset{\$}{\leftarrow} \mathsf{Gen}(1^\lambda, 1^\mu), x, y, z \overset{\$}{\leftarrow} \mathbf{Z}/n\mathbf{Z}, X = g^x, Y = g^y, \\ b \overset{\$}{\leftarrow} \{0,1\}, Z_0 = g^z, Z_1 = g^{xy}, b^\star \overset{\$}{\leftarrow} \mathscr{A}(B, p, g, f, G, F, X, Y, Z_b, \mathsf{Solve}(.))\Big] - \frac{1}{2}\Bigg|$$

   is negligible for all probabilistic polynomial time attacker $\mathscr{A}$.

The bound B for the order s in Definition C–1 can be chosen as $B = 2^{2\lambda}$. Indeed, according to Lemma C–4 in Appendix C.C, the statistical distance of $\{g^r, r \overset{\$}{\leftarrow} \{0, \dots, Bp-1\}\}$ to the uniform distribution is upper bounded by $n/(4pB) = s/2^{2\lambda+2} \leqslant 2^{-\lambda-2}$ which a negligible function of λ.

It is fundamental to note that in this definition, the order n of the group G is *not* an input of the adversary or of the Solve algorithm: Only the bound Bp is implicitly given. Indeed, if n or s were efficiently computable from the description of G, a DDH group with an easy DL subgroup would not exist since it would be possible to partially compute discrete logarithms. More formally, let us define the following partial discrete logarithm problem initially introduced by Paillier in [Pai99], in the context of the group $(\mathbf{Z}/N^2\mathbf{Z})^\times$.

Definition C – 2 (Partial Discrete Logarithm (PDL) Problem)**.** Let $(\mathsf{Gen}, \mathsf{Solve})$ be a DDH group with an easy DL subgroup. Let $(B, n, p, s, g, f, G, F) \overset{\$}{\leftarrow} \mathsf{Gen}(1^\lambda, 1^\mu)$, $x \overset{\$}{\leftarrow} \mathbf{Z}/n\mathbf{Z}, X = g^x$. The *Partial Discrete Logarithm Problem* consists in computing x modulo p; given (B, p, g, f, G, F, X) and access to the Solve algorithm.

Lemma C – 1. *Let* $(\mathsf{Gen}, \mathsf{Solve})$ *be a* DDH *group with an easy* DL *subgroup and let the tuple* (B, n, p, s, g, f, G, F) *be an output of* $\mathsf{Gen}(1^\lambda, 1^\mu)$*. The knowledge of* n *makes the* PDL *problem easy.*

Proof. If an adversary is given an instance (B, p, g, f, G, F, X) of the PDL problem, as well as n, he can compute $s = n/p$ and then for all $h \in G$, h^s lies in F. The adversary can run the Solve algorithm with g^s as input to find $\alpha \in \mathbf{Z}/p\mathbf{Z}$ such that $g^s = f^\alpha$. Note that $\alpha \not\equiv 0 \pmod p$ as g has order n. Thanks to another run of the Solve algorithm with X^s as input, the adversary obtains $\beta \in \mathbf{Z}/p\mathbf{Z}$ such that $X^s = g^{sx} = f^\beta$. Eventually, he computes $x \equiv \beta\alpha^{-1} \pmod p$. □

Lemma C – 2. *Let* G *be a* DDH *group with an easy* DL *subgroup. The* DDH *problem in* G *reduces to the* PDL *problem.*

Proof. Let $(B, p, g, f, G, F, X, Y, Z)$ be an instance of the DDH problem in G. Three queries to the PDL oracle respectively on (B, p, g, f, G, F, X), (B, p, g, f, G, F, Y) and (B, p, g, f, G, F, Z), gives the adversary x, y and z modulo p. His answer to the DDH instance will be 1 if and only if $xy \equiv z \pmod p$. Indeed, if (X, Y, Z) is a true DDH triple then $xy \equiv z \pmod n$ and he always finds the right answer. Conversely, if (X, Y, Z) is not a DDH triple, $xy \not\equiv z \pmod n$, and then the adversary fails to correctly responds if $xy \equiv z \pmod p$. But this happens with probability $1/p$. As a result, we have sketched a probabilistic polynomial time adversary against DDH with a non negligible advantage equals to $\frac{1}{2}(1 - \frac{1}{p})$. □

Remark C – 1. Combining Lemmas C – 1 and C – 2 we get that as previously mentioned, with the notation of Definition C – 1, if n is easily computable from the description of G, then the DDH problem in G is easy so, G can not be a DDH group with an easy DL subgroup.

The following problem was introduced in [BCP03] in $(\mathbf{Z}/N^2\mathbf{Z})^\times$. It is a variant of the computational Diffie-Hellman problem, that we adapt to our general context.

Definition C – 3 (Lift Diffie-Hellman (LDH) Problem)**.** Let $(\mathsf{Gen}, \mathsf{Solve})$ be a DDH group with an easy DL subgroup. Let $(B, n, p, s, g, f, G, F) \overset{\$}{\leftarrow} \mathsf{Gen}(1^\lambda, 1^\mu)$. Let $x, y \overset{\$}{\leftarrow} \mathbf{Z}/n\mathbf{Z}, X = g^x, Y = g^y$ and $Z = g^{xy}$ and $\pi : G \to G/F$ be the canonical surjection. The *Lift Discrete Logarithm Problem* consists in computing Z, given the tuple $(B, p, g, f, G, F, X, Y, \pi(Z))$ and access to the Solve algorithm.

In the following theorem we prove that this problem is equivalent to the PDL problem. Curiously only one implication was proved in [BCP03].

Theorem C – 1. *In a* DDH *group with an easy* DL *subgroup, the* LDH *and* PDL *are equivalent.*

Proof. In all the proof, we implicitly set $s = n/p$ and $\alpha \in (\mathbf{Z}/p\mathbf{Z})^\times$ such that $g^s = f^\alpha$ and denote $\beta \equiv \alpha^{-1} \pmod p$. Let us first prove that the PDL problem reduces to the LDH problem, which is a direct generalization of the proof of [BCP03, Theorem 10]. Let $(\mathrm{B}, p, g, f, \mathrm{G}, \mathrm{F}, \mathrm{X})$ be a PDL challenge and let denote $\mathrm{X} = g^x$ where $x = x_1 + x_2 p$ with $x_1 = x \mod p$. The adversary draws $r_1 \overset{\$}{\leftarrow} \{0, \dots, \mathrm{B}-1\}$, $r_2 \overset{\$}{\leftarrow} \{0, \dots, p-1\}$ and sets $\mathrm{Y} = g^{r_1} f^{r_2}$. Note that $\mathrm{Y} = g^{r_1 + s\beta r_2}$. Let us prove that the random variable Y is statistically indistinguishable from the uniform distribution in G.

The distance between Y and the uniform distribution in G is the same than the distance between $\mathrm{Y}' = r_1 + s\beta r_2 \mod n$ with r_1 uniformly drawn in $\{0, \dots, \mathrm{B}-1\}$ and r_2 independently uniformly drawn in $\{0, \dots, p-1\}$ and the uniform distribution in $\{0, \dots, n-1\}$. Let y be an element of $\{0, \dots, n-1\}$, we denote $y = y_1 + y_2 s$ with $y_1 \in \{0, \dots, s-1\}$ and $y_2 \in \{0, \dots, p-1\}$ the euclidean division of y by s. We have

$$\Pr[\mathrm{Y}' = y] = \Pr[\mathrm{Y}' = y_1 + y_2 s] = \Pr[r_1 + s\beta r_2 \equiv y_1 + y_2 s \pmod n] =$$

$$\Pr[r_1 \equiv y_1 \pmod s] \Pr[r_2 \beta \equiv y_2 \pmod p] = \Pr[r_1 \equiv y_1 \pmod s]/p$$

as $\beta \not\equiv 0 \pmod p$. Now let $\mathrm{B} = qs + r$ with $0 \leqslant r < s$ be the euclidean division of B by s. We proceed as in the proof of Lemma C – 4 in Appendix C.C. For $y_1 < r$, $\Pr[r_1 \equiv y_1 \pmod s] = (q+1)/\mathrm{B} > \frac{1}{s}$ and for $y_1 \geqslant r$, $\Pr[r_1 \equiv y_1 \pmod s] = q/\mathrm{B} \leqslant \frac{1}{s}$. Eventually,

$$\Delta(\mathrm{X}, \mathrm{Y}) = r\left(\frac{q+1}{\mathrm{B}p} - \frac{1}{n}\right) = \frac{r(s-r)}{\mathrm{B}n} = \frac{r(n-pr)}{p\mathrm{B}n} \leqslant \frac{r(n-r)}{p\mathrm{B}n}.$$

This last quantity is the statistical distance of $\{g^r, r \overset{\$}{\leftarrow} \{0, \dots, \mathrm{B}p-1\}\}$ to the uniform distribution in G which is suppose to be negligible. This proves that Y is statistically indistinguishable from the uniform distribution in G.

The adversary then compute $\mathrm{Z}' = \pi(\mathrm{X}^{r_1}) = \pi(\mathrm{X}^{r_1 + s\beta r_2})$ and queries the LDH oracle with $(\mathrm{B}, p, g, f, \mathrm{G}, \mathrm{F}, \mathrm{X}, \mathrm{Y}, \mathrm{Z}')$. The oracle provides with non negligible probability

$$\mathrm{Z} = \mathrm{X}^{r_1 + s\beta r_2} = \mathrm{X}^{r_1}(g^x)^{s\beta r_2} = \mathrm{X}^{r_1} g^{(x_1 + x_2 p)(s\beta r_2)} = \mathrm{X}^{r_1} g^{x_1 s\beta r_2} = \mathrm{X}^{r_1} f^{x_1 r_2}.$$

Then, $\mathrm{Z}/\mathrm{X}^{r_1} = f^{x_1 r_2}$ and running the Solve algorithm on this value gives $x_1 r_2 \pmod p$ to the adversary from which he can get x_1, the answer to the PDL instance.

Now, let us prove that the LDH problem reduces to the PDL problem. Let us consider $\mathrm{X} = g^x, \mathrm{Y} = g^y, \mathrm{Z} = g^{xy}$ for random x and y, such that the LDH challenge writes as $(\mathrm{B}, p, g, f, \mathrm{G}, \mathrm{F}, \mathrm{X}, \mathrm{Y}, \mathrm{Z}' = \pi(\mathrm{Z}))$. The adversary makes two queries to the PDL oracle relative to X and Y, from which he obtains $x \pmod p$ and $y \pmod p$. The adversary draws $r_1 \overset{\$}{\leftarrow} \{0, \dots \mathrm{B}-1\}$ and $r_2 \overset{\$}{\leftarrow} \{0, \dots, p-1\}$ and sets $\mathrm{U} = g^{r_1} f^{r_2}$, which is as before statistically indistinguishable from the uniform distribution in G. The adversary queries the PDL oracle with U, which gives $r_1 + s\beta r_2 \pmod p$ as $\mathrm{U} = g^{r_1+s\beta r_2}$. From this answer, the adversary can compute $s\beta \pmod p$. From the definition of a DDH group with an easy DL subgroup, the adversary can compute $\mathrm{Z}'_\ell \in \pi^{-1}(\mathrm{Z}')$. He then draws $r \xleftarrow{\$} \mathbf{Z}/p\mathbf{Z}$ and computes $\mathrm{V} = f^r \mathrm{Z}'_\ell$. The random variable V is uniformly distributed in G. As $\pi(\mathrm{V}) = \mathrm{Z}' = \pi(\mathrm{Z})$, there exists $\gamma \in \mathbf{Z}/p\mathbf{Z}$ such that $\mathrm{V} = f^\gamma \mathrm{Z} = g^{s\beta\gamma+xy}$. From a last call to the PDL oracle, the adversary can get $s\beta\gamma + xy \pmod p$ from which he can compute γ since $\gcd(s\beta, p) = 1$. Eventually, the adversary deduces Z from $\mathrm{V} = f^\gamma \mathrm{Z}$. □

We now further analyze the relations between the problems in G/F and G. We first give a lemma that shows that we can define a morphism in order to lift the elements from G/F to G.

Lemma C−3. *Let* $(\mathrm{B}, n, p, s, g, f, \mathrm{G}, \mathrm{F}) \xleftarrow{\$} \mathsf{Gen}(1^\lambda, 1^\mu)$ *where* (Gen, Solve) *is a* DDH *group with an easy* DL *subgroup. Denote* $\pi : \mathrm{G} \to \mathrm{G}/\mathrm{F}$ *the canonical surjection. The map* $\psi : \mathrm{G}/\mathrm{F} \to \mathrm{G}$ *s.t.* $h \mapsto h_\ell^p$*, where* $h_\ell \in \pi^{-1}(h)$*, is an effective injective morphism.*

Proof. First ψ is well defined: if $h_\ell^{(1)}, h_\ell^{(2)} \in \pi^{-1}(h)$ are two distinct pre-images of h then there exists an element $f^r \in \mathrm{F}$ such that $h_\ell^{(1)} = f^r h_\ell^{(2)}$, and $(h_\ell^{(1)})^p = (h_\ell^{(2)})^p$ as F is of order p. Moreover it is easy to see that ψ is a morphism. Consider h in G/F such that $\psi(h) = h_\ell^p = 1$ in G, with $h_\ell \in \pi^{-1}(h)$. Applying π gives $\pi(h_\ell)^p = h^p = 1$. As G/F is of order s prime to p then $h = 1$, so the map is injective. Eventually, ψ is efficiently computable as computing h_ℓ is easy by definition of a DDH group with an easy DL subgroup. □

Theorem C−2. *Let* $(\mathrm{B}, n, p, s, g, f, \mathrm{G}, \mathrm{F}) \xleftarrow{\$} \mathsf{Gen}(1^\lambda, 1^\mu)$ *where* (Gen, Solve) *is a* DDH *group with an easy* DL *subgroup. The* DL *problem in* G/F *reduces to the* DL *problem in* G.

Proof. Consider a DL problem instance in G/F: Let $h = g^x$ where g is a generator of G/F of order s and $x \xleftarrow{\$} \mathbf{Z}/s\mathbf{Z}$. The adversary chooses $r, r' \xleftarrow{\$} (\mathbf{Z}/p\mathbf{Z})^\times$ and computes $g_\ell = \psi(g) f^r$ and $h_\ell = \psi(h) f^{r'}$ where the map ψ is defined in Lemma C−3. The element $\psi(g)$ has order s as ψ is injective and f^r has order p. As $\gcd(p, s) = 1$, g_ℓ has order ps and is a generator of G. Moreover, G can be viewed as the direct product $\psi(\mathrm{G}/\mathrm{F}) \times \mathrm{F}$. The element h is uniformly distributed in G/F, $f^{r'}$ is uniformly distributed in F so $h_\ell = \psi(h) f^{r'}$ is uniformly distributed in G. As a consequence, an oracle for the DL problem in G gives x_ℓ such that $g_\ell^{x_\ell} = h_\ell$ to the adversary with a non negligible advantage. He then has $g_\ell^{x_\ell} = \psi(g)^{x_\ell} f^{r x_\ell} = h_\ell = \psi(h) f^{r'}$. By the uniqueness of the decomposition of an element of G in a product of an element of $\psi(\mathrm{G}/\mathrm{F})$ and an element of F, and because ψ is injective, we must have $g^{x_\ell} = h$ and therefore $x_\ell \equiv x \pmod s$. □

Unfortunately, it seems unlikely that a similar reduction of the DDH problem in G/F to the DDH problem in G exists. Indeed, a DDH challenge in G/F can be lifted into $\psi(\mathrm{G}/\mathrm{F}) \subset \mathrm{G}$. But $\mathrm{G} = \psi(\mathrm{G}/\mathrm{F}) \times \mathrm{F}$, so the reduction has to fill the F−part to keep the DDH challenge's form. This seems impossible with a non-negligeable advantage.

C.2.2. Examples

Let G be the group of quadratic residues modulo N^2 where $N = (2p' + 1)(2q' + 1)$ is the product of two safe primes. In this case, the order of G is $Np'q'$. The subgroup H of order N of G is generated by $1 + N$, and since $(1 + N)^k \equiv 1 + kN \pmod{N^2}$, the DL problem is easy in H (cf. [Pai99]). If the factorization of N is known, then DDH problem in G can not be hard (cf. Remark C−1). This inspired [BCP03, Theorem 4] where the factorization acts as a second trapdoor to an Elgamal-like protocol in G. A generalization of this protocol is given in the next subsection.

Now, let G be the group of quadratic residues modulo p^2 where $p = 2p' + 1$ is a safe prime. In this case, the DL problem is easy in the subgroup of order p generated by $1 + p$. The order of G is pp' and it can not be hidden from the description of G (*i.e.*, the integer p^2). As a result, the PDL and DDH problems are easy. In [CPP06], the partial logarithm of an element is called the *class* of an element. They define a variant of the DDH problem, namely the *Decision Class Diffie-Hellman problem*, which is believed to be intractable in such a group G. From that problem, [CPP06] derived a modification of Elgamal which, unfortunately, is *partially* homomorphic: it only supports the addition of a constant.

C.2.3. A Generic Linearly Homomorphic Encryption Scheme

From a DDH group with an easy DL subgroup, we can devise generically a linearly homomorphic encryption scheme. An Elgamal type scheme is used in G, with plaintext message $m \in \mathbf{Z}/p\mathbf{Z}$ mapped to $f^m \in F$. The resulted scheme is linearly homomorphic. Thanks to the Solve algorithm, the decryption does not need a complex DL computation. We depict this scheme in Fig. C.1. Note that the outputs n and s of Gen are not used in the algorithms.

Let us prove the homomorphic property of the scheme. Let us consider an output of the EvalSum algorithm on an input corresponding to encryptions of m and m'. Due to Elgamal's multiplicativity, the first line of the decryption algorithm applied on this output gives $M = f^m f^{m'} = f^{m+m' \bmod p}$ as f as multiplicative order p. As a consequence, the decryption process indeed returns $m + m' \mod p$, and the EvalSum algorithm gives a random encryption of $m + m' \pmod p$ (in the sense that it has the same output distribution than the encryption algorithm on the input $m + m' \pmod p$). The same argument works for the EvalScal algorithm, with any scalar $\alpha \in \mathbf{Z}/p\mathbf{Z}$.

C.2.4. Security

The total break of our scheme (tb − cpa attack) consists in finding x from (B, p, g, g^x, f), *i.e.*, in computing a discrete logarithm in G. From Theorem C−2, this is harder than computing a discrete logarithm in G/F.

Theorem C−3. *The scheme described in Fig. C.1 is one-way under chosen plaintext attack* (ow − cpa) *if and only if the Lift Diffie-Hellman* (LDH) *problem is hard (so if and only if the partial discrete logarithm problem* (PDL) *is hard).*

```
$\mathsf{KeyGen}(1^\lambda)$
  1. $(B, n, p, s, g, f, G, F) \xleftarrow{\$} \mathsf{Gen}(1^\lambda, 1^\mu)$
  2. Pick[a] $x \xleftarrow{\$} \{0, \dots, Bp-1\}$ and set $h \leftarrow g^x$
  3. Set $pk \leftarrow (B, p, g, h, f)$ and $sk \leftarrow x$.
  4. Return $(pk, sk)$

$\mathsf{Encrypt}(1^\lambda, pk, m)$
  1. Pick $r \xleftarrow{\$} \{0, \dots, Bp-1\}$
  2. Compute $c_1 \leftarrow g^r$
  3. Compute $c_2 \leftarrow f^m h^r$
  4. Return $(c_1, c_2)$
```

[a] As n will be unknown in the sequel, x is picked at random in $\{0, \dots, Bp-1\}$

```
$\mathsf{Decrypt}(1^\lambda, pk, sk, (c_1, c_2))$
  1. Compute $M \leftarrow c_2 / c_1^x$
  2. $m \leftarrow \mathsf{Solve}(p, g, f, G, F, M)$
  3. Return $m$

$\mathsf{EvalSum}(1^\lambda, pk, (c_1, c_2), (c_1', c_2'))$
  1. Compute $c_1'' \leftarrow c_1 c_1'$, $c_2'' \leftarrow c_2 c_2'$
  2. Pick $r \xleftarrow{\$} \{0, \dots, Bp-1\}$
  3. Return $(c_1'' g^r, c_2'' h^r)$

$\mathsf{EvalScal}(1^\lambda, pk, (c_1, c_2), \alpha)$
  1. Compute $c_1' \leftarrow c_1^\alpha$ and $c_2' \leftarrow c_2^\alpha$
  2. Pick $r \xleftarrow{\$} \{0, \dots, Bp-1\}$
  3. Return $(c_1' g^r, c_2' h^r)$
```

Figure C.1: A generic linearly homomorphic encryption scheme

Proof. From the equivalence of Theorem C−1, it suffices to prove the equivalence between the ow − cpa security and the hardness of the LDH problem. Let us consider $(c_1, c_2) = (g^r, f^m h^r)$, a ciphertext with the public key $h = g^x$. Then as $\pi(c_2) = \pi(h^r) = \pi(g^{xr})$, the triplet $(h, c_1, \pi(c_2))$ is an LDH challenge. Given a LDH oracle, we obtain $Z = g^{xr} = h^r$ and recover m by running Solve on c_2/Z.

Conversely let $(X, Y, Z') = (g^x, g^y, \pi(Z))$ be an LDH challenge with $Z = g^{xy}$. From this triplet, one can set X as public key and construct the ciphertext $(c_1, c_2) = (Y, f^r Z'_\ell)$ where $Z'_\ell \in \pi^{-1}(Z')$ and r is a random element of $\mathbf{Z}/p\mathbf{Z}$. As $\pi(c_2) = Z' = \pi(Z)$, one as $c_2 = f^m Z$ for an element $m \in \mathbf{Z}/p\mathbf{Z}$. As a result, (c_1, c_2) is a correct ciphertext of m, and a decryption oracle would respond m from which we can compute c_2/f^m and recover Z. □

Theorem C − 4. *The scheme described in Fig. C.1 is semantically secure under chosen plaintext attacks (*ind − cpa*) if and only the decisional Diffie-Hellman problem is hard in* G.

Proof. Let's construct a reduction $\mathscr{R}$ that solve the DDH assumption using an efficient ind−cpa adversary $\mathscr{A}$. $\mathscr{R}$ takes as input a DDH instance $(B, p, g, f, G, F, X, Y, Z)$ and sets $pk = (B, p, g, X, f)$. When $\mathscr{A}$ requests an encryption of one of his choice of challenge messages m_0 and m_1, $\mathscr{R}$ flips a bit b encrypts m_b as $(Y, f^{m_b}Z)$ and sends this ciphertext as its answer to $\mathscr{A}$. If Z was not a random element, this ciphertext would be indistinguishable from a true encryption of m_b because of the choice of the bound B, and $\mathscr{A}$ will correctly answer with its (non-negligeable) advantage ϵ. Otherwise, the encryption is independent of the message and $\mathscr{A}$'s advantage to distinguish is 1/2. Therefore, the reduction returns one if and only $\mathscr{A}$ correctly guessed b and has advantage $\epsilon/2$ to solve the DDH assumption. □

C.3. A Linearly Homomorphic Encryption from DDH

We prove that, somewhat like in Paillier's encryption scheme [Pai99] within $\mathbf{Z}/N^2\mathbf{Z}$, a subgroup with an easy discrete logarithm problem exists in class groups of imaginary quadratic fields, and it allows to design a new linearly homomorphic encryption scheme. We refer the reader to Appendix C.B for background on class groups of imaginary quadratic fields and their use in Discrete Logarithm based cryptography.

C.3.1. A Subgroup with an Easy DL Problem

The next proposition, inspired by Theorem A−2 of [CL09], establish the existence of a subgroup of a class group of an imaginary quadratic fields where the DL problem is easy.

Proposition C−1. *Let Δ_K be a fundamental discriminant with $\Delta_K \equiv 1 \pmod 4$ of the form $\Delta_K = -pq$ where p is an odd prime and q a non-negative integer prime to p such that $q > 4p$. Let $\mathfrak{f} = (p^2, p)$ be an ideal of $\mathscr{O}_{\Delta_p}$, the order of discriminant $\Delta_p = \Delta_K p^2$. Denote by $f = [\mathfrak{f}]$ the class of $\mathfrak{f}$ in $C(\mathscr{O}_{\Delta_p})$. For $m \in \{1, \dots, p-1\}$, $Red(f^m) = (p^2, L(m)p)$ where $L(m)$ is the odd integer in $[-p, p]$ such that $L(m) \equiv 1/m \pmod p$. Moreover, f is a generator of the subgroup of order p of $C(\mathscr{O}_{\Delta_p})$.*

Proof. We consider the surjection $\bar{\varphi}_p : C(\mathscr{O}_{\Delta_p}) \longrightarrow C(\mathscr{O}_{\Delta_K})$. From Lemma A−1 of [CL09] and [Cox99, Proposition 7.22 and Theorem 7.24], the kernel of $\bar{\varphi}_p$ is isomorphic to $(\mathscr{O}_{\Delta_K}/p\mathscr{O}_{\Delta_K})^\times/(\mathbf{Z}/p\mathbf{Z})^\times$. As $p \mid \Delta_K$, the group $(\mathscr{O}_{\Delta_K}/p\mathscr{O}_{\Delta_K})^\times$ is isomorphic to the group $(\mathbf{F}_p[X]/(X^2))^\times$. This group contains $p(p-1)$ elements of the form $a + b\sqrt{\Delta_K}$ where $a \in (\mathbf{Z}/p\mathbf{Z})^\times$ and $b \in \mathbf{Z}/p\mathbf{Z}$. Now let us consider the quotient group $(\mathscr{O}_{\Delta_K}/p\mathscr{O}_{\Delta_K})^\times/(\mathbf{Z}/p\mathbf{Z})^\times$ where $[x] = [y]$ with $x, y \in (\mathscr{O}_{\Delta_K}/p\mathscr{O}_{\Delta_K})^\times$ if and only if there exists $\lambda \in (\mathbf{Z}/p\mathbf{Z})^\times$ such that $x = \lambda y$. This quotient is cyclic of order p and a system of representatives is $[1]$ and $[a + \sqrt{\Delta_K}]$ where a is an element of $(\mathbf{Z}/p\mathbf{Z})^\times$. Let $g = [1 + \sqrt{\Delta_K}]$, one has $g^m = [1 + m\sqrt{\Delta_K}] = [L(m) + \sqrt{\Delta_K}]$ for all $m \in \{1, \dots, p-1\}$ and $g^p = [1]$.

Let $\alpha_m = \frac{L(m)+\sqrt{\Delta_K}}{2} \in \mathscr{O}_{\Delta_K}$. Then α_m is a representative of the class g^m. The element g^m maps to the class $[\varphi_p^{-1}(\alpha_m \mathscr{O}_{\Delta_K})]$ of the kernel of $\bar{\varphi}_p$. From [BTW95, Proposition 2.9], one can see that $\alpha_m \mathscr{O}_{\Delta_K} = (N(\alpha_m), -L(m) \mod 2N(\alpha_m))$ where the remainder is computed from the centered euclidean division. Now,

$$\varphi_p^{-1}(\alpha_m \mathscr{O}_{\Delta_K}) = \left(N(\alpha_m), -L(m)p \mod 2N(\alpha_m)\right).$$

As $\mathrm{N}(\alpha_m) = \frac{\mathrm{L}(m)^2-\Delta_\mathrm{K}}{4}$ and $q > 4p$, it follows first that $p^2 < \mathrm{N}(\alpha_m)$ and moreover that $-\mathrm{L}(m)p \mod 2\mathrm{N}(\alpha_m) = -\mathrm{L}(m)p$. As a consequence, this ideal $\varphi_p^{-1}(\alpha_m\mathscr{O}_{\Delta_\mathrm{K}})$ corresponds to the quadratic form $\left(\frac{\mathrm{L}(m)^2-\Delta_\mathrm{K}}{4}, -\mathrm{L}(m)p, p^2\right)$, of discriminant Δ_p. Moreover this form is equivalent to the form $(p^2, \mathrm{L}(m)p, \frac{\mathrm{L}(m)^2-\Delta_\mathrm{K}}{4})$ which corresponds to the ideal $(p^2, \mathrm{L}(m)p)$. Eventually, this ideal of $\mathscr{O}_{\Delta_p}$ is reduced as $|\mathrm{L}(m)p| < p^2 < \sqrt{|\Delta_p|/2}$, where the second inequality holds because $q > 4p$. Consequently, if $\mathfrak{f} = (p^2, p)$, then $[\mathfrak{f}]$ generates the kernel of $\bar{\varphi}_p$ as $[\mathfrak{f}] = [\varphi_p^{-1}(\alpha_1\mathscr{O}_{\Delta_\mathrm{K}})]$. Moreover, $[\mathfrak{f}]^m = [\varphi^{-1}(\alpha_m\mathscr{O}_{\Delta_\mathrm{K}})]$ so $\mathsf{Red}([\mathfrak{f}]^m) = (p^2, \mathrm{L}(m)p)$, for $m \in \{1, \dots, p-1\}$. □

We devise, in Fig. C.2, a new DDH group with an easy DL subgroup in class groups of imaginary quadratic fields, by assuming the difficulty of the DDH problem. In the Gen algorithm, we first construct a fundamental discriminant $\Delta_\mathrm{K} = -pq$ such that the 2-Sylow subgroup of $C(\Delta_\mathrm{K})$ is of order 2 (cf. Appendix C.B.3). Then, using [HJPT98, Subsection 3.1]'s method, we construct an ideal $\mathfrak{r}$ of $\mathscr{O}_{\Delta_\mathrm{K}}$ of norm r, where r is a prime satisfying $\left(\frac{\Delta_\mathrm{K}}{r}\right) = 1$. We then assume, as in the previous implementations of Elgamal (cf. Appendix C.B.4) that the class $[\mathfrak{r}^2]$ will be of order s, an integer of the same order of magnitude than the odd part, $h(\Delta_\mathrm{K})/2$. Due to our choice of p and q, pq is 2λ-bit integer, and as s is close to $\sqrt{|\Delta_\mathrm{K}|}$ (cf. Appendix C.B.3), it will be a λ-bit integer.

If $\mu > 80$, following the Cohen-Lenstra heuristics, the probability that p divides $h(\Delta_\mathrm{K})$ and s is negligible. Therefore, we can assume that $\gcd(p, h(\Delta_\mathrm{K})) = 1$. We consider the non-maximal order $\mathscr{O}_{\Delta_p}$ of discriminant $p^2\Delta_\mathrm{K}$ as in Proposition C−1. The fact that $\lambda \geqslant \mu+2$ ensures that $q > 4p$. As a result, the subgroup F generated by f gives an easy DL subgroup. The morphism $\bar{\varphi}_p$ defined in Appendix C.B.1 plays the role of the surjection π between $C(\mathscr{O}_{\Delta_p})$ and $C(\mathscr{O}_{\Delta_p})/\mathrm{F} \simeq C(\mathscr{O}_{\Delta_\mathrm{K}})$, which is computable in polynomial time, knowing p (cf. [HJPT98, Algorithm 3]). Moreover, still with the knowledge of p, it is possible to lift elements of $C(\mathscr{O}_{\Delta_\mathrm{K}})$ in $C(\mathscr{O}_{\Delta_p})$, using [HPT99, Algorithm 2]. We can then apply the injective morphism of Lemma C−3 on $[\mathfrak{r}^2]$ to get a class of $C(\Delta_p)$ with the same order s and multiply this class by f^k where $k \overset{\$}{\leftarrow} \{1, p-1\}$. As $\gcd(p, s) = 1$ the result, g is of order ps (this procedure to get an element of order ps was also used in the proof of Theorem C−2). Note that g is still a square of $C(\Delta_p)$: as the map of Lemma C−3 is a morphism, the lift of $[\mathfrak{r}^2]$ gives a square of $C(\Delta_p)$. Moreover, F is a subgroup of the squares: $f = (f^{2^{-1} \bmod p})^2$ as p is odd. As a consequence, g is a square as it is a product of two squares.

Eventually, we take $\mathrm{B} = \lceil|\Delta_\mathrm{K}|^{3/4}\rceil$. According to Lemma C−4 in Appendix C.C, the statistical distance of $\{g^r, r \overset{\$}{\leftarrow} \{0, \dots, \mathrm{B}p-1\}\}$ to the uniform distribution is upper bounded by $ps/(4p\mathrm{B}) = s/(4\lceil|\Delta_\mathrm{K}|^{3/4}\rceil)$. By Equation J.1 in Appendix C.B.3 , this is less than

$$\frac{\log|\Delta_\mathrm{K}|}{4\pi\lceil|\Delta_\mathrm{K}|^{1/4}\rceil} \in \tilde{\mathscr{O}}(2^{-\lambda/2})$$

```
Gen(1^λ, 1^μ)
  1. Assume $\lambda \geqslant \mu + 2$
  2. Pick $p$ a random $\mu$-bits prime and $q$ a random $(2\lambda - \mu)$ prime such that $pq \equiv -1 \pmod{4}$ and $(p/q) = -1$.
  3. Set $\Delta_K \leftarrow -pq$ and $\Delta_p \leftarrow p^2\Delta_K$.
  4. Set $f \leftarrow [(p^2, p)]$ in $C(\Delta_p)$ and $F = \langle f \rangle$
  5. Let $r$ be a small prime, with $r \neq p$ and $\left(\frac{\Delta_K}{r}\right) = 1$ and set $\mathfrak{r}$ an ideal lying above $r$.
  6. Let $k \xleftarrow{\$} \{1, p-1\}$ and set $g \leftarrow [\varphi_p^{-1}(\mathfrak{r}^2)]^p f^k$ in $C(\Delta_p)$ and $G \leftarrow \langle g \rangle$
  7. Let $B \leftarrow \lceil |\Delta_K|^{3/4} \rceil$
  8. Return $(B, \emptyset, p, \emptyset, g, f, G, F)$
```

```
Solve(B, p, g, f, G, F, X)
  1. Parse Red(X) as $(p^2, \tilde{x}p)$
  2. If fails return $\perp$
  3. Else return $\tilde{x}^{-1} \pmod{p}$
```

Figure C.2: A new DDH Group with an Easy DL Subgroup

which is a negligible function of λ. As a consequence, the distribution

$$\{g^r, r \xleftarrow{\$} \{0, \dots, Bp - 1\}\},$$

is statistically indistinguishable from the uniform distribution in $G = \langle g \rangle$. In practice, for performance issue, one can take a better bound for B, for example

$$B = 2^{80}\lceil \log(|\Delta_K|)|\Delta_K|^{1/2}/(4\pi)\rceil,$$

which makes the statistical distance less than 2^{-80}.

C.3.2. The new protocol

The DDH group with an easy DL subgroup of Fig. C.2 gives rise to a linearly homomorphic encryption scheme in quadratic fields, using the generic construction of Fig. C.1. Compared to previous solutions based on a similar construction ([BCP03]), this scheme is only based on the difficulty of the discrete logarithm in G, and does not rely on the difficulty of factorization.

In practice, the best attack against the scheme consists in retrieving the private key, *i.e.*, in computing a discrete logarithm. As said in Appendix C.B.3, the problems of computing discrete logarithm in $C(\mathcal{O}_{\Delta_K})$ and computing $h(\mathcal{O}_{\Delta_K})$ have similar complexity.

Given oracle for both problems, one can compute discrete logarithm in $C(\mathcal{O}_{\Delta_p})$ and totally break the scheme. Indeed, if $s = h(\mathcal{O}_{\Delta_K})$, given g and $h = g^x$, we can compute $\bar{\varphi}_p(g)$ and $\bar{\varphi}_p(h) = \bar{\varphi}_p(g)^{x \bmod s}$. The oracle for discrete logarithm in $C(\mathcal{O}_{\Delta_K})$ gives x mod s. Furthermore, as shown in Lemma C−1, if s is known the PDL problem is easy, so one can compute x mod p and we get x as $\gcd(p, s) = 1$ with the Chinese remainder theorem. Moreover, finding $h(\mathcal{O}_{\Delta_K})$ or the multiplicative order of g can be sufficient: knowing $s = h(\mathcal{O}_{\Delta_K})$ breaks the PDL problem (cf. Lemma C−1) and the one wayness of the scheme by Theorem C−3.

C.4. Extensions

C.4.1. Removing the Condition on the Relative Size of p and q

To have a polynomial Solve algorithm, we impose that $q > 4p$, in order that the reduced elements of $\langle f \rangle$ are the ideals of norm p^2. If we want a large message space, for example 2048 bits (as for the cryptosystem of Paillier or the scheme of [BCP03] with a 2048 bit RSA integer), this means that p has 2048 bits, so $|\Delta_p| = p^3q > 4p^4$ has more than 8194 bits and $|\Delta_K| = pq > 4p^2$ has more than 4098 bits. Therefore we loose our advantage over factoring based schemes, as we only need a discriminant Δ_K of 1348 bits to have the same security than a 2048 bit RSA integer (cf. Appendix C.B.3).

For example, suppose that we work with $\Delta_K = -p$. In the order $\mathcal{O}_{\Delta_p}$ of discriminant $\Delta_p = p^2\Delta_K = -p^3$, the ideals of norm p^2 are no longer reduced. However, we can still have a polynomial time algorithm to solve the discrete logarithm in $\langle f \rangle$ where $f = [(p^2, p)]$. From the proof of Proposition C−1, f still generate the subgroup of order p, and for $k \in \{1, \dots, p-1\}$, the class f^k still contains a non reduced ideal $(p^2, L(k)p)$ where $L(k)$ is defined as in Proposition C−1. We can use the main result of [CL09] constructively to find this non reduced ideal that will disclose the discrete logarithm k given the reduced element of the class f^k. The idea is to lift this reduced element in a class group of a suborder where the ideals of norm p^2 are reduced. Let $\Delta_{p^2} = p^4\Delta_K$. For $p > 4$, we have $p^2 < \sqrt{|\Delta_{p^2}|/2}$ so the ideals of norm p^2 are reduced. We lift an element of $\mathcal{O}_{\Delta_p}$ in $\mathcal{O}_{\Delta_{p^2}}$ by computing $[\varphi_p^{-1}(\cdot)]^p$ on a representative ideal prime to p (we can use [HJPT98, Algorithm 1] to find an ideal prime to p in a given class). This map is injective, so applied on f we get a class f_ℓ of order p in $C(\mathcal{O}_{\Delta_{p^2}})$. Moreover, this class is in the kernel of the map $\bar{\varphi}_{p^2}$ from $C(\mathcal{O}_{\Delta_{p^2}})$ to $C(\mathcal{O}_{\Delta_K})$, and an easy generalization of Proposition C−1 shows that the subgroup of $C(\mathcal{O}_{\Delta_{p^2}})$ generated by f_ℓ is also generated by $[(p^2, p)]$. As a result, if $h = f^x$ in $C(\mathcal{O}_{\Delta_p})$, we have $h_\ell = [\varphi_p^{-1}([h])]^p = ([\varphi_p^{-1}([f])]^p)^x = f_\ell^x$ and x can be computed as $x = y/z$ where y is the discrete logarithm of h_ℓ in basis $[(p^2, p)]$ and y is the discrete logarithm of f_ℓ in basis $[(p^2, p)]$. Both logarithms can be computed as in $C(\mathcal{O}_{\Delta_p})$.

This variant can also work with $\Delta_K = -pq$ and $q < 4p$, so p can be chosen independently from the security level, with the restriction that p must have at least 80 bits.

C.4.2. A Faster Variant

We can change the KeyGen algorithm as follows: g is now in the class group of the maximal order (*i. e.*, g is the class of $\mathfrak{r}^2$) and we set $h = g^x$ where x is the secret key and the computation is done in $C(\mathcal{O}_{\Delta_K})$. Let us denote by $\psi : C(\mathcal{O}_{\Delta_K}) \to C(\mathcal{O}_{\Delta_p})$ the injective morphism of Lemma C−3, that computes $[\varphi_p^{-1}(\cdot)]^p$ on a representative ideal prime to p.

To encrypt $m \in \mathbf{Z}/p\mathbf{Z}$, we compute $c_1 = g^r$ and $c_2 = f^m\psi(h^r)$ in $C(\mathcal{O}_{\Delta_p})$. To decrypt, we first compute c_1^x and lift it, by computing $c_1' = \psi(c_1^x)$ in $C(\mathcal{O}_{\Delta_p})$. Then we retrieve $f^m = c_2/c_1'$. This variant can be viewed as a mix of an Elgamal cryptosystem in $C(\mathcal{O}_{\Delta_K})$ (lifted in $C(\mathcal{O}_{\Delta_p})$ by applying ψ) and of a cryptosystem based on the subgroup decomposition problem using the direct product between $\psi(\langle g\rangle)$ and $\langle f\rangle$. The advantage of this variant is that ciphertexts are smaller (c_1 is in $C(\mathcal{O}_{\Delta_K})$ instead of $C(\mathcal{O}_{\Delta_p})$) and that computations are faster: encryption performs two exponentiations in $C(\mathcal{O}_{\Delta_K})$ instead of $C(\mathcal{O}_{\Delta_p})$ and one lift (which computational cost is essentially the exponentiation to the power p). Decryption similarly involves one exponentiation in $C(\mathcal{O}_{\Delta_K})$ instead of $C(\mathcal{O}_{\Delta_p})$ and a lift. However, the semantic security is now based on a non standard problem. Let g be a generator of a subgroup of $C(\mathcal{O}_{\Delta_K})$ of order s. After having chosen m, the adversary is asked to distinguished the following distributions : $\{(g^x, g^y, \psi(g^{xy})), x, y \overset{\$}{\leftarrow} \mathbf{Z}/s\mathbf{Z}\}$ and $\{(g^x, g^y, \psi(g^{xy})f^m), x, y \overset{\$}{\leftarrow} \mathbf{Z}/s\mathbf{Z}\}$. The total break is equivalent to the DL problem in $C(\mathcal{O}_{\Delta_K})$.

C.5. Performances and Comparisons.

We now compare the efficiency of our cryptosystem with some other linearly homomorphic encryptions schemes, namely the system of Paillier and the one from [BCP03]. The security of the Paillier cryptosystem is based on the factorization problem of RSA integers, while [BCP03] is based on both the factorization and the DL problems. For our scheme, the best attack consists in computing DL in $C(\mathcal{O}_{\Delta_K})$ or in computing $h(\mathcal{O}_{\Delta_K})$ and both problems have similar complexity.

As said in Appendix C.B.3, in [BJS10], the Discrete Logarithm problem with a discriminant Δ_K of 1348 (resp. 1828 bits) is estimated as hard as factoring a 2048 (resp. 3072 bits) RSA integer n. In Fig. C.1, we give the timings in ms of the time to perform an encryption and decryption for the three schemes. Concerning Paillier, for encryption and decryption, the main operation is an exponentiation of the form $x^k \mod n^2$ where k has the same bit length as n. Concerning [BCP03], which has an Elgamal structure, two exponentiations of the form $x^k \mod n^2$ with k an integer of the same bit length as n^2 are used for encryption and one for decryption. Our scheme has also this structure with two exponentiations for encryption and one for decryption. Decryption also involves an inversion modulo p. The exponentiations are made in $C(\mathcal{O}_{\Delta_p})$ with $\Delta_p = p^2\Delta_K$. The size of the exponent is bounded by Bp where we have seen that B can be chosen roughly of the bit size of $\sqrt{\Delta_K}$ plus 80 bits. For a same security level, our scheme is thus more efficient for a small p.

The timings where performed with Sage 6.3 on a standard laptop with a straightforward implementation. The exponentiation in class group uses a PARI/GP function (qfbnupow). We must stress that *this function is far less optimized than the exponentiation in $\mathbf{Z}/n\mathbf{Z}$, so there is a huge bias in favor of BCP and Paillier*. A more optimized implementation would give much better results for our system. Nevertheless, we see that for a 2048 bits modulus, our cryptosystem is already faster than the protocol from [BCP03]. Moreover, for stronger securities, our system will be faster, as asymptotically, the factorization algorithms have complexity $L(1/3, \cdot)$ whereas the algorithms for class groups of quadratic fields have complexity $L(1/2, \cdot)$. Moreover the multiplication modulo n and the composition of quadratic forms have both quasi linear complexity [Sch91b]. As shown in Table C.1, already with a 3072 bits modulus our cryptosystem is competitive: faster than Paillier for decryption. For a very high security level (7680 bits modulus), our system would be twice as fast as Paillier for encryption, for messages of 512 bits. We also give timings of our faster variant of Subsection C.4.2. For a same security level, this variant becomes more interesting when the message space grows. In Table C.1, we see that even with a naive implementation, our system is competitive for message space up to 256 bits (resp. 912 bits) for 2048 bits security (resp. for 3072 bits security).

Note that a medium size message space can be sufficient for applications. For example, our system may be used as in [CGS97] to design a voting scheme. For a yes/no pool, a voter encrypts 0 (resp. 1) to vote no (resp. to vote yes). By combining all the ciphertexts, the election manager would get an encryption of the sum of the vote modulo p. Decryption allows to decide the result if the number of voters ℓ satisfies $\ell < p$. So a 80-bit p is largely sufficient as $2^{80} \approx 10^{24}$. With Elgamal, in [CGS97], the discrete logarithm in decryption involves a baby-step giant-step computation of time $\mathcal{O}(\sqrt{\ell})$ (so a very low number of voters can be handled) whereas a single inversion modulo p is needed for our scheme. For a multi-candidate election system with m candidates and ℓ voters, one votes for the i^{th} candidate by encrypting ℓ^i. The tally is decrypted with a decomposition in base ℓ, so we must have $\ell^m < p$. With a 256 bit integer p, we can have for example 2^{16} voters and 16 candidates, which is the good order of magnitude for real life elections, for which there are around a thousand registered voters by polling stations.

C.6. Conclusion

We proposed the first linearly homomorphic encryption whose security relies on a sole Diffie-Hellman-like assumption. Our construction crucially uses the algebraic properties of the class group of a non maximal order of an imaginary quadratic field. They make it possible to avoid the factorization assumption and to have $\mathbf{Z}/p\mathbf{Z}$ as the set of messages. Other improvements than those we presented are possible: we can gain efficiency using the Chinese Remainder Theorem using discriminant of the form $\Delta_K = -(\prod_{i=1}^{n} p_i)q$, and generalizing *à la* Damgård and Jurik (cf. [DJ01]), with discriminants of

Cryptosystem	Parameter	Message Space	Encryption (ms)	Decryption (ms)
Paillier	2048 bits modulus	2048 bits	**28**	**28**
AC:BreCatPoi03	2048 bits modulus	2048 bits	107	54
New Proposal	1348 bits Δ_K	80 bits	93	49
Variant Subsec. C.4.2	1348 bits Δ_K	80 bits	82	45
Variant Subsec. C.4.2	1348 bits Δ_K	256 bits	105	68
Paillier	3072 bits modulus	3072 bits	**109**	109
AC:BreCatPoi03	3072 bits modulus	3072 bits	427	214
New Proposal	1828 bits Δ_K	80 bits	179	91
Variant Subsec. C.4.2	1828 bits Δ_K	80 bits	145	**78**
Variant Subsec. C.4.2	1828 bits Δ_K	512 bits	226	159
Variant Subsec. C.4.2	1828 bits Δ_K	912 bits	340	271

Table C.1: Efficiency Comparison of Linearly Homomorphic Encryption Schemes

the form $\Delta_{p^t} = p^{2t}\Delta_K$, with $\Delta_K = -pq$ and $t \geqslant 1$ to enlarge the message space to $\mathbf{Z}/p^t\mathbf{Z}$ without losing the homomorphic property. A non-trivial adaptation may also be possible with real quadratic fields. Experiments show that our protocol is competitive with existing ones.

Acknowledgement: This work has been supported in part by ERC Starting Grant ERC-2013-StG-335086-LATTAC and by the financial support from the French State, managed by the French National Research Agency (ANR) in the frame of the "Investments for the future" Programme IdEx Bordeaux (ANR-10-IDEX-03-02), Cluster of excellence CPU.

C.A. Public-key Encryption: Definitions

Encryption Scheme: Definition.

Let λ be an integer. An encryption scheme is a tuple of algorithms $\Pi = (\mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$. The probabilistic polynomial-time key generation algorithm KeyGen takes a security parameter λ in unary as input and returns a pair (pk, sk) of public key and the matching secret key. The probabilistic polynomial-time encryption algorithm Encrypt takes a security parameter, a public key pk and a message m as inputs, and outputs a ciphertext c. The deterministic polynomial-time Decrypt decryption algorithm takes a security parameter, a secret key sk and a ciphertext c and returns either a message m or the symbol $\perp$ which indicates the invalidity of the ciphertext. The scheme must be *correct*, which means that for all security parameters λ, and for all messages m, if $(pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$ then $\mathsf{Decrypt}(1^\lambda, sk, \mathsf{Encrypt}(1^\lambda, pk, m)) = m$ with probability (taken on all internal random coins and random choices).

Encryption Scheme: Security.

The *total break* of an encryption scheme is declared if an attacker can recover the secret key from (at least) the public key. Therefore any probabilistic polynomial-time Turing machine $\mathscr{B}$ must have a success in recovering the public key arbitrarily small, where the *success* is defined, for an integer λ, as the probability

$$\Pr[(pk, sk) \leftarrow \Pi.\mathsf{KeyGen}(1^\lambda) : \mathscr{B}(pk) = sk].$$

The intuitive security notion expected from an encryption scheme is *one-wayness*, which means that, given only the public data, an adversary cannot recover the message corresponding to a given ciphertext. More precisely, any probabilistic polynomial-time Turing machine $\mathscr{A}$ (the *attacker*) has a success in inverting the encryption algorithm arbitrarily small, where the *success* is defined, for an integer λ, as the probability

$$\Pr[(pk, sk) \leftarrow \Pi.\mathsf{KeyGen}(1^\lambda) : \mathscr{A}(pk, \Pi.\mathsf{Encrypt}(1^\lambda, sk, m)) = m].$$

The previous definition supposes that the attacker has no more information that the public key : the attacker is said to do a *chosen-plaintext attack* (since he can produce the ciphertext of message of his choice). If he has access to a decryption oracle, the attack is said be a *chosen-ciphertext attack*.

An encryption scheme must indeed reach a stronger notion of security : it must have *semantic security* (aka *indistinguishability*). This means that an attacker is computationally unable to distinguish between two messages, chosen by himself, which one has been encrypted, with a probability significantly better than one half. The *indistinguishability game* is formally defined as:

```
$\boxed{\text{Experiment } \mathbf{Exp}_\Pi^{\mathsf{ind\text{-}atk}}(\mathscr{A})}$
$(pk, sk) \leftarrow \Pi.\mathsf{KeyGen}(1^\lambda)$
$(m_0, m_1, s) \leftarrow \mathscr{A}_1^{\mathscr{O}_1}(pk)$
$b^\star \overset{\$}{\leftarrow} \{0,1\}$
$c^\star \leftarrow \Pi.\mathsf{Encrypt}(pk, m_{b^\star})$
$b \leftarrow \mathscr{A}_2^{\mathscr{O}_2}(s, c^\star)$
Return $b == b^\star$
```

with

- atk = cpa and
  - $\mathscr{O}_1 = \emptyset$
  - $\mathscr{O}_2 = \emptyset$
- atk = cca1 and
  - $\mathscr{O}_1 = \Pi.\mathsf{Decrypt}(\mathsf{params}, sk, \cdot)$
  - $\mathscr{O}_2 = \emptyset$
- atk = cca2 and
  - $\mathscr{O}_1 = \Pi.\mathsf{Decrypt}(\mathsf{params}, sk, \cdot)$
  - $\mathscr{O}_2 = \Pi.\mathsf{Decrypt}(\mathsf{params}, sk, \cdot)$

where the adversary $\mathscr{A}$ is modelled as a 2-stage PPTM $(\mathscr{A}_1, \mathscr{A}_2)$. The *advantage* of the attacker is then defined as

$$\mathsf{Adv}_\Pi^{\mathsf{ind\text{-}atk}}(\mathscr{A}) = \left| \Pr\left(\mathbf{Exp}_\Pi^{\mathsf{ind\text{-}atk}}(\mathscr{A}) = 1\right) - \frac{1}{2} \right|.$$

Linearly Homomorphic Encryption

Let suppose that the set of plaintexts $\mathscr{M}$ (resp. the set of ciphertexts $\mathscr{C}$) is equipped with an additive (resp. a multiplicative) group structure. An encryption scheme Π is said to be *homomorphic* if $\forall \lambda \in \mathbf{N}$, $\forall (pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$, $\forall m_1, m_2 \in \mathscr{M}$, if $c_1 \leftarrow \mathsf{Encrypt}(pk, m_1)$ and $c_2 \leftarrow \mathsf{Encrypt}(pk, m_2)$, then the product c_1c_2 is a valid encryption of $m_1 + m_2$. The exponentiation of a ciphertext c_1 to a power α is as well a valid encryption of αm_1. The formal definition of a linearly homomorphic encryption includes two algorithms EvalSum and EvalScal that fulfills the corresponding correctness property.

Of course, to achieve semantic security Π.Encrypt has to be probabilistic, but even though, the highest level of indistinguishability an homomorphic encryption scheme can achieve is indeed $\mathsf{ind-cca1}$. As a matter of fact, an attacker will always win the cca2 game by querying, in the second phase, for instance $c^\star \cdot \mathsf{Encrypt}(pk, 0)$ to the decryption oracle: this one will answer with $m_{b^\star} + 0 = m_{b^\star}$.

C.B. Background on Imaginary Quadratic Fields

C.B.1. Imaginary Quadratic Fields and Class Group

Let $D < 0$ be a squarefree integer and consider the quadratic imaginary field $K = \mathbf{Q}(\sqrt{D})$. The *fundamental discriminant* Δ_K of K is defined as $\Delta_K = D$ if $D \equiv 1 \pmod 4$ and $\Delta_K = 4D$ otherwise. An *order* $\mathscr{O}$ in K is a subset of K such that $\mathscr{O}$ is a subring of K containing 1 and $\mathscr{O}$ is a free $\mathbf{Z}$-module of rank 2. The ring $\mathscr{O}_{\Delta_K}$ of algebraic integers in K is the *maximal* order of K. It can be written as $\mathbf{Z} + \omega_K\mathbf{Z}$, where $\omega_K = \frac{1}{2}(\Delta_K + \sqrt{\Delta_K})$. If we set $f = [\mathscr{O}_{\Delta_K} : \mathscr{O}]$ the *finite* index of any order $\mathscr{O}$ in $\mathscr{O}_{\Delta_K}$, then $\mathscr{O} = \mathbf{Z} + f\omega_K\mathbf{Z}$. The integer f is called the *conductor* of $\mathscr{O}$. The discriminant of $\mathscr{O}$ is then $\Delta_f = f^2\Delta_K$. Now, let $\mathscr{O}_\Delta$ be an order of discriminant Δ and $\mathfrak{a}$ be a nonzero ideal of $\mathscr{O}_\Delta$, its norm is $N(\mathfrak{a}) = |\mathscr{O}_\Delta/\mathfrak{a}|$. A *fractional* ideal is a subset $\mathfrak{a} \subset K$ such that $d\mathfrak{a}$ is an ideal of $\mathscr{O}_\Delta$ for $d \in \mathbf{N}$. A fractional ideal $\mathfrak{a}$ is said to be *invertible* if there exists an another fractional ideal $\mathfrak{b}$ such that $\mathfrak{ab} = \mathscr{O}_\Delta$. The *ideal class group* of $\mathscr{O}_\Delta$ is $C(\mathscr{O}_\Delta) = I(\mathscr{O}_\Delta)/P(\mathscr{O}_\Delta)$, where $I(\mathscr{O}_\Delta)$ is the group of invertible fractional ideals of $\mathscr{O}_\Delta$ and $P(\mathscr{O}_\Delta)$ the subgroup consisting of principal ideals. Its cardinality is the *class number* of $\mathscr{O}_\Delta$ denoted by $h(\mathscr{O}_\Delta)$. The group $\mathscr{O}_\Delta^\star$ of units in $\mathscr{O}_\Delta$ is equal to $\{\pm 1\}$ for all $\Delta < 0$, except when Δ is equal to -3 and -4 ($\mathscr{O}_{-3}^\star$ and $\mathscr{O}_{-4}^\star$ are respectively the group of sixth and fourth roots of unity)

A nonzero ideal $\mathfrak{a}$ of $\mathscr{O}_\Delta$, $\mathfrak{a}$ is said to be *prime to* f if $\mathfrak{a} + f\mathscr{O}_\Delta = \mathscr{O}_\Delta$. We denote by $I(\mathscr{O}_\Delta, f)$ the subgroup of $I(\mathscr{O}_\Delta)$ of ideals prime to f. The map $\varphi_f : I(\mathscr{O}_{\Delta_f}, f) \to I(\mathscr{O}_{\Delta_K}, f)$, $\mathfrak{a} \mapsto \mathfrak{a}\mathscr{O}_{\Delta_K}$ is an isomorphism. This map induces a surjection $\bar{\varphi}_f : C(\mathscr{O}_{\Delta_f}) \twoheadrightarrow C(\mathscr{O}_{\Delta_K})$. In our settings, we will use a prime conductor $f = p$ and consider $\Delta_p = p^2\Delta_K$, for a fundamental discriminant Δ_K divisible by p. The order of the kernel of $\bar{\varphi}_p$ is then given by the classical *analytic class number formula* (see for instance [BV07]): $\frac{h(\mathscr{O}_{\Delta_p})}{h(\mathscr{O}_{\Delta_K})} = p$ if $\Delta_K < -4$.

C.B.2. Representation of the Classes

Working with ideals modulo the equivalence relation of the class group is essentially equivalent to work with binary quadratic forms modulo $\mathsf{SL}_2(\mathbf{Z})$ (cf. [Coh00, Section 5.2]). Every (primitive) ideal $\mathfrak{a}$ of $\mathscr{O}_\Delta$ can be written as $\mathfrak{a} = \left(a\mathbf{Z} + \frac{-b+\sqrt{\Delta}}{2}\mathbf{Z}\right)$ with $a \in \mathbf{N}$ and $b \in \mathbf{Z}$ such that $b^2 \equiv \Delta \pmod{4a}$, and denoted by (a, b) for short. The norm of such an ideal is then a. This notation also represents the binary quadratic form $ax^2+bxy+cy^2$ with $b^2 - 4ac = \Delta$. This representation of the ideal is unique if the form is normal: $-a < b \leqslant a$.

An ideal is reduced if it is normal, and $a \leqslant c$, and $b \geqslant 0$ for $a = c$. Note that in every class of $\mathscr{O}_\Delta$-ideals there exists exactly one reduced ideal. We note $\mathsf{Red}(\mathfrak{a})$ the unique reduced ideal equivalent to an ideal $\mathfrak{a}$, or $\mathsf{Red}([\mathfrak{a}])$ the unique reduced ideal in the class $[\mathfrak{a}]$. From the theory of quadratic forms, we can efficiently compute $\mathsf{Red}(\mathfrak{a})$ given $\mathfrak{a}$. The algorithm, which is due to Gauss, is described in [Coh00, Algorithm 5.4.2 p. 243] and is called Red in this paper. In general, instead of working with classes, we will work with reduced ideals. The product of ideals is also efficiently computable with the composition of quadratic forms algorithm, see [Coh00, Algorithm 5.4.7 p. 243]. These two algorithms have quadratic complexity (and even quasi linear using fast arithmetic).

A crucial fact for our purpose is described in [Coh00, Lemma 5.3.4] and [BV07, Lemma 6.5.1]: a normal ideal $\mathfrak{a} = (a, b)$ with $|a| < \sqrt{|\Delta|/2}$ is reduced.

C.B.3. Class Number Computation and DL Problem

In 2000, Jacobson has described an index-calculus method to solve the discrete logarithm problem in class group of imaginary quadratic field of discriminant Δ_K [Jac00]. Various improvements have been proposed to this algorithm: In [BJS10], it is conjecture that a state of the art implementation has conjectured complexity $L_{|\Delta_K|}[1/2, o(1)]$. Moreover, the best known algorithm to compute class numbers of fundamental discriminant are again index-calculus method with the same complexity.

In [HM00a], Hamdy and Möller discuss the selection of a discriminant Δ_K such that the discrete logarithm problem in $C(\mathscr{O}_{\Delta_K})$ is as hard as in finite fields: It is advised to construct a fundamental discriminant Δ_K and to minimize to 2-Sylow subgroup of the class group. In our case, by construction Δ_K will be the product of two odd primes. If we take $\Delta_K = -pq$ with p and q such that $p \equiv -q \pmod 4$ then Δ_K is a fundamental discriminant. Moreover the 2-Sylow subgroup will be isomorphic to $\mathbf{Z}/2\mathbf{Z}$ if we choose p and q such that $(p/q) = (q/p) = -1$ (cf. [Kap73, p. 598]). In that case, we will work with the odd part, which is the group of squares of $C(\mathscr{O}_{\Delta_K})$.

Following the Cohen-Lenstra heuristics, cf. [Coh00, Chapter 5.10.1], the probability that the odd part of the class group is cyclic is 97.757% and the probability that an odd prime r divides $h(\mathscr{O}_{\Delta_K})$ is approximately $1/r + 1/r^2$. As a result, we can not guarantee that the order of the odd part is not divisible by small primes. Nevertheless, as indicated in [HM00a], this does not lead to a weakness on the discrete logarithm problem, as there is no efficient algorithm to compute $h(\mathscr{O}_{\Delta_K})$ or odd multiples or factors of $h(\mathscr{O}_{\Delta_K})$, hence an adaptation of the Pohlig-Hellman Algorithm is not possible.

On average, $h(\mathcal{O}_{\Delta_K})$ is in the order of $\sqrt{|\Delta_K|}$, see [Coh00, Theorem 4.9.15 (Brauer-Siegel)]. Moreover (cf. [Coh00, p. 295]),

$$h(\Delta_K) < \frac{1}{\pi}\log|\Delta_K|\sqrt{|\Delta_K|}. \tag{J.1}$$

To conclude, following [HM00a], if Δ_K is taken large enough, generic methods to compute discrete logarithm such as Pollard λ−method are slower than the index-calculus algorithms. Thus, since index-calculus algorithms for solving the discrete logarithm problem are asymptotically much slower than index-calculus algorithms to solve the integer factorization problem, the discriminant can be taken smaller than RSA modulus. In [BJS10], the discrete logarithm problem with a discriminant of 1348 bits (resp. 1828 bits) is estimated as hard as factoring a 2048 bits (resp. 3072 bits) RSA integer.

C.B.4. Elgamal Cryptosystem Adaptations in Class Group

In [BW88], Buchmann and Williams proposed an adaptation of the Diffie-Hellman key exchange in imaginary quadratic fields and briefly described an adaptation of the Elgamal cryptosystem in the same setting. Efficient implementations of these cryptosystems are discussed in [BDW90, SP05, BH01] and [BV07]. At a high level, the key generation process of these adaptations of Elgamal can be sketched as follows:

- Generate Δ_K a fundamental negative discriminant, such that $|\Delta_K|$ is large enough to thwart the computation of discrete logarithm (cf. previous subsection) ;
- choose g a class of $C(\mathcal{O}_{\Delta_K})$ of even order (from the discussion of the previous subsection, the order of g will be close to $h(\Delta_K) \approx \sqrt{|\Delta_K|}$ with high probability) ;
- the private key is $x \xleftarrow{\$} \{0, \dots, \lfloor\sqrt{|\Delta_K|}\rfloor\}$ and the public key is (g, h), where $h = g^x$.

To implement Elgamal, it remains the problem of the embedding of a message. In [BW88], an integer m is encrypted as $(g^r, m + \mathrm{N}(h^r))$ where $\mathrm{N}(h^r)$ denotes the norm of the reduced ideal of the class h^r. As a result, the scheme is not based on the traditional DDH assumption.

Another solution is given in [SP05, Section 2]. An integer message $m \leq \sqrt{|\Delta|}/2$ is mapped to the class M of an ideal above p where p is the first prime with $p > m$ such that Δ is a quadratic residue modulo p. If $d = m - p$, the message m is encrypted as $(g^r, \mathrm{M}h^r, d)$: The distance d seems to be public, in order to recover m from M. This can be a problem for semantic security: the first stage adversary can choose two messages m_0, m_1 such that $d_0 \neq d_1$ and easily win the indistinguishability game with probability one by recognizing the message thanks to the distance.

In [BH01], a "hashed" version is used, a bit-string m is encrypted as $(g^r, m \oplus \mathrm{H}(h^r))$ where H is a cryptographic hash function. In [BV07], an adaptation of DHIES is described.

An variant of the Elgamal cryptosystem in a non maximal order of discriminant $\Delta_q = q^2\Delta_K$ is presented in [HJPT98]. A traditional setup of Elgamal is done in $C(\mathcal{O}_{\Delta_q})$,

$h = g^x$. A ciphertext is (g^r, mh^r) in $C(\mathcal{O}_{\Delta_q})$ where m is an ideal of norm smaller than $\sqrt{\Delta_K}/2$. To decrypt, the ciphertext is moved in the maximal order with the trapdoor q where a traditional decryption is made to recover the message in $C(\mathcal{O}_{\Delta_K})$. Eventually, the message is lifted back in $C(\mathcal{O}_{\Delta_q})$. This variant can be seen as an Elgamal with a CRT decryption procedure: its advantage is that most of the decryption computation is done in $C(\mathcal{O}_{\Delta_K})$ and Δ_K can be chosen relatively small (big enough such the factorization of Δ_q is intractable, the discrete logarithm problem can be easy in $C(\mathcal{O}_{\Delta_K})$). The problem of the embedding of the plaintext in an ideal is not addressed in this paper. A chosen-ciphertext attack against this cryptosystem has been proposed in [JJ00].

In [KM03], an adaptation of the Diffie-Hellman key exchange and of the Elgamal cryptosystem are given using class semigroup of an imaginary non-maximal quadratic order. Unfortunately a cryptanalysis of this proposal has been presented in [Jac04].

A final important remark on the adaptation of the Elgamal cryptosystem is that it is necessary to work in the group of squares, *i. e.*, the *principal genus*. We didn't find this remark in previous works: in the whole class group, the DDH problem is easy. Indeed, it is well known that in $(\mathbf{Z}/p\mathbf{Z})^\times$, one can compute Legendre symbols and defeats the DDH assumption. As a consequence, it is necessary to work in the group of squares. In a class group, for example if the discriminant $\Delta = -\prod_{i=1}^{k} p_i$ is odd and the p_i are distinct primes numbers, we can associate to a class the value of the generic characters, the Legendre symbols (r, p_i) for i from 1 to k where r is an integer represented by the class (see [Cox99] for details on genus theory). It is easy to see that the previous attack in $(\mathbf{Z}/p\mathbf{Z})^\times$ can be adapted in class groups with the computation of the generic characters. As a result, it is necessary to work in the group of squares, which is the principal genus (cf. [Cox99, Theorem 3.15]), *i. e.*, the set of classes such that the generic characters all equal 1.

C.C. Uniform Sampling in a Cyclic Group

Let us first recall some well known facts on the statistical distance of two discrete random variables.

Let X and Y two discrete random variables with values in Ω. The statistical distance between X and Y is defined as $\Delta(X, Y) = \sup_{A\subseteq\Omega} |\Pr[X \in A] - \Pr[Y \in A]|$.

Note that $\Pr[X \in \bar{A}] - \Pr[Y \in \bar{A}] = 1 - \Pr[X \in A] - 1 + \Pr[Y \in A] = \Pr[Y \in A] - \Pr[X \in A]$. So we can restrict to subsets A such that the difference is positive. Moreover, in order to maximize the difference, we can let $A = \{\omega \in \Omega, \Pr[X = \omega] > \Pr[Y = \omega]\}$. So

$$\Delta(X, Y) = \sum_{\omega\in A} \Pr[X = \omega] - \Pr[Y = \omega].$$

Now as

$$\Delta(X, Y) = \Delta(Y, X) = \Pr[Y \in \bar{A}] - \Pr[X \in \bar{A}] = \sum_{\omega\in\bar{A}} \Pr[Y = \omega] - \Pr[X = \omega],$$

we have

$$2\Delta(X,Y) = \sum_{\omega\in\Omega} |\Pr[X=\omega] - \Pr[Y=\omega]|, \text{ so } \Delta(X,Y) = \frac{1}{2}\sum_{\omega\in\Omega} |\Pr[X=\omega] - \Pr[Y=\omega]|.$$

Lemma C – 4. *Let* G *be a cyclic group of order* n*, generated by* g*. Consider the random variable* X *with values in* G *with uniform distribution:* $\Pr[X = h] = \frac{1}{n}$ *for all* h *in* G*, and* Y *the random variable with values in* G *defined as follows. Draw* y *in* $\{0,\dots,B-1\}$ *from the uniform distribution with* $B \geqslant n$*, and* $Y = g^y$*. Let* $r = B \mod n$*. Then,* $\Delta(X,Y) = \frac{r(n-r)}{nB} \leqslant \frac{n}{4B}$.

Proof. Let X′ the random variable with values in $\{0,\dots,n-1\}$ with uniform distribution and Y′ defined by $Y' = (y \mod n)$ where y is drawn in $\{0,\dots,B-1\}$ with uniform distribution. Clearly, $\Delta(X,Y) = \Delta(X',Y')$. Let $B = qn + r$ with $0 \leqslant r < n$ be the euclidean division of B by n. For $c \in \{0,\dots,r-1\}$, $\Pr[Y' = c] = (q+1)/B > \frac{1}{n}$ as $(q+1)n > B$. For $c \in \{r,\dots,n-1\}$, $\Pr[Y' = c] = q/B \leqslant 1/n$. So for $A = \{0,\dots,r-1\}$, we have

$$\Delta(X,Y) = \Delta(X',Y') = \sum_{c\in A} \Pr[Y'=c] - \Pr[X'=c] = r\left(\frac{q+1}{B} - \frac{1}{n}\right).$$

Using the fact that $q = \frac{B-r}{n}$, this simplifies to $\Delta(X,Y) = \frac{r(n-r)}{nB}$. Moreover as $r(n-r) \leqslant n^2/4$,

$$\Delta(X,Y) \leqslant \frac{n}{4B}.$$

□

Chapter D

Encryption Switching Protocols Revisited: Switching modulo p

Joint work with Laurent Imbert and Fabien Laguillaumie
[CIL17]

Abstract. At CRYPTO 2016, Couteau, Peters and Pointcheval introduced a new primitive called *encryption switching protocols*, allowing to switch ciphertexts between two encryption schemes. If such an ESP is built with two schemes that are respectively additively and multiplicatively homomorphic, it naturally gives rise to a secure 2-party computation protocol. It is thus perfectly suited for evaluating functions, such as multivariate polynomials, given as arithmetic circuits. Couteau et al. built an ESP to switch between Elgamal and Paillier encryptions which do not naturally fit well together. Consequently, they had to design a clever variant of Elgamal over $\mathbf{Z}/n\mathbf{Z}$ with a costly shared decryption.

In this paper, we first present a conceptually simple generic construction for encryption switching protocols. We then give an efficient instantiation of our generic approach that uses two well-suited protocols, namely a variant of Elgamal in $\mathbf{Z}/p\mathbf{Z}$ and the Castagnos-Laguillaumie encryption which is additively homomorphic over $\mathbf{Z}/p\mathbf{Z}$. Among other advantages, this allows to perform all computations modulo a prime p instead of an RSA modulus. Overall, our solution leads to significant reductions in the number of rounds as well as the number of bits exchanged by the parties during the interactive protocols. We also show how to extend its security to the malicious setting.

D.1. Introduction

Through interactive cryptographic protocols, secure multi-party computation (MPC) allows several parties to compute the image of a pre-agreed function of their private inputs. At the end of the interaction, anything that a party (or a sufficiently small coalition of parties) has learned from the protocol could have been deduced from its public and secret inputs and outputs. In other words, the adversary's view can be efficiently forged by a simulator that has only access to the data publicly known by the adversary. This important area of research emerged in the 80's with the works of Yao [Yao82] and Goldreich, Micali and Wigderson [GMW87]. Formal security notions can be found in [MR92, Bea92, Can00]. Initially considered as a theoretical subject due to overly inefficient protocols, MPC has nowadays reached a reasonable complexity and has became relevant for practical purposes [BDJ+06] especially in the 2-party case [PSSW09, MNPS04, LPS08]. Several techniques may be used to design secure multi-party computation. Some recently proposed solutions use or combine tools from oblivious transfer [ALSZ15,LP11], secret sharing with pre-processing [NNOB12,DZ13], garbled circuits [LPS08], homomorphic encryption [CDN01,DN03], and somewhat or fully homomorphic encryption [BDOZ11, AJL+12].

In [CPP16], Couteau, Peters and Pointcheval formalized an innovative technique to securely compute functions between two players, thanks to interactive cryptographic protocols called *encryption switching protocols* (ESP). This mechanism permits secure 2-party computations against semi-honest adversaries (honest-but-curious) as well as malicious adversaries, i.e. opponents which might not follow the specifications of the protocol. Couteau et al.'s proposal relies on a pair of encryption schemes $(\Pi_+, \Pi_\times)$ which are respectively additively and multiplicatively homomorphic and which share a common message space. Furthermore, there exists switching algorithms to securely convert ciphertexts between Π_+ and $\Pi_\times$. More precisely, there exists a protocol $\mathsf{Switch}_{+\to\times}$ which takes as input an encryption C_m^+ of a message m under Π_+, and returns a ciphertext $\mathrm{C}_m^\times$ of the same message m under $\Pi_\times$. Symmetrically, there exists a second protocol $\mathsf{Switch}_{\times\to+}$ which computes a ciphertext for m under Π_+ when given a ciphertext for m under $\Pi_\times$. The advantage of this construction is that it benefits from the intrinsic efficiency of multiplicatively homomorphic encryption like Elgamal [ElG85] or additively homomorphic encryption like Paillier [Pai99]. In [CPP16], Couteau et al. present a natural construction for secure 2-party computation from any ESP.

Applications.

Two-party computation is the most important application of an ESP. In [CDN01], an MPC protocol is built from only an additively homomorphic encryption scheme which is a natural alternative to an ESP. The round complexity of their protocol is in $\mathcal{O}(d)$, where d is the depth of the circuit $\mathscr{C}$ to be evaluated, and if we suppose that the multiplicative gates can be evaluated in parallel at each level. With an ESP, gathering the additive and multiplicative gates separately would imply a dramatic improvement. Fortunately, the result by Valiant, Skyum, Berkowitz and Rackoff from [VSBR83, Theorem

3], states that for any circuit $\mathscr{C}$ of size s and degree d computing a polynomial f, there is another circuit $\mathscr{C}'$ of size $\mathscr{O}(s^3)$ and depth $\mathscr{O}(\log(s)\log(d))$ which computes the same polynomial f. Moreover, Allender, Jiao, Mahajan and Vinay showed that the circuit $\mathscr{C}'$ is by construction layered (see [AJMV98]), in the sense that it is composed of layers whose gates are all the same and alternatively + and ×. Roughly speaking, $\mathscr{C}'$ is of the form $(\sum\prod)^{\mathscr{O}(\log(s)\log(d))}$ where $\sum$ has only additive gates and $\prod$ has only multiplicative gates. In other words, the polynomial f can be written as a composition of $\mathscr{O}(\log(s)\log(d))$ polynomials written in a sparse representation. The ESP allows to treat each $\sum$ and $\prod$ independently, so that the number of switches and therefore the number of rounds is essentially $\mathscr{O}(\log(s)\log(d))$, instead of $\mathscr{O}(d)$ for [CDN01]. Any enhancement of an ESP will naturally improve any protocol which requires to evaluate on encrypted data a polynomial given in the form of a sum of monomials. Especially it is well-suited to oblivious evaluation of multivariate polynomials [NP06, KY01, TJB13] or private disjointness testing [YWPZ08].

Related works.

The idea of switching between ciphertexts for different homomorphic schemes was first proposed by Gavin and Minier in [GM09] in the context of oblivious evaluation of multivariate polynomials. They proposed to combine a variant of Elgamal over $(\mathbf{Z}/N\mathbf{Z})^*$ (where N is an RSA modulus) with a Goldwasser-Micali encryption protocol [GM84]. Unfortunately, as noticed by Couteau et al. [CPP16], their design contains a serious flaw which renders their scheme insecure (the public key contains a square root of unity with Jacobi symbol -1, which exposes the factorization of N).

Another attempt was proposed in [TSCS13] with a compiler designed to embed homomorphic computation into C programs to operate on encrypted data. The security of this construction relies on a very strong assumption since switching between the encryption schemes is done using a secure device which decrypts and re-encrypts using the secret key.

In [LTSC14], Lim, Tople, Saxena and Chang proposed a primitive called *switchable homomorphic encryption* implemented using Paillier and Elgamal, in the context of computation on encrypted data. Again, this proposal uses an insecure version of Elgamal, which does not satisfy the indistinguishability under a chosen plaintext attack. It is indeed very difficult to design two compatible encryption schemes from unrelated protocols like Paillier and Elgamal.

Couteau et al. managed to tune Elgamal so that it can switch with Paillier, but their construction remains fairly expensive. In particular, they constructed a variant of Elgamal over $(\mathbf{Z}/n\mathbf{Z})^*$, where n is an RSA modulus, which is the same as the Paillier modulus. As Elgamal is secure only in the subgroup $\mathbf{J}_n$ of $(\mathbf{Z}/n\mathbf{Z})^*$ of elements of Jacobi symbol $+1$, they need a careful encoding of the group $(\mathbf{Z}/n\mathbf{Z})^*$. The security relies on the DDH assumption in $\mathbf{J}_n$ and the quadratic residuosity assumption in $(\mathbf{Z}/n\mathbf{Z})^*$. Because their Elgamal variant does not support a simple 2-party decryption (a Paillier layer has to be added to Elgamal in order to simulate a threshold decryption), the switching protocols are intricate and specific to their construction.

Our contributions and overview of our results.

In this paper, we first propose a generic ESP inspired by Couteau et al's solid basis. Our construction relies on the existence of an additively homomorphic encryption Π_+ and a multiplicatively homomorphic encryption $\Pi_\times$ which support a 1-round threshold decryption and achieve classical security properties (IND-CPA and zero-knowledge of the 2-party decryption). Because the message spaces must be compatible, we suppose that Π_+ works over a ring $\mathscr{R}$ and $\Pi_\times$ over a monoid $\mathscr{M}$ with $\mathscr{R} \cap \mathscr{M} = \mathscr{R}^*$ where $\mathscr{R}^*$ is the set of invertible elements of $\mathscr{R}$. A major issue when designing an ESP is to embed the zero message[1] into the message space for $\Pi_\times$, while preserving the homomorphic and security properties. In Section D.4.2, we propose a generic technique to do so, inspired by the approach employed in [CPP16]. Contrary to their construction, our switching protocols over $\mathscr{R}^*$ (i.e. without the zero-message) are symmetrical, i.e. both $\mathsf{Switch}_{+\to\times}$ and $\mathsf{Switch}_{\times\to+}$ follow the same elementary description given in Figure D.3. This is possible for two reasons: first because we suppose that both Π_+ and Π_+ admit a single round 2-party decryption, and second because they both possess a $\mathsf{ScalMul}$ algorithm which takes as input a ciphertext of m and a plaintext α and outputs a ciphertext of $\alpha \times m$ (which is why we consider a ring as the message space for Π_+ instead of an additive group).

Besides, they are very efficient: as detailed in Section D.5.3, they only require 2 rounds, whereas Couteau et al.'s $\mathsf{Switch}_{\times\to+}$ needs 6. Our full switching protocols work over $\mathscr{R}^* \cup \{0\}$. They are built on top of the switching protocols over $\mathscr{R}^*$ (i.e. without 0), plus some additional tools like 2-party re-encryption, encrypted zero test, and a 2-party protocol to homomorphically compute a product under Π_+ (see Figure D.1). Our security proofs are simpler than Couteau et al's. In terms of round complexity, the savings are substantial: our full ESP protocols require 7 and 4 rounds respectively, whereas Couteau et al's ESP need 7 and 11.

In a second part, we propose an efficient instantiation of our generic protocol over the field $\mathbf{Z}/p\mathbf{Z}$. Working over $\mathbf{Z}/p\mathbf{Z}$ has several advantages compared to $\mathbf{Z}/n\mathbf{Z}$ (for an RSA modulus n): it means true message space equality, instead of computational equality. It also means faster arithmetic by carefully choosing the prime p. Our instantiation combines a variant of Elgamal together with the Castagnos-Laguillaumie additively homomorphic encryption from [CL15]. Because Elgamal is only secure in the subgroup of squares modulo p, our variant over $\mathbf{Z}/p\mathbf{Z}^*$, denoted Eg^*, encodes the messages into squares and adds the encryption of a witness bit (i.e. the Legendre Symbol) under Goldwasser-Micali [GM84] for its homomorphic properties modulo 2. For Π_+, we use a variant of the Castagnos-Laguillaumie encryption scheme (CL) described in Section C.4 of [CL15]. We work over (subgroups of) the class group of an order of a quadratic field of discriminant $\Delta_p = -p^3$. Computations are done in this class group. The elements are represented by their unique reduced representative, i.e. by two integers of size $\sqrt{|\Delta_p|}$. Thus, an element of the class group requires $3\log p$ bits. Under slightly different security assumptions, it is possible to further reduce the size of the elements and

[1] The zero message has to be taken into account since it can arise easily by homomorphically subtracting two equivalent ciphertexts of the same message.

to achieve a better bit complexity. We discuss these implementation options in Section D.5.3 and compare their costs with the ESP from Couteau et al. [CPP16]. Our ESP protocol reduces the round complexity by a factor of almost 3 in the × → + direction, while remaining constant in the other direction. Using the variant of CL optimized for size, the bit complexity is also significantly reduced in the × → +, while remaining in the same order of magnitude in the other.

We also propose improvements on CL that can be on independent interest. That system makes exponentiations in a group whose order is unknown but where a bound is known. We show that using discrete Gaussian distribution instead of uniform distribution improves the overall computational efficiency of the scheme. Moreover in order to use our generic construction, we devise a 2-party decryption for CL.

Eventually we discuss in Section D.6 how to adapt our generic construction and our instantiation against malicious adversaries.

D.2. Cryptographic Building Blocks

In this section, we recall some classical definitions and operations that will be useful in the rest of the paper.

D.2.1. Homomorphic encryption schemes.

In Section D.3, we will give a definition of *Encryption Switching Protocols (ESP)*, previously proposed in [CPP16]. An ESP allows to switch a ciphertext under an encryption protocol Π_1 into a ciphertext under another encryption protocol Π_2, and vice versa. ESP require the protocols Π_1 and Π_2 to be (partially) homomorphic. In this paper, we consider ESP between an additively homomorphic encryption Π_+ and a multiplicatively homomorphic encryption $\Pi_\times$.

In Definitions D − 1 and D − 2 below, we define Π_+ and $\Pi_\times$ formally in a generic context. An additive homomorphic encryption is most commonly defined over a group. In our setting, Π_+ is defined over a ring $\mathscr{R}$ to guarantee that for $m, m' \in \mathscr{R}$, the product $m \times m'$ is well defined. For genericity $\Pi_\times$ is defined over an algebraic structure with a single associative binary operation (denoted ×) and an identity element; i.e. a monoid. By doing so, our definition encapsulates encryption schemes over $(\mathbf{Z}/pq\mathbf{Z})^* \cup \{0\}$ (with p, q primes) such as [CPP16], as well as our instantiation over $\mathbf{Z}/p\mathbf{Z}$ presented in Section D.5.

Definition D − 1 (Additively homomorphic encryption)**.** Let $(\mathscr{R}, +, \times, 1_{\mathscr{R}}, 0_{\mathscr{R}})$ be a ring. An *additively homomorphic encryption scheme* over the message space $\mathscr{R}$ is a tuple $\Pi_+ = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt}, \mathsf{Hom}_+, \mathsf{ScalMul})$ such that:

Setup is a PPT algorithm algorithm which takes as input a security parameter 1^λ and outputs public parameters params (these public parameters will be omitted in the algorithms' inputs).

KeyGen is a PPT algorithm taking public parameters as inputs and outputting a pair of public and secret key (pk, sk).

$\mathsf{Encrypt}$ is a PPT algorithm which takes as input some public parameters, a public key pk and a message $m \in \mathscr{R}$, and outputs an encryption c.

$\mathsf{Decrypt}$ is a PPT algorithm which takes as input public parameters, a public key pk (omitted in Decrypt's input), a secret key sk and a ciphertext c, and outputs a message $m \in \mathscr{R}$.

Hom_+ is a PPT algorithm which takes as inputs some public parameters, a public key pk and two ciphertexts c and c' of $m \in \mathscr{R}$ and $m' \in \mathscr{R}$ respectively, and outputs a ciphertext c'' such that $\Pi_+.\mathsf{Decrypt}(sk, c'') = m + m' \in \mathscr{R}$.

$\mathsf{ScalMul}$ is a PPT algorithm which takes as inputs some public parameters, a public key pk, a ciphertext c of $m \in \mathscr{R}$ and a plaintext $\alpha \in \mathscr{R}$, and outputs a ciphertext c' such that $\Pi_+.\mathsf{Decrypt}(sk, c') = \alpha \times m \in \mathscr{R}$.

Remark D – 1. A generic algorithm for computing $\Pi_+.\mathsf{ScalMul}(pk, c, \alpha)$ is given by $\mathsf{2Mul}_+(c, \Pi_+.\mathsf{Encrypt}(\alpha))$, where $\mathsf{2Mul}_+$ is an interactive PPT algorithm which computes homomorphically the product of two ciphertexts for Π_+. $\mathsf{2Mul}_+$ is defined more formally in Section D.2.3. For our instantiation we provide a non-interactive, more efficient version over $\mathbf{Z}/p\mathbf{Z}$ (see Section D.5).

Definition D – 2 (Multiplicatively homomorphic encryption)**.** Let $(\mathscr{M}, \times, 1_{\mathscr{M}})$ be a monoid. A ***multiplicatively homomorphic encryption scheme*** over the message space $\mathscr{M}$ is $\Pi_\times = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt}, \mathsf{Hom}_\times, \mathsf{ScalMul})$ such that:

Setup, KeyGen, $\mathsf{Encrypt}$ and $\mathsf{Decrypt}$ as in Definition D – 1 except that Encrypt and Decrypt receives the input messages from $\mathscr{M}$ instead of $\mathscr{R}$.

$\mathsf{Hom}_\times$ is a PPT algorithm which takes as input some public parameters, a public key pk and two ciphertexts c and c' of $m \in \mathscr{M}$ and $m' \in \mathscr{M}$ respectively, and outputs a ciphertext c'' such that $\Pi_\times.\mathsf{Decrypt}(sk, c'') = m \times m' \in \mathscr{M}$.

$\mathsf{ScalMul}$ is a PPT algorithm which takes as inputs some public parameters, a public key pk, a ciphertext c of $m \in \mathscr{M}$ and a plaintext $\alpha \in \mathscr{M}$, and outputs a ciphertext c' such that $\Pi_\times.\mathsf{Decrypt}(sk, c') = \alpha \times m \in \mathscr{M}$.

Remark D – 2. A generic algorithm for computing $\Pi_\times.\mathsf{ScalMul}(pk, c, \alpha)$ is given by $\Pi_\times.\mathsf{Hom}_\times(pk, c, c')$, where $c' = \Pi_\times.\mathsf{Encrypt}(pk, \alpha)$. In Section D.5, we provide a more efficient version over $(\mathbf{Z}/p\mathbf{Z})^*$.

The above encryption schemes must be correct in the usual sense. Moreover, we consider as a security requirement the indistinguishability under a chosen plaintext attack (IND-CPA). We refer the reader to e.g, [KL14] for the standard definition of IND-CPA.

D.2.2. One round 2-Party Decryption.

A crucial feature of the encryption protocols which are used in the ESP is the fact that they support a 2-party decryption (threshold cryptosystems were introduced in [DF90]). These encryption schemes are equipped with a Share procedure that is run by a trusted dealer, which works as follows: it takes as input a pair of keys (sk, pk) obtained from the KeyGen algorithm and produces two shares sk_{A} and sk_{B} of the secret key sk. It outputs $(sk_{\mathrm{A}}, sk_{\mathrm{B}})$ and an updated public part still denoted pk. Its decryption procedure is an interactive protocol denoted 2Dec which takes as inputs the public parameters, a ciphertext c, and the secret key of each participant sk_{A} and sk_{B} and outputs a plaintext m which would have been obtained as $\mathsf{Decrypt}(sk, c)$.

Contrary to the classical definition of threshold decryption, we suppose that the protocol is in a *single* round. The protocol $\mathsf{2Dec}(pk, c; sk_{\mathrm{A}}; sk_{\mathrm{B}})$ is supposed as follows: Alice starts the protocol and sends her information in one flow to Bob which ends the computation and gets the plaintext. This is because in our context, we do not decrypt plaintexts but plaintexts which are masked by a random element. For example, protocols whose decryption only performs exponentiations with secret exponents gives one round 2-party decryption by sharing the exponentiations. This is the case for many cryptosystems.

The semantic security is adapted from the standard IND-CPA notion by giving the adversary one of the two secret keys, as well as a share decryption oracle which simulate the party with the other secret key. A formal definition can be found for instance in [SG02, CDN01].

We need as an additional security requirement the notion of zero-knowledge defined in Appendix D.A, which means that no information on the secret keys are leaked during an interaction with a curious adversary. Cryptosystems like Elgamal [DF90] or Paillier [FPS01] satisfy all these properties. We will propose a variant of Elgamal and a variant of Castagnos-Laguillaumie [CL15] that satisfy also these properties in Section D.5.

D.2.3. Homomorphically computing a product with Π_+.

A core routine of our protocol is the computation of a Π_+-encryption of a product XY given Π_+-encryptions of X and Y (this is why we assume that Π_+ has a ring $\mathscr{R}$ as message space). We describe in Fig. D.1 a protocol which is implicitly used in [CPP16]. It is a simplified variant of a protocol proposed by Cramer, Damgård and Nielsen from [CDN01]: the main difference comes from the fact that the result of this 2-party computation is obtained only by one of the user, who can forward the result to the other. This leads to the use of a single randomness on Alice's side, instead of one on each side. We will denote by $\mathsf{2Mul}_+(pk, \mathrm{C}_{\mathrm{X}}^+, \mathrm{C}_{\mathrm{Y}}^+; sk_{\mathrm{A}}; sk_{\mathrm{B}})$ a call to this protocol. Again, this protocol will be a 2-round protocol since the shared decryption is single round, and the first ciphertext can be sent along with the shared decryption. This protocol has to be zero-knowledge in the sense similar to those of Def. D − 5 and D − 7 (we do not write down this definition which can be readily adapted).

Theorem D − 1. *Let Π_+ be an additively homomorphic encryption scheme with a zero-knowledge one round 2-party decryption. Then, the protocol described in Fig. D.1 is correct and zero-knowledge.*

Proof. The correctness follows from the correctness of the encryption scheme and its homomorphic properties. Let us prove first that it is zero-knowledge for Alice. We

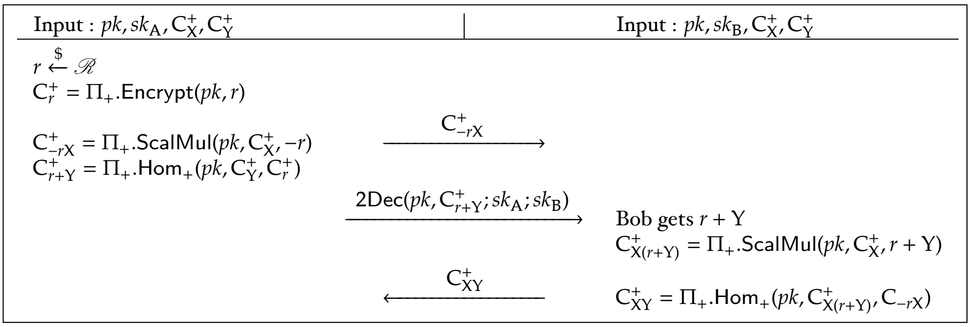

Input : pk, sk_A, C_X^+, C_Y^+
Input : pk, sk_B, C_X^+, C_Y^+
$r \xleftarrow{\$} \mathscr{R}$
$C_r^+ = \Pi_+.\mathsf{Encrypt}(pk, r)$
$C_{-rX}^+ = \Pi_+.\mathsf{ScalMul}(pk, C_X^+, -r)$
$C_{r+Y}^+ = \Pi_+.\mathsf{Hom}_+(pk, C_Y^+, C_r^+)$
C_{-rX}^+
$\mathsf{2Dec}(pk, C_{r+Y}^+; sk_A; sk_B)$
Bob gets $r + Y$
$C_{X(r+Y)}^+ = \Pi_+.\mathsf{ScalMul}(pk, C_X^+, r + Y)$
C_{XY}^+
$C_{XY}^+ = \Pi_+.\mathsf{Hom}_+(pk, C_{X(r+Y)}^+, C_{-rX})$

Figure D.1: $\mathsf{2Mul}_+$: 2-party protocol to compute C_{XY}^+ from C_X^+ and C_Y^+

describe a simulator Sim whose behavior is indistinguishable from Alice's behavior in front of an adversarial Bob. The simulator receives as input the public key pk_+ and will set $\mathsf{Sim}_{\mathsf{Share}}$ as follows: it calls out $\mathsf{Sim}_{\mathsf{Share}}^{2d}$ procedure of the zero-knowledge property of 2-party decryption for Π_+ with pk_+ as input. It obtains a simulated share $sk_B = x_B^+$ and feeds the adversary with it. When Sim is requested for the 2-party computation of C_{XY}^+ from C_X^+ and C_Y^+, it receives a pair of $((C_X^+, C_Y^+), \bar{C})$ where $\bar{C}$ is a ciphertext of XY, it does the following to simulate C_{-rX}^+, C_{r+Y}^+ and C_A:

- It picks R at random in the plaintext space and sets $C_{r+Y}^+ = \mathsf{Encrypt}(pk, R)$.
- Then it uses the simulator for the zero-knowledge for Alice of the 2-party decryption $\mathsf{Share}_A^{2d}(C_{r+Y}^+, R, x_B^+)$ so that Bob decrypts R (which is equivalent to the decryption $\mathsf{Decrypt}(sk, C_{r+Y}^+)$).
- Eventually, it sets $C_{-rX} = \mathsf{Hom}_+(\bar{C}, \mathsf{ScalMul}(C_X^+, -R))$. This ciphertext encrypts $XY - RX$ so that Bob's final Hom_+ evaluation will cancel out the RX part and lead to $\bar{C}$.

The simulated view is the same as a genuine one with $R = r + Y$, which means that they are indistinguishable, and the protocol is zero-knowledge for Alice. The protocol is obviously zero-knowledge for Bob: Bob's contribution is simulated by just sending $\bar{C}$. □

D.2.4. 2-Party re-Encryption.

The final tool we need to build our encryption switching protocol is an interactive 2-party protocol to re-encrypt a ciphertext from an encryption scheme Π_+ intended to pk into a ciphertext of the same encryption scheme of the same message, but intended to another key pk'. This protocol is depicted in Fig. D.2. Note that the initial ciphertext to be transformed is not known to Bob. This protocol readily extends to the multiplicative case, which is useless for our purpose. With a proof similar to the proof of Theorem D – 1, we showed that

Theorem D – 2. *Let Π_+ be an additively homomorphic encryption scheme with a zero-knowledge one round 2-party decryption then the protocol described in Fig. D.2 is correct and zero-knowledge.*

Input : pk, pk', sk_A, C_m		Input : pk, pk', sk_B
$r \overset{\$}{\leftarrow} \mathscr{R}$		
$C'_{-r} = \Pi_+.\mathsf{Encrypt}(pk', -r)$ $C_r = \Pi_+.\mathsf{Encrypt}(pk, r)$ $C_{m+r} = \Pi_+.\mathsf{Hom}_+(pk, C_m, C_r)$	$\xrightarrow{C'_{-r}}$	
	$\xrightarrow{\mathsf{2Dec}(pk, C^+_{m+r}; sk_A; sk_B)}$	Bob gets $m + r$ $C'_{m+r} = \Pi_+.\mathsf{Encrypt}(pk', m + r)$
	$\xleftarrow{C'_m}$	$C'_m = \Pi_+.\mathsf{Hom}_+(pk', C'_{m+r}, C'_{-r})$

Figure D.2: 2-party ReEnc_+

D.3. Encryption Switching Protocols

The global scenario is established as follows: two semantically secure threshold homomorphic encryption schemes, one additive, and the other multiplicative, are at the disposal of two players. A public key is provided for each protocols, and the matching secret key is shared among the players by a trusted dealer. Ideally, these two encryption schemes should have the same plaintext space, which is assumed to be a ring or a field. An encryption switching protocols makes it possible to interactively transform a ciphertext from a source encryption scheme into a ciphertext for the other encryption scheme (the target one) and *vice versa*. The formal definitions are given in the following paragraphs.

Definition D – 3 (Twin ciphertexts)**.** Let Π_+ and $\Pi_\times$ be two different encryption schemes with plaintext and ciphertext spaces respectively $\mathscr{M}_+, \mathscr{C}_+$ and $\mathscr{M}_\times, \mathscr{C}_\times$. If $C^+_m \in \mathscr{C}_+$ and $C^\times_m \in \mathscr{C}_\times$ are two encryptions of the same message $m \in \mathscr{M}_+ \cap \mathscr{M}_\times$, they are said to be *twin ciphertexts*.

We will say that two ciphertexts from the same encryption scheme which decrypt to the same plaintext are *equivalent*.

Definition D – 4 (Encryption Switching Protocols)**.** An encryption switching protocol (ESP) between Π_+ and $\Pi_\times$, denoted $\Pi_+ \rightleftharpoons \Pi_\times$, is a protocol involving three parties:

a trusted dealer $\mathscr{D}$ and two users A and B. It uses common Setup and KeyGen algorithms to set the message space between Π_+ and $\Pi_\times$ and keys. It is a pair of interactive protocols (Share, Switch) defined as follows:

- $\mathsf{Share}((pk_+, sk_+), (pk_\times, sk_\times)) \to (pk, sk_\mathrm{A}, sk_\mathrm{B})$: It is a protocol (run by $\mathscr{D}$) which takes as input two pairs of keys (pk_+, sk_+) and $(pk_\times, sk_\times)$ produced by the algorithms $\Pi_+.\mathsf{KeyGen}$, $\Pi_\times.\mathsf{KeyGen}$ and Setup. It outputs the shares sk_A (sent to A) and sk_B (sent to B) of $(sk_+, sk_\times)$ and updates the public key pk.

- $\mathsf{Switch}_{\mathsf{way}}((pk, sk_\mathrm{A}, c), (pk, sk_\mathrm{B}, \mathrm{C})) \to \mathrm{C}'$ or $\perp$: It is an interactive protocol in the direction way $\in \{+ \to \times, \times \to +\}$ which takes as common input the public key and a ciphertext C under the source encryption scheme and as secret input the secret shares sk_A and sk_B. The output is a twin ciphertext C' of C under the target encryption scheme or $\perp$ if the execution encountered problems.

Correctness

An encryption switching protocols $\Pi_+ \rightleftharpoons \Pi_\times$ is *correct* if for any $\lambda \in \mathbf{N}$, $(\mathsf{params}_+, \mathsf{params}_\times) \leftarrow \mathsf{Setup}(1^\lambda)$, for any pair of keys $(pk_+, sk_+) \leftarrow \Pi_+.\mathsf{KeyGen}(1^\lambda, \mathsf{params}_+)$ and $(pk_\times, sk_\times) \leftarrow \Pi_\times.\mathsf{KeyGen}(1^\lambda, \mathsf{params}_\times)$, for any shares $(pk, sk_\mathrm{A}, sk_\mathrm{B}) \leftarrow \mathsf{Share}((pk_+, sk_+), (pk_\times, sk_\times))$, for any twin ciphertext pair $(\mathrm{C}_m^+, \mathrm{C}_m^\times)$ of a message $m \in \mathscr{M}_+ \cap \mathscr{M}_\times$,

$$\begin{aligned} \Pi_+.\mathsf{Decrypt}(sk_+, \mathsf{Switch}_{\times\to+}((pk, sk_\mathrm{A}, \mathrm{C}_m^\times), (pk, sk_\mathrm{B}, \mathrm{C}_m^\times))) &= m \\ \Pi_\times.\mathsf{Decrypt}(sk_\times, \mathsf{Switch}_{+\to\times}((pk, sk_\mathrm{A}, \mathrm{C}_m^+), (pk, sk_\mathrm{B}, \mathrm{C}_m^+))) &= m. \end{aligned}$$

Zero-knowledge

An ESP has to satisfy a notion of zero-knowledge similar to the notion for threshold decryption (see def. D−7). This property means that an adversary will not learn any other information on the secret share of a participant that he can learn from his own share, the input, and the output of the protocol.

Definition D − 5. An encryption switching protocols $\Pi_+ \rightleftharpoons \Pi_\times$ is *zero-knowledge for* A if there exists an efficient simulator $\mathsf{Sim} = (\mathsf{Sim}_{\mathsf{Share}}, \mathsf{Sim}_\mathrm{A})$ which simulates the sharing phase and the player A.

The subroutine $\mathsf{Sim}_{\mathsf{Share}}$ takes as input a public key $(pk_+, pk_\times)$ and outputs (pk', sk'_B) that simulates the public key obtained from the Share algorithm and Bob's share of the secret key.

The subroutine Sim_A takes as input a direction way $\in \{+ \to \times, \times \to +\}$, a source ciphertext C, a twin ciphertext $\bar{\mathrm{C}}$ and a flow flow. It emulates the output of an honest player A would answer upon receiving the flow flow when running the protocol $\mathsf{Switch}_{\mathsf{way}}((pk, sk_\mathrm{A}, \mathrm{C}), (pk, sk_\mathrm{B}, \mathrm{C}))$ without sk_A but possibly sk_B, and forcing the output to be a ciphertext C' which is equivalent to $\bar{\mathrm{C}}$.

Then, for all $\lambda \in \mathbf{N}$, for any parameters $(\mathsf{params}_+, \mathsf{params}_\times) \leftarrow \mathsf{Setup}(1^\lambda)$, for any pairs of keys $(pk_+, sk_+) \leftarrow \Pi_+.\mathsf{KeyGen}(1^\lambda, \mathsf{params}_+)$ and $(pk_\times, sk_\times) \leftarrow \Pi_\times.\mathsf{KeyGen}(1^\lambda,$

$\mathsf{params}_\times$), $(pk, sk_{\mathrm{A}}, sk_{\mathrm{B}}) \leftarrow \mathsf{Share}((pk_+, sk_+), (pk_\times, sk_\times))$ or for any *simulated* share $(pk', sk'_{\mathrm{B}}) \leftarrow \mathsf{Sim}_{\mathsf{Share}}(pk)$, and for any adversary $\mathscr{D}$ playing the role of B, the advantage

$$\mathsf{Adv}^{zk}_{\mathrm{A},\Pi_+ \rightleftharpoons \Pi_\times}(\mathscr{D}) = \left|\Pr[1 \leftarrow \mathscr{D}^{\mathrm{A}}(pk, sk_{\mathrm{B}})] - \Pr[1 \leftarrow \mathscr{D}^{\mathsf{Sim}_{\mathrm{A}}()}(pk', sk'_{\mathrm{B}})]\right|$$

is negligible.

We define similarly an encryption switching protocols $\Pi_+ \rightleftharpoons \Pi_\times$ that is *zero-knowledge for* B. It is *zero-knowledge* if it is zero-knowledge for A and B.

D.4. Generic Construction of an ESP on a Ring

We describe in this section a generic construction of an encryption switching protocol in the semi-honest model. Even though an ESP could allow to switch between any encryption schemes, its main interest is when its implemented with homomorphic encryptions. Therefore, we start with an additively homomorphic encryption Π_+ and a multiplicatively homomorphic encryption $\Pi_\times$ whose message space is respectively a ring $\mathscr{R}$ and a monoid $\mathscr{M}$. To fit most of the applications, we will make the assumption that $\mathscr{M} = \mathscr{R}^*$, the subgroup of invertible elements of $\mathscr{R}$, since in general the multiplicative homomorphic encryption will have a group as message space. In particular, this means that the intersection over which the switches are defined is $\mathscr{R} \cap \mathscr{M} = \mathscr{R}^*$.

As in [CPP16], in the first place, we are going to describe how we can switch between Π_+-encryptions and $\Pi_\times$-encryptions over $\mathscr{R}^*$. Then we will show how to modify $\Pi_\times$ in order to extend its message space to $\mathscr{R}^* \cup \{0\}$.

Definition D – 6 (Compatible encryption protocols)**.** Let $(\mathscr{R}, +, \times)$ be a ring. Let Π_+ and $\Pi_\times$ be an additively and multiplicatively homomorphic encryption in the sense of Def. D – 1 and D – 2. They are said to be *compatible* if

- Π_+ and $\Pi_\times$ have respectively $\mathscr{R}$ and $\mathscr{R}^*$ as message space,
- both of them admit a one-round 2-party decryption as defined in Section D.2.2,
- there exists a common setup algorithm Setup and common KeyGen which allows to set common parameters.

Remark D – 3. To illustrate this, our instantiation (resp. Couteau et al.'s instantiation) switches between an additively homomorphic encryption whose message space is the field $(\mathbf{Z}/p\mathbf{Z}, +, \times)$ (resp. the ring $(\mathbf{Z}/N\mathbf{Z}, +, \times)$) and a multiplicative homomorphic encryption whose message space is the group $((\mathbf{Z}/p\mathbf{Z})^*, \times)$ (resp. $((\mathbf{Z}/N\mathbf{Z})^*, \times)$) and the former is modified so that its message space is the monoid $(\mathbf{Z}/p\mathbf{Z}, \times)$ (resp. $((\mathbf{Z}/N\mathbf{Z})^* \cup \{0\}, \times)$). In particular, Couteau et al.'s make the additional algorithmic assumption that $(\mathbf{Z}/N\mathbf{Z})^*$ is computationally equal to $\mathbf{Z}/N\mathbf{Z}$.

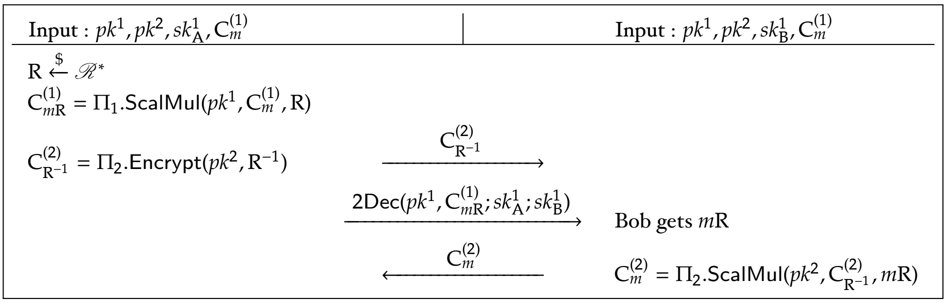
Share Protocol of the ESP The keys of Π_+ and $\Pi_\times$ are first shared by a trusted dealer, this corresponds to the Share algorithm from Def. D – 4. From public parameters generated using the common Setup algorithm and two pairs of keys (pk_+, sk_+) and $(pk_\times, sk_\times)$ it outputs the secret share $sk_{\mathrm{A}} = (sk^+_{\mathrm{A}}, sk^\times_{\mathrm{A}})$ for Alice and $sk_{\mathrm{B}} = (sk^+_{\mathrm{B}}, sk^\times_{\mathrm{B}})$ for Bob using the Share procedures of the 2-party decryption of Π_+ and $\Pi_\times$.

D.4.1. Switching protocols over $\mathscr{R}^*$

We describe here the two 2-party switching protocols from an additive homomorphic encryption of m to a multiplicative one and *vice versa*. Contrary to Couteau *et al.*'s protocol [CPP16], the two protocols are actually the *same* since both the additive and the multiplicative scheme support a $\mathsf{ScalMul}$ operation and a *single-round* 2-party decryption. It is important to note that in our instantiation, the round complexity is only 2, since the first ciphertext $C^{(2)}_{R^{-1}}$ can be sent within the flow of the 2-party decryption which is only one round (cf. D.2.2 or Fig. D.9 and D.11). We suppose that $m \neq 0$ here, and more precisely the message to be switched lies in $\mathscr{R} \cap \mathscr{M} = \mathscr{R}^*$.

Switching protocols between $\Pi_1.\mathsf{Encrypt}(m)$ and $\Pi_2.\mathsf{Encrypt}(m)$

In Fig. D.3, as our switching protocols in the two directions are the same, the pair (Π_1, Π_2) is either $(\Pi_+, \Pi_\times)$ or $(\Pi_\times, \Pi_+)$.



Input : $pk^1, pk^2, sk^1_A, C^{(1)}_m$

Input : $pk^1, pk^2, sk^1_B, C^{(1)}_m$

$R \xleftarrow{\$} \mathscr{R}^*$

$C^{(1)}_{mR} = \Pi_1.\mathsf{ScalMul}(pk^1, C^{(1)}_m, R)$

$C^{(2)}_{R^{-1}} = \Pi_2.\mathsf{Encrypt}(pk^2, R^{-1})$ $\xrightarrow{C^{(2)}_{R^{-1}}}$

$\xrightarrow{2\mathsf{Dec}(pk^1, C^{(1)}_{mR}; sk^1_A; sk^1_B)}$ Bob gets mR

$\xleftarrow{C^{(2)}_m}$ $C^{(2)}_m = \Pi_2.\mathsf{ScalMul}(pk^2, C^{(2)}_{R^{-1}}, mR)$

Figure D.3: 2-party $\mathsf{Switch}_{1\to 2}$ from Π_1 to Π_2 where (Π_1, Π_2) is either $(\Pi_+, \Pi_\times)$ or $(\Pi_\times, \Pi_+)$.

The correctness of these two protocols is clear. They are generic and and the switch from $\Pi_\times$ to Π_+ is highly simpler than the one in [CPP16] (ours is 2-round instead of 6-round) and our instantiation will keep this simplicity. We prove in the following theorem that they are zero-knowledge, and the security proof itself is also very simple. It only relies on the zero-knowledge property of the shared decryptions.

Theorem D − 3. *The ESP between Π_+ and $\Pi_\times$, whose switching routines are described in Figure D.3, is zero-knowledge if Π_+ and $\Pi_\times$ are two compatible encryption schemes which have zero-knowledge one round 2-party decryptions.*

Proof. The proof consists in proving that after a share of the secret keys, both switching procedures are zero-knowledge for Alice and Bob. Let us start with the proof that the ESP is zero-knowledge for Alice. We are going to describe a simulator Sim whose behavior is indistinguishable from Alice's behavior in front of an adversarial Bob.
$\mathsf{Sim}_{\mathsf{Share}}$: The simulator receives as input the public key $(pk_+, pk_\times)$ and simulates the Share procedure as follows: it calls out the $\mathsf{Sim}^{2d}_{\mathsf{Share}}$ procedures of the zero-knowledge

property of Alice for 2-party decryption of respectively Π_+ and $\Pi_\times$ with pk_+ and $pk_\times$ as input. In particular it obtains $sk'_\mathrm{B} = (x^+_\mathrm{B}, x^\times_\mathrm{B})$ it can feed the adversary with. When Sim is requested for a switch, it receives a pair of twin ciphertexts $(\mathrm{C}, \bar{\mathrm{C}})$.

Game G_0**.** This game is the real game. The simulator generates all the secret shares in an honest way and gives his share to Bob. It plays honestly any switching protocols on an input $(\mathrm{C}, \bar{\mathrm{C}})$ using Alice's secret key.

Game G_1**.** The first modification concerns the *additive to multiplicative* direction. The setup and key generation are the same as in the previous game. When requested to participate to a $\mathsf{Switch}_{+\to\times}$, with $(\mathrm{C}, \bar{\mathrm{C}})$ as input, the simulator picks uniformally at random $x \in \mathscr{R}^*$ and sets $\mathrm{C}^+_{m\mathrm{R}} = \Pi_+.\mathsf{Encrypt}(x)$ and $\mathrm{C}^\times_{\mathrm{R}^{-1}} = \Pi_\times.\mathsf{ScalMul}(\bar{\mathrm{C}}, x^{-1})$. The simulator then concludes the protocol honestly. This game is indistinguishable from the previous since, as x is random, it is equivalent to a genuine protocol using $\mathrm{R} = x/m$, where m is the plaintext under C and $\bar{\mathrm{C}}$.

Game G_2**.** In this game, we modify the shared decryption for Π_+ using the simulator of 2-party decryption. First, the simulation gives the key x^+_B obtained by the simulation of the shares to Bob. Then after Sim simulated the pair $(\mathrm{C}^+_{m\mathrm{R}}, \mathrm{C}^\times_{\mathrm{R}^{-1}})$ as above, it uses the simulator $\mathsf{Sim}^{2d}_\mathrm{A}(\mathrm{C}^+_{m\mathrm{R}}, x, x^+_\mathrm{B}, \cdot)$ for the 2-party decryption of Π_+ to interact with Bob, where $\mathrm{C}^+_{m\mathrm{R}}$ is an encryption of x. Thanks to the property of this simulator this game is indistinguishable from the previous one (note that the key x^+_B is only used in that part of the protocol). Eventually, the last computation done by Bob, $\Pi_\times.\mathsf{ScalMul}(\mathrm{C}^\times_{\mathrm{R}^{-1}}, x)$ gives a multiplicative ciphertext of m equivalent to $\bar{\mathrm{C}}$.

Game G_3**.** In this game, we address the *multiplicative to additive* way. The setup and key generation are the same as in the previous games. As in Game G_1, when requested to participate to a $\mathsf{Switch}_{\times\to+}$, with $(\mathrm{C}, \bar{\mathrm{C}})$ as input, the simulator picks uniformally at random $x \in \mathscr{R}^*$ and sets $\mathrm{C}^\times_{m\mathrm{R}} = \Pi_\times.\mathsf{Encrypt}(x)$ and $\mathrm{C}^+_{\mathrm{R}^{-1}} = \Pi_+.\mathsf{ScalMul}(\bar{\mathrm{C}}, x^{-1})$. Then, Sim continues honestly the protocol. This game is indistinguishable from the previous one.

Game G_4**.** The shared $\Pi_\times$ decryption is modified as in Game G_2. The simulation now gives the simulated key $x^\times_\mathrm{B}$ to Bob and then uses the simulator for the 2-party decryption of $\Pi_\times$ with ciphertext $\mathrm{C}^\times_{m\mathrm{R}}$ and corresponding plaintext x. Thanks to the property of this simulator this game is indistinguishable from the previous one. Again, Bob's last computation $\Pi_+.\mathsf{ScalMul}(\mathrm{C}^+_{\mathrm{R}^{-1}}, x)$ gives a ciphertext equivalent to $\bar{\mathrm{C}}$.
In conclusion, the advantage of the attacker is negligible.

We now prove that the ESP is zero-knowledge for Bob, by describing a simulator Sim whose behavior is indistinguishable from Bob's behavior in front of an adversarial Alice. The simulator receives as input the public key $pk = (pk_+, pk_\times)$ and simulates the Share procedure as above and feed the adversary (Alice) with the corresponding secret key. When Sim is requested for a switch, it receives a pair of twin ciphertexts $(\mathrm{C}, \bar{\mathrm{C}})$.

In both directions, the simulation is trivial, since Bob's only flow is the final forward of the twin ciphertext (we have suppose that the 2-party decryption has only one round from Alice to Bob), which is done by sending the $\bar{\mathrm{C}}$ ciphertext. This is indistinguishable from a true execution since $\bar{\mathsf{C}}$ is a random ciphertext which encrypts the same plaintext that C. □

D.4.2. Modification of $\Pi_\times$ to embed the zero message

One technical issue to design switching protocols between Π_+ and $\Pi_\times$ is to embed the zero message into $\Pi_\times$'s message space so that the message spaces match. To do so, we need to modify the $\Pi_\times$ encryption. We will use a technique quite similar to those in [CPP16]: During their encryption, if the message m is equal to 0, a bit b is set to 1. It is set to 0 for any other message. Then, the message $m + b$ (which is never 0) is encrypted using their Elgamal encryption. As this encryption scheme is no longer injective, to discriminate an encryption of 0, the ciphertext is accompanied by two encryptions under classical Elgamal of T^b and T'^b where T, T' are random elements. We note that these two encryptions are in fact encryptions of b which are homomorphic for the or gate : If $b = 0$, we get an Elgamal encryption of 1 and if $b = 1$, an Elgamal encryption of a random element (which is equal to 1 only with negligible probability). Thanks to the multiplicativity of Elgamal, if we multiply an encryption of b and an encryption of b', we get an Elgamal encryption of 1 only if $b = b' = 0$ and an Elgamal encryption of a random element otherwise. In [CPP16] the second encryption of b is actually an extractable commitment, the corresponding secret key is only known by the simulator in the security proof.

In our case, we use the additively homomorphic encryption Π_+ to discriminate the zero message: Π_+ is used to encrypt a random element r if $m = 0$ and 0 otherwise. As a consequence, it will be possible to directly obtain an encryption of $\bar{b}$ (the complement of the bit b used during encryption) under Π_+ using the zero-testing procedure during the switch from $\Pi_\times^0$ to Π_+ (see Fig. D.7). This gives a real improvement compared to [CPP16] when we instantiate our generic protocols. As in [CPP16] we add a useless second encryption of r to be used by the simulation in the security proof. Our modification is formally described in Fig. D.4. The $\mathsf{Hom}_\times$ procedure is obtained by applying the $\mathsf{Hom}_\times$ procedures of $\Pi_\times$, and Π_+. For the $\mathsf{ScalMul}$ procedure, which corresponds to a multiplication by a plaintext α, it applies the $\mathsf{ScalMul}$ procedure of $\Pi_\times$ if $\alpha \neq 0$ and add an encryption of 0 to the additive part. If $\alpha = 0$, it outputs an encryption of 0.

The protocol $\Pi_\times^0$ directly inherits the indistinguishability under a chosen message attack from those of $\Pi_\times$ and Π_+. By a standard hybrid argument, we can prove the following theorem, whose proof is omitted.

Theorem D − 4. *If Π_+ and $\Pi_\times$ are IND-CPA, then $\Pi_\times^0$ is also IND-CPA.*

D.4.3. Full Switching protocols

Encrypted zero-test

In [CPP16] an encrypted zero test (EZT) to obliviously detect the zero messages during switches is presented. In our case, EZT takes as input a ciphertext C_m^+ from the additively homomorphic encryption Π_+ of a message m and outputs a Π_+ ciphertext C_b^+ of

```
Algo. $(\Pi_+ \rightleftharpoons \Pi^0_\times)$.KeyGen($1^\lambda$)
  1. params $\leftarrow (\Pi_+ \rightleftharpoons \Pi_\times)$.Setup($1^\lambda$)
  2. $((pk^\times, sk^\times), (pk^+, sk^+)) \leftarrow (\Pi_+ \rightleftharpoons \Pi_\times)$.KeyGen($1^\lambda$, params)
  3. $(pk', sk') \leftarrow \Pi_+$.KeyGen($1^\lambda$, params)
  4. Set $pk \leftarrow (pk^\times, pk^+, pk')$ and $sk \leftarrow (sk^\times, sk^+, sk')$
  5. Return $(pk, sk)$
```

```
Algo. $\Pi^0_\times$.Encrypt($pk, m$)
  1. Parse $pk$ as $(pk^\times, pk^+, pk')$
  2. If $m = 0$ set $b \leftarrow 1$ and $r \xleftarrow{\$} \mathscr{R}^*$
     otherwise set $b \leftarrow 0$ and $r \leftarrow 0$
  3. $C^\times_{m+b} \leftarrow \Pi_\times$.Encrypt($pk^\times, m + b$)
  4. $C^+_r \leftarrow \Pi_+$.Encrypt($pk^+, r$)
  5. $C'_r \leftarrow \Pi_+$.Encrypt($pk', r$)
  6. Return $(C^\times_{m+b}, C^+_r, C'_r)$
```

```
Algo. $\Pi^0_\times$.Decrypt($sk, (C^\times_{m+b}, C^+_r, C'_r)$)
  1. Parse $sk$ as $(sk^\times, sk^+, sk')$
  2. $B \leftarrow \Pi_+$.Decrypt($sk^+, C^+_r$)
  3. If $B \neq 0$ return 0
     else
     return $\Pi_\times$.Decrypt($sk^\times, C^\times_{m+b}$)
```

Figure D.4: $\Pi_\times$ over $\mathscr{R}$: $\Pi^0_\times$

a bit b equals to 1 if $m = 0$ and equals to 0 otherwise. The EZT has to be zero-knowledge in the sense that there exists an efficient simulator for each player which, on input a pair of twin ciphertext $(C, \bar{C})$, is indistinguishable from these honest players. This simulator runs without the secret share of the the user it simulates.

During the security proof, the simulator will obtain the bit b thanks to the knowledge of the secret key which decrypts the additional encryption of r appended during encryption (see Fig. D.4).

This EZT protocol is done using garbled circuits techniques. An alternative would be to use techniques based on homomorphic encryption [LT13]. The resulting protocol is described in Fig. D.5. The function $H : \mathscr{R}^* \longrightarrow \{0,1\}^\kappa$ (for a security parameter κ) belongs to a universal hash function family (in practice, this will be a reduction modulo 2^κ of the integer representation of an element of $\mathscr{R}$). We denote by eq the function that on input $(u, v) \in \{0,1\}^\kappa$ outputs 1 if $u = v$ and 0 otherwise and we denote by Garble(f) the computation of a garbled circuit evaluating the function f.

The correctness of the protocol comes from the fact that the last three lines of the protocol compute the encryption of $b_A \oplus b_B$ by homomorphically evaluating $b_A + b_B -$

Input : $pk, sk_{\mathrm{A}}, \mathsf{C}^+_m$		Input : pk, sk_{B}
$r \xleftarrow{\$} \mathscr{R}^*$		
$r' \leftarrow \mathrm{H}(r)$		
$\mathsf{C}^+_r \leftarrow \Pi_+.\mathsf{Encrypt}(pk, r)$		
$\mathsf{C}^+_{m+r} \leftarrow \Pi_+.\mathsf{Hom}_+(pk, \mathsf{C}^+_m, \mathsf{C}^+_r)$		
	$\xrightarrow{2\mathsf{Dec}(pk, \mathsf{C}^+_{m+r}; sk_{\mathrm{A}}; sk_{\mathrm{B}})}$	Bob gets $m+r$
		$r'' \leftarrow \mathrm{H}(m+r)$
$b_{\mathrm{A}} \xleftarrow{\$} \{0,1\}$		
$\mathsf{C}^+_{b_{\mathrm{A}}} \leftarrow \Pi_+.\mathsf{Encrypt}(pk, b_{\mathrm{A}})$		
$\mathscr{C}(\cdot) \leftarrow \mathsf{Garble}(b_{\mathrm{A}} \oplus \mathsf{eq}(r', \cdot))$	$\xrightarrow{\mathscr{C}(\cdot)}$	$b_{\mathrm{B}} \leftarrow \mathscr{C}(r'')$
	$\xleftarrow{\mathsf{C}^+_{b_{\mathrm{B}}}}$	$\mathsf{C}^+_{b_{\mathrm{B}}} \leftarrow \Pi_+.\mathsf{Encrypt}(pk, b_{\mathrm{B}})$
$\mathsf{C}^+_{-2b_{\mathrm{A}}b_{\mathrm{B}}} \leftarrow \Pi_+.\mathsf{ScalMul}(pk, \mathsf{C}^+_{b_{\mathrm{B}}}, -2b_{\mathrm{A}})$		
$\mathsf{C}^+_{b_{\mathrm{B}}-2b_{\mathrm{A}}b_{\mathrm{B}}} \leftarrow \Pi_+.\mathsf{Hom}_+(pk, \mathsf{C}^+_{b_{\mathrm{B}}}, \mathsf{C}^+_{-2b_{\mathrm{A}}b_{\mathrm{B}}})$		
$\mathsf{C}_b \leftarrow \Pi_+.\mathsf{Hom}_+(pk, \mathsf{C}^+_{b_{\mathrm{A}}}, \mathsf{C}^+_{b_{\mathrm{B}}-2b_{\mathrm{A}}b_{\mathrm{B}}})$		

Figure D.5: EZT: 2-party protocol to compute C^+_b from C^+_m

$2b_{\mathrm{A}}b_{\mathrm{B}}$ from the encryptions of b_{A} and b_{B}. By construction, $b_{\mathrm{A}} \oplus b_{\mathrm{B}} = \mathsf{eq}(r', r'')$ which is equals to 1 if $m = 0$ and 0 otherwise, with probability $1 - 2^{-\kappa}$. This is exactly the encryption of the bit b. This protocol is zero-knowledge (see [CPP16]). Using [KS08], the communication needed is $8\kappa^2$ bits of preprocessing for the garbled circuit and κ^2 bits and κ oblivious transfers for the online phase (cf. [CPP16, Fig. 4]).

D.4.4. 2-party ESP between $\Pi_+.\mathsf{Encrypt}(m)$ and $\Pi^0_\times.\mathsf{Encrypt}(m)$

The protocol of Fig. D.6 is quite similar to the one of [CPP16]. First we use the EZT sub-protocol to get a Π_+ encryption of the bit b. A notable difference with the protocol of [CPP16] is that this encryption of b can be used directly to set an element of the ciphertext for Π^0_+, which saves many rounds in the interaction. Since the bit b is encrypted twice (this second encryption is only used during the security proof), the ReEnc_+ protocols allows to re-encrypt the output of EZT to the right public key. Then, thanks to the homomorphic property of the Π_+ scheme, Alice can construct an additive encryption of $m + b$ and the $\mathsf{Switch}_{+\to\times}$ protocol of Fig. D.3 is used to get the Π_+-encryption of $m + b$. The two ciphertexts of b are randomized to get a proper multiplicative ciphertext.

In Fig. D.7, starting from a multiplicative ciphertext of m, we run an $\mathsf{Switch}_{\times\to+}$ with the first component of $\mathsf{C}^\times_m$, which is a $\Pi_\times$-encryption of $m + b$. Hence, we get C^+_{m+b}. Then, we run the EZT protocol on the second component C^+_r and the output the encryption of a bit b' whose value is 1 when $r = 0$, i.e., when $b = 0$ and 0 otherwise. Therefore $b' = \bar{b}$ and EZT actually outputs an encryption of $\bar{b}$. It is now possible to homomorphically remove the bit b remaining in the Π_+-encryption of $m + b$, C^+_{m+b}. Inspired by the implicit technique used in [CPP16], we use the $2\mathsf{Mul}_+$ protocol to obtain,

Input : $(pk^\times, pk^+, pk'), sk_\mathrm{A}, \mathrm{C}^+_m$		Input : $(pk^\times, pk^+, pk'), sk_\mathrm{B}, \mathrm{C}^+_m$
Alice gets C^+_b	$\xleftrightarrow{\mathsf{EZT}(\mathrm{C}^+_m)}$	
Alice gets C'_b	$\xleftrightarrow{\mathsf{ReEnc}_+(pk^+, pk', \mathrm{C}^+_b)}$	Bob gets C'_b
$\mathrm{C}^+_{m+b} = \Pi_+.\mathsf{Hom}_+(pk^+, \mathrm{C}^+_m, \mathrm{C}^+_b)$	$\xrightarrow{\mathrm{C}^+_{m+b}}$	
Alice gets $\mathrm{C}^\times_{m+b}$	$\xleftrightarrow{\mathsf{Switch}_{+\to\times}(\mathrm{C}^+_{m+b})}$	Bob gets $\mathrm{C}^\times_{m+b}$
$r, r' \xleftarrow{\$} \mathscr{R}^*$ $\mathrm{C}^+_{rb} = \Pi_+.\mathsf{ScalMul}(pk^+, \mathrm{C}^+_b, r)$ $\mathrm{C}'_{r'b} = \Pi_+.\mathsf{ScalMul}(pk', \mathrm{C}'_b, r')$		
$\mathrm{C}^\times_m = (\mathrm{C}^\times_{m+b}, \mathrm{C}^+_{rb}, \mathrm{C}'_{r'b})$	$\xrightarrow{\mathrm{C}^\times_m}$	

Figure D.6: 2-party $\mathsf{Switch}_{+\to\times}$ from Π_+ to $\Pi^0_\times$ over $\mathscr{R}^* \cup \{0\}$

Input : $pk, sk_\mathrm{A}, \mathrm{C}^\times_m = (\mathrm{C}^\times_{m+b}, \mathrm{C}^+_r, \mathrm{C}'_r)$		Input : $pk, sk_\mathrm{B}, \mathrm{C}^\times_m = (\mathrm{C}^\times_{m+b}, \mathrm{C}^+_r, \mathrm{C}'_r)$
Alice gets C^+_{m+b}	$\xleftrightarrow{\mathsf{Switch}_{\times\to+}(\mathrm{C}^\times_{m+b})}$	Bob gets C^+_{m+b}
Alice gets $\mathrm{C}^+_{\bar{b}}$	$\xleftrightarrow{\mathsf{EZT}(\mathrm{C}^+_r)}$	
Alice gets $\mathrm{C}^+_{\bar{b}(m+b)} = \mathrm{C}^+_m$	$\xleftrightarrow{\mathsf{2Mul}_+(\mathrm{C}^+_{m+b}, \mathrm{C}^+_{\bar{b}}; sk_\mathrm{A}; sk_\mathrm{B})}$	Bob gets $\mathrm{C}^+_{\bar{b}(m+b)} = \mathrm{C}^+_m$

Figure D.7: 2-party $\mathsf{Switch}_{\times\to+}$ from $\Pi^0_\times$ to Π_+ over $\mathscr{R}^* \cup \{0\}$

from $\mathsf{C}^+_{\bar{b}}$ and C^+_{m+b}, a Π_+-encryption of $(m+b)\bar{b}$ which is equal to a Π_+-encryption of m.

Note that we can not simply use the fact that $m + b + \bar{b} - 1 = m$ over $\mathbf{Z}$ to get m: The expression $m + b$ is really equal to the message m plus the bit b only for fresh multiplicative ciphertexts. After an homomorphic multiplication between a ciphertext of a non zero message with a ciphertext of zero it becomes something random. As a result, we have to multiply it by $\bar{b}$ to get 0.

The zero-knowledge property of our ESP essentially comes from the fact that each routine (ReEnc_+ and $\mathsf{2Mul}_+$) is individually zero-knowledge, inherited from the zero-knowledge of the 2-party decryption of the encryption protocols. We also use the fact that the encryption schemes are IND-CPA, in order to be able to simulate intermediate ciphertexts. This means that the assumptions in our theorem are weak and very natural.

Theorem D – 5. *The ESP between Π_+ and $\Pi^0_\times$ whose routines are described in Fig. D.6 and D.7 is zero-knowledge if Π_+ and $\Pi_\times$ are two compatible encryptions that are IND-CPA and whose 2-party decryptions are zero-knowledge, and* EZT *is zero-knowledge.*

Proof. (sketch) The full proof of this theorem can be found in Appendix D.B. This proof can be sketched as follows: First we give the secret key sk' to the simulation. From a pair of twin ciphertexts, it allows the simulation to know the bit that encode the fact that the plaintext is 0 or not. With that knowledge, the simulation can retrieve the ciphertexts that constitute the input and the output of each building block, and use their zero-knowledge simulator to emulate them. Then, we remove the knowledge of sk' from the simulation which replaces each input and output by random ciphertexts. Thanks to the IND-CPA property of the encryption schemes this is indistinguishable from the previous step. As a result, the whole protocol is simulated without knowing any secret. □

D.5. Instantiation of our Generic Construction on $\mathbf{Z}/p\mathbf{Z}$

In this section we provide an instantiation of our generic construction on a field $\mathbf{Z}/p\mathbf{Z}$ for a prime p, by describing an additively homomorphic encryption and a multiplicatively homomorphic one. Both schemes enjoy an Elgamal structure. For the additively homomorphic encryption scheme, we will use as a basis the scheme introduced in [CL15] (denoted CL in the following). It uses the notion of a DDH *group with an easy* DL *subgroup*, which is instantiated using class groups of quadratic fields. For the multiplicatively homomorphic scheme, we devise a variant of the traditional Elgamal encryption over the whole group $(\mathbf{Z}/p\mathbf{Z})^*$. Both schemes are described in the next subsection. We also describe their 2-party decryption, since it is required by the generic construction.

D.5.1. Additively Homomorphic Scheme over $\mathbf{Z}/p\mathbf{Z}$

Castagnos-Laguillaumie encryption

The encryption scheme from [CL15] is additively homomorphic modulo a prime p. The general protocol is well suited for relatively small p. For the ESP context, we need a large message space as p must be at least of 2048 bits for the security of the Elgamal protocol. As a result, we use the first variant of CL described in Section C.4 of [CL15]. This variant is defined with subgroups of the class group of an order of a quadratic field of discriminant $\Delta_p = -p^3$. Thus all computations are done in this class group. Note that elements are classes of ideals, that can be represented by their unique reduced elements, i.e., by two integers of roughly the size of $\sqrt{|\Delta_p|}$. As a consequence, a group element can be represented with $3 \log p$ bits.

We provide some improvements detailed in the following. The CL scheme does exponentiations to some random powers in a cyclic group of unknown order. Let us denote by $\mathfrak{g}$ a generator of this group. Only an upper bound B on this order is known. In order to make the result of these exponentiations look like uniform elements of the cyclic group, the authors of [CL15] choose to sample random exponents from a large enough uniform distribution, and more precisely over $\{0, \ldots, \mathrm{B}'\}$ where $\mathrm{B}' = 2^{\lambda-2}\mathrm{B}$, so that the resulting distribution is as distance to uniform less than $2^{-\lambda}$.

However, it is more efficient to use a folded discrete Gaussian Distribution instead of a folded uniform distribution. Let $z \in \mathbf{Z}$ and $\sigma > 0$ a real number and let us denote by $\rho_\sigma(z) = \exp(-\pi z^2/\sigma^2)$ a Gaussian centered function and define the probability mass function $\mathscr{D}_\sigma$ over $\mathbf{Z}$ by $\mathscr{D}_\sigma(z) := \rho_\sigma(z)/\sum_{z\in\mathbf{Z}} \rho_\sigma(z)$.

If z is sampled from $\mathscr{D}_\sigma$, we have $|z| > \tau\sigma$ with probability smaller than

$$\sqrt{2\pi e}\tau \exp(-\pi\tau^2)$$

(cf. [MR07, Lemma 2.10]). We denote by $\tau(\lambda)$ the smallest τ such that this probability is smaller than $2^{-\lambda}$.

If we set $\sigma = \sqrt{\ln(2(1+2^{\lambda+1}))/\pi}\mathrm{B}$, Lemma D−1 of Appendix D.C shows that the distribution obtained by sampling z from $\mathscr{D}_\sigma$ and computing $\mathfrak{g}^z$ is at distance less than $2^{-\lambda}$ to the uniform distribution in $\langle\mathfrak{g}\rangle$.

For instance for $\lambda = 128$, we only add, in the worst case, 6 iterations in the square and multiply algorithm to compute $\mathfrak{g}^z$, whereas one has to add 126 iterations with a folded uniform distribution.

Description of the scheme. We denote by CL.Gen a parameter generator for CL. It takes as input 1^λ and outputs a tuple $(p, \mathfrak{g}, \mathfrak{f}, \sigma)$. This tuple is such that p is a prime satisfying $p \equiv 3 \pmod 4$ so that computing discrete logarithms in $C(-p)$, the ideal class group of the quadratic order of discriminant $-p$, takes 2^λ times. Then $\mathfrak{g} \in C(-p^3)$ is a class of order ps where s is unknown and expected to be of the order of magnitude of the class number of $C(-p)$: a concrete implementation for $\mathfrak{g}$ is given in Fig. C.2 of [CL15]. It consists in generating a random ideal of the maximal order of discriminant $-p$, and lifting it in the order of discriminant $-p^3$. Eventually, $\mathfrak{f} \in C(-p^3)$ is the class of the ideal $p^2\mathbf{Z}+((-p+\sqrt{-p^3})/2)\mathbf{Z}$ and σ will be the standard deviation of the Gaussian Distribution discussed before: $\sigma = \sqrt{\ln(2(1+2^{\lambda+1}))/\pi}\mathrm{B}$, with $\mathrm{B} = \log(p)p^{3/2}/(4\pi)$.

The scheme relies on the notion of a DDH group with an easy DL subgroup. It is IND-CPA under the DDH assumption in the group generated by $\mathfrak{g}$. On the other hand, in the subgroup of order p generated by $\mathfrak{f}$, there is a polynomial time algorithm, denoted

CL.Solve which takes as input an element of $\langle \mathfrak{f} \rangle$ and which outputs its discrete logarithm in basis $\mathfrak{f}$. We refer the reader to [CL15] for concrete implementation of CL.Gen and CL.Solve. The resulted scheme is given in Fig. D.8.

```
CL.Setup($1^\lambda$)
  1. $(p, \mathfrak{g}, \mathfrak{f}, \sigma) \leftarrow$ CL.Gen($1^\lambda$)
  2. Return params := $(p, \mathfrak{g}, \mathfrak{f}, \sigma)$

CL.KeyGen
  1. Pick $x \xleftarrow{\$} \mathcal{D}_\sigma$ and set $\mathfrak{h} \leftarrow \mathfrak{g}^x$
  2. Set $pk \leftarrow \mathfrak{h}$ and set $sk \leftarrow x$.
  3. Return $(pk, sk)$

CL.Decrypt($sk, (\mathfrak{c}_1, \mathfrak{c}_2)$)
  1. Set $\mathfrak{M} \leftarrow \mathfrak{c}_2/\mathfrak{c}_1^x$
  2. $m \leftarrow$ CL.Solve($\mathfrak{M}$)
  3. Return $m$

CL.Encrypt($pk, m$)
  1. Pick $r \xleftarrow{\$} \mathcal{D}_\sigma$
  2. Compute $\mathfrak{c}_1 \leftarrow \mathfrak{g}^r$
  3. Compute $\mathfrak{c}_2 \leftarrow \mathfrak{f}^m \mathfrak{h}^r$
  4. Return $(\mathfrak{c}_1, \mathfrak{c}_2)$

CL.Hom$_+$($pk, (\mathfrak{c}_1, \mathfrak{c}_2), (\mathfrak{c}'_1, \mathfrak{c}'_2)$)
  1. Pick $r \xleftarrow{\$} \mathcal{D}_\sigma$
  2. Return $(\mathfrak{c}_1 \mathfrak{c}'_1 \mathfrak{g}^r, \mathfrak{c}_2 \mathfrak{c}'_2 \mathfrak{h}^r)$

CL.ScalMul($pk, (\mathfrak{c}_1, \mathfrak{c}_2), \alpha$)
  1. Pick $r \xleftarrow{\$} \mathcal{D}_\sigma$
  2. Return $(\mathfrak{c}_1^\alpha \mathfrak{g}^r, \mathfrak{c}_2^\alpha \mathfrak{h}^r)$
```

Figure D.8: Castagnos-Laguillaumie over $\mathbf{Z}/p\mathbf{Z}$: CL

Theorem D – 6 ([CL15]). *The* CL *scheme of Fig. D.8 is an additively homomorphic encryption scheme over $\mathbf{Z}/p\mathbf{Z}$, IND-CPA under the* DDH *assumption in the ideal class group of the quadratic order of discriminant $-p^3$.*

One round 2-party decryption for CL

We now devise in Fig. D.9 a one round 2-party decryption for CL as defined in Section D.2.2, i.e. subroutines to share the secret key and the interactive protocol for decryption. As the scheme has an Elgamal structure, it can be readily adapted from the threshold variant of the original Elgamal scheme (cf. [Ped91] for instance) with a simple additive secret sharing of the key $x = x_A + x_B$. However, as the group order is unknown, this secret sharing must be done over the integers. This kind of sharing has been addressed before (cf. [DM10, Section 4] for instance).

As x is sampled from $\mathcal{D}_\sigma$, we saw before that $x \in [-\tau(\lambda)\sigma, \tau(\lambda)\sigma]$ for a small $\tau(\lambda)$ except with negligible probability. Then the integer x_A is taken uniformly at random in

$\mathsf{CL.Share}(sk, pk)$

1. Parse $x = sk$
2. Pick $x_A \overset{\$}{\leftarrow} \{-\tau(\lambda)\sigma 2^\lambda, \dots, \tau(\lambda)\sigma 2^\lambda\}$ and set $x_B = x - x_A$
3. Return $(pk, sk_A = x_A, sk_B = x_B)$

$\mathsf{CL.2Dec}(pk, C = (\mathfrak{c}_1, \mathfrak{c}_2); sk_A = x_A; sk_B = x_B)$

Input : pk, sk_A, C		Input : pk, sk_B
$\mathfrak{c}_{1,A} \leftarrow \mathfrak{c}_1^{x_A}$	$\xrightarrow{C,\, \mathfrak{c}_{1,A}}$	Compute $\mathfrak{M} \leftarrow \mathfrak{c}_2/(\mathfrak{c}_1^{x_B}\mathfrak{c}_{1,A})$
		$m \leftarrow \mathsf{CL.Solve}(\mathfrak{M})$
		return m

Figure D.9: 2-party Decryption for CL

the interval $[-\tau(\lambda)\sigma 2^\lambda, \tau(\lambda)\sigma 2^\lambda]$, and $x_B = x - x_A$. This choice makes the secret sharing private. Note that in that case, there is no gain in using a Gaussian Distribution to generate the shares. We refer the interested reader to Appendix D.D for details.

Theorem D − 7. *The 2-party Decryption for* CL *described in Fig. D.9 is correct and zero-knowledge.*

Proof. Correctness follows from the shared exponentiation. Let us prove first that the protocol is zero-knowledge for Alice (see Def. D − 7 in Appendix D.A).

For the secret key shares, the simulator $\mathsf{Sim}^{2d}_{\mathsf{Share}}$ picks x' from $\mathscr{D}_\sigma$:

$$x'_A \overset{\$}{\leftarrow} \{-\tau(\lambda)\sigma 2^\lambda, \dots, \tau(\lambda)\sigma 2^\lambda\},$$

and set $x'_B = x' - x'_A$. As the secret sharing is private, the distribution of x'_B is statistically indistinguishable from the distribution of the real x_B (see Appendix D.D for the computation of the statistical distance).

Then we describe the simulator Sim^{2d}_A which emulates Alice. From a ciphertext C, a plaintext m, it computes $\mathfrak{M} = \mathfrak{f}^m$. Then it simulates $\mathfrak{c}_{1,A}$ by setting $\mathfrak{c}_{1,A} = \mathfrak{c}_2/(\mathfrak{M}\mathfrak{c}_1^{x'_B})$, so that Bob's computations leads to $\mathfrak{M}$. The value sent by the simulation is thus perfectly indistinguishable from the real one.

It is straightforward to see that the protocol is zero-knowledge for Bob: secret key shares are simulated as previously, and x'_A is obviously indistinguishable from the real x_A, and then Bob sends nothing during the protocol. □

D.5.2. Multiplicatively Homomorphic Scheme over $\mathbf{Z}/p\mathbf{Z}$

Elgamal over $(\mathbf{Z}/p\mathbf{Z})^*$

Let q be an odd Sophie Germain prime, and let us denote by p the associated prime, *i.e.*, $p = 2q + 1$. The DDH assumption is widely supposed to hold in the subgroup of order q of $(\mathbf{Z}/p\mathbf{Z})^*$ which is the subgroup of quadratic residues modulo p, denoted S_p. The Elgamal cryptosystem defined in S_p is multiplicatively homomorphic and semantically secure if the DDH assumption holds in that subgroup.

It is well-known that the DDH assumption does not hold in the whole group $(\mathbf{Z}/p\mathbf{Z})^*$. As a result, in order to extend the message space to $(\mathbf{Z}/p\mathbf{Z})^*$, we need to encode elements of $(\mathbf{Z}/p\mathbf{Z})^*$ as quadratic residues. The situation is quite similar to the Elgamal over $(\mathbf{Z}/n\mathbf{Z})^*$ of [CPP16], but actually simpler to handle since we work modulo a prime p and not modulo an RSA integer n (in particular, we can publicly compute square roots or distinguish quadratic and non quadratic residues and we do not have to hide the factorization of n).

Since $p = 2q + 1$, we have $p \equiv 3 \pmod 4$, and -1 is not a quadratic residue modulo p. Let $m \in (\mathbf{Z}/p\mathbf{Z})^*$, let us denote by (m/p) the Legendre symbol of m modulo p. Then $(m/p) \times m$ is a quadratic residue mod p. Let L be the group morphism from $((\mathbf{Z}/p\mathbf{Z})^*, \times)$ to $(\mathbf{Z}/2\mathbf{Z}, +)$ that maps m to 0 (resp. to 1) if m is a quadratic residue (resp. is a non quadratic residue). The map

$$\begin{array}{ccccc} ((\mathbf{Z}/p\mathbf{Z})^*, \times) & \longrightarrow & (\mathrm{S}_p, \times) & \times & (\mathbf{Z}/2\mathbf{Z}, +) \\ m & \longmapsto & \big((m/p) \times m & , & \mathrm{L}(m)\big) \end{array}$$

is a group isomorphism. As a consequence we can encode elements of $(\mathbf{Z}/p\mathbf{Z})^*$ as a square plus one bit. The square can be encrypted with the traditional Elgamal encryption, and the bit $\mathrm{L}(m)$ has to be encrypted separately. In order to have a multiplicatively homomorphic encryption, $\mathrm{L}(m)$ has to be encrypted with a scheme that is homomorphic for the addition in $\mathbf{Z}/2\mathbf{Z}$.

We choose Goldwasser-Micali encryption [GM84] for that. The drawback is that we need an additional assumption, namely the Quadratic Residuosity assumption (QR) for the security of our protocol.

To avoid that, an idea could have been to encrypt $\mathrm{L}(m)$ as an integer in the exponent with another Elgamal scheme or with the additive scheme of the previous Subsection. However, after computing the product of ℓ messages $m_1, \dots, m_\ell$ over encrypted data, the decryption would give more information than the Legendre symbol of the product of the m_i's, namely $\sum_{i=1}^{n} \mathrm{L}(m_i)$ in the integers, instead of modulo 2. Moreover, this extra information has to be taken into account to devise a zero-knowledge 2-party decryption. As this information can not be simulated, this gives a complex 2-party protocol, perhaps by using an extra homomorphic encryption scheme like in [CPP16]. Note that a solution consisting in randomizing $\mathsf{L}(m)$ by adding a (small) even integer, with a Gaussian Distribution, for instance, still leaks the number ℓ of multiplications that have been made.

As a result, it seems to be an interesting open problem to devise an encryption scheme that allows homomorphic addition in $\mathbf{Z}/2\mathbf{Z}$, or that simulates it without leaks, without relying on a factorization-based assumption (in [CPP16], the same problem was handled more smoothly thanks to the fact that the authors worked with a composite modulus).

Description of the scheme. Let λ be a security parameter. Let $\mathsf{GM.Gen}$ be a parameter generator for the Goldwasser-Micali encryption scheme. It takes as input 1^λ and outputs (N, p', q') such that $p', q' \equiv 3 \pmod 4$ are primes and $\mathrm{N} = p'q'$ is such that factoring N takes 2^λ time. We use the threshold variant of Goldwasser-Micali described in [KY02] to define a suitable 2-party decryption.

We also define $\mathsf{Eg}^*.\mathsf{Gen}$ a parameter generator for Elgamal. It takes as input 1^λ and outputs (p, q, g) such that q is a prime, $p = 2q + 1$ is a prime such that computing discrete logarithms in $(\mathbf{Z}/p\mathbf{Z})^*$ takes 2^λ times, and g a generator of S_p, *i.e.*, and element of $(\mathbf{Z}/p\mathbf{Z})^*$ of order q. We depict in Fig. D.10, the adaptation of Elgamal over the whole multiplicative group $(\mathbf{Z}/p\mathbf{Z})^*$, denoted Eg^*.

The following theorem is a consequence of the previous discussion and the properties of the Goldwasser-Micali variant. Note that modulo N, $c_3^{(\mathrm{N}-p'-q'+1)/4}$ equals 1 if c_3 is a quadratic residue, and -1 if c_3 has Jacobi symbol 1 and is not a quadratic residue.

Theorem D – 8. *The* Eg^* *scheme of Fig. D.10 is multiplicatively homomorphic over* $(\mathbf{Z}/p\mathbf{Z})^*$*, and it is IND-CPA under the* DDH *assumption in the subgroup of quadratic residues of* $(\mathbf{Z}/p\mathbf{Z})^*$ *and the* QR *assumption in* $(\mathbf{Z}/\mathrm{N}\mathbf{Z})^\times$.

One round 2-party decryption for Eg^*

We describe in Fig. D.11 a one round 2-party decryption for Eg^*. This protocol is adapted from the threshold variant of the original Elgamal scheme and the basic threshold Goldwasser-Micali of [KY02, Subsection 3.1].

This simple protocol gives a huge performance improvement compared to the Elgamal over $(\mathbf{Z}/n\mathbf{Z})^*$ of [CPP16]: in that work, after the exponentiations, a CRT reconstruction is needed to recover m, and a quantity that leads to the factorization of n must be shared. To make this 2-party reconstruction zero-knowledge, the authors use an additional additively homomorphic encryption, and have to do the reconstruction over encrypted data. As a result, the protocol is very complex (implicitly described in [CPP16, Fig. 2]) with 5 rounds instead of 1.

Theorem D – 9. *The 2-party Decryption for* Eg^* *described in Fig. D.11 is correct and zero-knowledge.*

```
$\mathsf{Eg}^*.\mathsf{Setup}(1^\lambda)$
    1. Set $(p, q, g) \leftarrow \mathsf{Eg}^*.\mathsf{Gen}(1^\lambda)$
    2. Return params := $(p, q, g)$

$\mathsf{Eg}^*.\mathsf{KeyGen}$
    1. Set $(N, p', q') \leftarrow \mathsf{GM.Gen}(1^\lambda)$
    2. Add N to params
    3. Pick $x \xleftarrow{\$} \{0, \dots, q-1\}$
    4. Set $h \leftarrow g^x$
    5. Set $pk \leftarrow h$
    6. Set $sk \leftarrow (x, p', q')$
    7. Return $(pk, sk)$

$\mathsf{Eg}^*.\mathsf{Decrypt}(sk, (c_1, c_2, c_3))$
    1. Set $M \leftarrow c_2/c_1^x \pmod{p}$
    2. Set $L \leftarrow c_3^{(N-p'-q'+1)/4} \pmod{N}$
    3. If $L = 1$ return M else return $-M$
```

```
$\mathsf{Eg}^*.\mathsf{Encrypt}(pk, m)$
    1. Pick $r \xleftarrow{\$} \{1, \dots, q-1\}$
    2. Pick $r' \xleftarrow{\$} \{1, \dots, N-1\}$
    3. Set $c_1 \leftarrow g^r \pmod{p}$
    4. Set $c_2 \leftarrow (m/p) m h^r \pmod{p}$
    5. Set $c_3 \leftarrow (-1)^{L(m)} r'^2 \pmod{N}$
    6. Return $(c_1, c_2, c_3)$

$\mathsf{Eg}^*.\mathsf{Hom}_\times(pk, (c_1, c_2, c_3), (c_1', c_2', c_3'))$
    1. Pick $r \xleftarrow{\$} \{0, \dots, q-1\}$
    2. Pick $r' \xleftarrow{\$} \{1, \dots, N-1\}$
    3. Return $(c_1 c_1' g^r, c_2 c_2' h^r, c_3 c_3' r'^2)$

$\mathsf{Eg}^*.\mathsf{ScalMul}(pk, (c_1, c_2, c_3), \alpha)$
    1. Pick $r \xleftarrow{\$} \{0, \dots, q-1\}$
    2. Pick $r' \xleftarrow{\$} \{1, \dots, N-1\}$
    3. Set $c_1' \leftarrow c_1 g^r \pmod{p}$
    4. Set $c_2' \leftarrow (\alpha/p) \alpha c_2 h^r \pmod{p}$
    5. Set $c_3' \leftarrow (-1)^{L(\alpha)} c_3 r'^2 \pmod{N}$
    6. Return $(c_1', c_2', c_3')$
```

Figure D.10: Elgamal over $(\mathbf{Z}/p\mathbf{Z})^*$: Eg^*

$\mathsf{Eg}^*.\mathsf{Share}(sk, pk)$

```
1. Parse $(x, p', q') = sk$
2. Pick $x_A \xleftarrow{\$} \{0, \dots, q-1\}$ and set $x_B \equiv x - x_A \pmod q$
3. Pick $p_A, p_B, q_A, q_B \xleftarrow{\$} \{0, \dots, N\}$ such that $p_A \equiv p_B \equiv q_A \equiv q_B \equiv 0 \pmod 4$
4. Set $p_0 = p' - p_A - p_B$ and $q_0 = q' - q_A - q_B$
5. Set $pk \leftarrow (pk, N_0 = (N - p_0 - q_0)/4)$
6. Return $(pk, sk_A = (x_A, x'_A), sk_B = (x_B, x'_B))$
```

$\mathsf{Eg}^*.\mathsf{2Dec}(pk, C = (c_1, c_2, c_3); sk_A = (x_A, p_A, q_A); sk_B = (x_B, p_B, q_B))$

Input : pk, sk_A, C		Input : pk, sk_B
$c_{1,A} \leftarrow c_1^{x_A} \pmod p$		
$c_{3,A} \leftarrow c_3^{(-p_A - q_A)/4} \pmod N$		
	$\xrightarrow{C, c_{1,A}, c_{3,A}}$	Set $M \leftarrow c_2/(c_1^{x_B} c_{1,A}) \pmod p$
		Set $L \leftarrow c_{3,A} c_3^{(-p_B - q_B)/4} C_3^{N_0} \pmod N$
		If $L = 1$ return M else return $-M$

Figure D.11: 2-party Decryption for Eg^*

Proof. The proof is similar to the proof of Theorem D−7. For the Elgamal part of the protocol, secret key shares are simply taken uniformly at random in $\{0, \dots, q-1\}$, and the value sent by Alice is computed as $c_{1,A} = c_2/(Mc_1^{x_B})$, where $M = (m/p)m$. The Goldwasser-Micali part of the protocol is also simulated in a similar fashion from c_3 and $L(m)$ and the key share from a fake factorization of N just as in [KY02, Subsection 3.1] □

Extension of the message space from $(\mathbf{Z}/p\mathbf{Z})^*$ to $\mathbf{Z}/p\mathbf{Z}$

We use the generic construction depicted in Fig. D.4 with the additively homomorphic scheme described in the previous subsection. We denote by $\mathsf{Eg}_p.\mathsf{Gen}$, a group generator which combines the generators for Eg^* and CL : on input 1^λ, it first runs $\mathsf{Eg}^*.\mathsf{Gen}$, which outputs (p, q, g). The prime p equals $3 \mod 4$ and is such that computing discrete logarithms in $(\mathbf{Z}/p\mathbf{Z})^*$ takes time 2^λ. As the best algorithms for computing such discrete logarithms are faster than the algorithms for computing discrete logarithms in the class group $C(-p)$ (the sub-exponential complexity is respectively $L_p[1/3, (64/9)^{1/3} + o(1)]$ and $L_p[1/2, 1 + o(1)]$, see [Tho12, Jac00]), this prime p is compatible with the prime generated by CL.Gen. As a result Eg_p.Gen executes CL.Gen by setting this prime p and adapts the others quantities accordingly. The resulting scheme is described in Fig. D.12 for completeness.

D.5.3. ESP over $\mathbf{Z}/p\mathbf{Z}$: Efficiency and Comparisons

In Table D.1 we give the round complexity (*rc*) and bit complexity (*bc*) of our algorithms and we compare our full ESP protocols with that of Couteau et al. [CPP16]. For sake of clarity, and because it is identical to that of [CPP16], we omit the complexities resultant from the garbled circuit-based EZT protocols. Our 2-party decryption algorithms (both for CL and Eg*) only require 1 round. Note that we carefully analyzed the interactive algorithms so as to gather consecutive flows when possible within a single round. For example our $\mathsf{2Mul}_+$ and ReEnc_+ protocols (see Fig. D.1 and D.2) require only 2 rounds since Alice can send $C^+_{r\chi}$ (resp. C'_{-r}) and her 2Dec data simultaneously. For the same reason, our generic switches in $\mathscr{R}^*$ also require 2 rounds. Therefore, our $(\Pi_+ \rightleftharpoons \Pi^0_\times).\mathsf{Switch}_{+\to\times}$ needs 7 rounds: 2 for the initial EZT, 2 for ReEnc_+, 2 for sending C^+_{m+b} and the $\mathsf{Switch}_{+\to\times}$, and 1 for sending the final result to Bob. In the other direction, the initial $\mathsf{Switch}_{\times\to+}$ and the EZT are independent and can thus be processed simultaneously in 2 rounds. Adding 2 rounds for the $\mathsf{2Mul}_+$, the round complexity for $(\Pi_+ \rightleftharpoons \Pi^0_\times).\mathsf{Switch}_{\times\to+}$ adds up to 4 rounds only. In comparison, and using the same optimizations, the ESP switches from Couteau et al. requires 7 and 11 rounds respectively.

We express the communication cost in terms of the number of bits exchanged between the parties. The bit complexity (*bc*) is given as a function of the ring/field size. Observe that although, the best (conjectured) asymptotic complexity to compute a discrete logarithm in the ideal class group used in CL is in $L_p[1/2, 1+o(1)]$ (see. [Jac00]), one must consider a prime p that is large enough to guarantee that the DLP over $(\mathbf{Z}/p\mathbf{Z})^*$ is hard, i.e such that $L_p[1/3, (64/9)^{1/3} + o(1)] > 2^\lambda$ (see e.g. [Tho12]). In Table D.1, ℓ, represents the bit length of p for our protocol over $\mathbf{Z}/p\mathbf{Z}$ and of n for Couteau et al.'s protocol over $\mathbf{Z}/n\mathbf{Z}$.

For our protocols, we give the bit complexities for two variants: for the version of CL used in this paper *bc* is the cost deduced from Fig. D.8. The drawback of this scheme is that ciphertexts are represented with 2 elements of $C(-p^3)$ which gives $2\times2\times\frac{3}{2}\times\ell = 6\ell$ against 2ℓ for Paillier. Therefore, we include a column with the cost *bc*' that correspond to the so-called "faster variant" of CL from Section C.4 of [CL15]. This variant defines ciphertexts in $C(-p) \times C(-p^3)$, represented with $\ell + 3\ell = 4\ell$ elements. Moreover, for 2-party decryption we only have to share an exponentiation in $C(-p)$ instead of $C(-p^3)$ so the cost drops from $6\ell + 3\ell = 9\ell$ to $4\ell + \ell = 5\ell$.

For the former variant, the security depends upon DDH in $C(-p^3)$ whereas for the faster variant it is based upon the following indistinguishability argument: Let $\mathfrak{g}$ be a generator of a subgroup of $C(-p)$. After having chosen m, the adversary is

```
$\mathsf{Eg}_p.\mathsf{Setup}(1^\lambda)$

  1. Set $(p, q, g, \mathfrak{g}, \mathfrak{f}, \sigma) \leftarrow \mathsf{Eg}^*.\mathsf{Gen}(1^\lambda)$
  2. Return params := $(p, q, g, \mathfrak{g}, \mathfrak{f}, \sigma)$

$\mathsf{Eg}_p.\mathsf{KeyGen}$

  1. Set $(\mathrm{N}, p', q') \leftarrow \mathsf{GM.Gen}(1^\lambda)$
  2. Add N to params
  3. Pick $x^\times \xleftarrow{\$} \{0, \dots, q-1\}$
  4. Set $h^\times \leftarrow g^{x^\times}$
  5. Pick $x^+ \xleftarrow{\$} \mathscr{D}_\sigma$ and set $\mathfrak{h}^+ \leftarrow \mathfrak{g}^{x^+}$
  6. Pick $x' \xleftarrow{\$} \mathscr{D}_\sigma$ and set $\mathfrak{h}' \leftarrow \mathfrak{g}^{x'}$
  7. Set $pk \leftarrow (h^\times, \mathfrak{h}^+, \mathfrak{h}')$
  8. Set $sk \leftarrow (x^\times, p', q', x^+, x')$
  9. Return $(pk, sk)$

$\mathsf{Eg}_p.\mathsf{Decrypt}(sk, (c_1, c_2, c_3))$

  1. Set $\mathfrak{M} \leftarrow \mathfrak{c}_2/\mathfrak{c}_1^{x^+}$
  2. Set $\mathrm{B} \leftarrow \mathsf{CL.Solve}(\mathfrak{M})$
  3. If $\mathrm{B} \neq 0$ return 0
  4. Set $\mathrm{M} \leftarrow c_2/c_1^{x^\times} \pmod{p}$
  5. Set $\mathrm{L} \leftarrow c_3^{(\mathrm{N}-p'-q'+1)/4} \pmod{\mathrm{N}}$
  6. If $\mathrm{L} = 1$ return M else return $-\mathrm{M}$
```

```
$\mathsf{Eg}_p.\mathsf{Encrypt}(pk, m)$

  1. If $m = 0$ set $b \leftarrow 1$ and $r \xleftarrow{\$} (\mathbf{Z}/p\mathbf{Z})^*$
     otherwise set $b \leftarrow 0$ and $r \leftarrow 0$
  2. Pick $r^\times \xleftarrow{\$} \{1, \dots, q-1\}$
  3. Pick $r^{\times\prime} \xleftarrow{\$} \{1, \dots, \mathrm{N}-1\}$
  4. Set $c_1 \leftarrow g^{r^\times} \pmod{p}$
  5. Set $c_2 \leftarrow ((m+b)/p)(m+b)h^{\times r^\times}$ (mod $p$)
  6. Set $c_3 \leftarrow (-1)^{\mathrm{L}(m+b)} r^{\times\prime 2} \pmod{\mathrm{N}}$
  7. Pick $r^+ \xleftarrow{\$} \mathscr{D}_\sigma$
  8. Compute $\mathfrak{c}_1 \leftarrow \mathfrak{g}^{r^+}, \mathfrak{c}_2 \leftarrow \mathfrak{f}^r \mathfrak{h}^{+r^+}$
  9. Pick $r' \xleftarrow{\$} \mathscr{D}_\sigma$
  10. Compute $\mathfrak{c}'_1 \leftarrow \mathfrak{g}^{r'}, \mathfrak{c}'_2 \leftarrow \mathfrak{f}^r \mathfrak{h}'^{r'}$
  11. Return $(c_1, c_2, c_3, \mathfrak{c}_1, \mathfrak{c}_2, \mathfrak{c}'_1, \mathfrak{c}'_2)$
```

Figure D.12: Elgamal over $\mathbf{Z}/p\mathbf{Z}$: Eg_p

asked to distinguished the following distributions : $\{(\mathfrak{g}^x, \mathfrak{g}^y, \psi(\mathfrak{g}^{xy})), x, y \leftarrow \mathscr{D}_{\sigma/p}\}$ and $\{(\mathfrak{g}^x, \mathfrak{g}^y, \psi(\mathfrak{g}^{xy})\mathfrak{f}^m), x, y \leftarrow \mathscr{D}_{\sigma/p}\}$, where $\mathscr{D}_\sigma$ is the Gaussian Discrete distribution defined in Subsection D.5.1 and ψ is a lifting map from $C(-p)$ to $C(-p^3)$, defined in Lemma C−3 of [CL15]. We denote LDDH by the corresponding assumption. The algorithmic assumptions required for each protocol are presented in Table D.2.

Table D.1: Comparisons of the round complexities and bit complexities of our protocols (v1 and v2) with that of Couteau et al. [CPP16]. (∗) For the EZT protocol the communication cost for the garbled circuit is omitted as it is the same for v1, v2 and [CPP16] (cf. Subsection D.4.3 for the cost).

	round complexity		bit complexity		
Algorithms	this work	[CPP16]	v1	v2	[CPP16]
$\mathsf{Eg}^*.\mathsf{2Dec}$	1	n/a	5ℓ	5ℓ	n/a
$\mathsf{CL.2Dec}$	1	n/a	9ℓ	5ℓ	n/a
$\mathsf{CL.EZT}(*)$	2	n/a	15ℓ	9ℓ	n/a
$\mathsf{CL.2Mul}_+$	2	n/a	21ℓ	13ℓ	n/a
$\mathsf{CL.ReEnc}_+$	2	n/a	21ℓ	13ℓ	n/a
$(\Pi_+ \rightleftharpoons \Pi_\times).\mathsf{Switch}_{+\to\times}$	2	2	15ℓ	11ℓ	10ℓ
$(\Pi_+ \rightleftharpoons \Pi_\times).\mathsf{Switch}_{\times\to+}$	2	6	17ℓ	13ℓ	36ℓ

	round complexity		bit complexity		
ESP protocols	this work	[CPP16]	v1	v2	[CPP16]
$(\Pi_+ \rightleftharpoons \Pi_\times^0).\mathsf{Switch}_{+\to\times}$	7	7	69ℓ	45ℓ	37ℓ
$(\Pi_+ \rightleftharpoons \Pi_\times^0).\mathsf{Switch}_{\times\to+}$	4	11	53ℓ	35ℓ	61ℓ

Table D.2: Algorithmic assumptions

This work (v1)	This work (v2)	[CPP16]
DDH in $C(-p^3)$	LDDH in $C(-p^3)$	DCR
DDH in S_p	DDH in S_p	DDH in S_n
QR	QR	QR

D.6. ESP secure against malicious adversaries

To reach the security against malicious adversaries, it is necessary to add zero-knowledge proofs by all parties that every computation is done correctly with the knowledge of every plaintext. In [CPP16], the zero-knowledge proofs are classical Schnorr-like proofs and range proofs, but they need also to design a new strong primitive called *twin ciphertext proof* (TCP) to prove that a pair of ciphertexts from two different encryption schemes is actually a pair of twin ciphertexts. This allows to avoid generic circuit-based zero-knowledge proofs, but still requires a costly cut-and-choose technique (which can be amortized). This proof consists first in gathering a large pool of random genuine twin ciphertexts (proved thanks to the knowledge of the plaintext and the randomness, and of the homomorphic property of the encryption schemes). This part is done once for all. During an ESP, each time a twin ciphertext proof is needed, a fresh twin ciphertext pair is taken from the pool to perform a simple co-linearity proof.

To enhance our generic construction against malicious adversaries, we use the same method. In fact, the additional properties needed for the homomorphic encryption schemes are the same as in [CDN01]: the Π_+ and the $\Pi_\times$ encryption schemes must support zero-knowledge proof of plaintext knowledge, proof that the ScalMul operation has been performed correctly and also support a 2-party decryption in the malicious setting. Then we use the TCP technique as in [CPP16] for twin ciphertext proofs.

As a result, we modify our generic construction by adding such proofs in each step of the switching protocols. This ensures honest behavior and thus make the ESP secure in the malicious settings. In particular this brings soundness in the sense of [CPP16]: no malicious player can force the output of an ESP not to be a twin ciphertext.

The protocols Π_+ and $\Pi_\times$ described in the instantiation from the previous section support the required features. For the $\Pi_\times$ encryption scheme, we need zero knowledge proofs and 2-party decryption secure against malicious adversary for the classical Elgamal and for the Goldwasser-Micali encryption scheme. This can be done with classical methods: zero-knowledge proof *à la* Schnorr, adding verification keys to the public keys for 2-party decryption and proof of exponentiations to the same power. Note that for Goldwasser-Micali, we need to modify key generation to use strong primes p' and q' as in [KY02].

For the Π_+ encryption scheme which is based on the Castagnos-Laguillaumie encryption scheme, we need proofs for an Elgamal variant in a group of unknown order, namely a class group of a quadratic order. Then 2-party decryption secure against malicious adversary is obtained as for the $\Pi_\times$ scheme.

Generalizations of Schnorr proofs in group of unknown orders have been addressed extensively in [CKY09]. In this framework, a generalized Schnorr proof can be used if the cyclic group considered is what is called a *safeguard group*, which is roughly a group whose set of small orders elements is small and known, and for which it is hard to find roots of elements. The case of class groups has been explicitly taken in account for example in [DK02,DF02], where it is argue in particular that class groups of discriminant $-p$, $C(-p)$, can be considered to have the properties of safeguard groups. As a result, we can apply directly the framework of [CKY09] for the faster variant of CL mentioned in Subsection D.5.3 as exponentiations are defined in $C(-p)$ for this variant.

D.7. Conclusion

The encryption switching protocol is a promising cryptographic primitive formalized by Couteau et al. in [CPP16]. We propose in this article a generic framework to build such an ESP. Our approach makes the design of an ESP simple and efficient. In particular, we propose an instantiation whose round complexity is dramatically improved compared to Couteau et al., since we reduce by a factor 3 the number of round in the multiplicative to additive direction (while we have the same number of rounds in the other way). Again, in terms of bit complexity, our switching protocol in the multiplicative to additive direction gains a factor almost 1.7, while in the other direction Couteau et al.'s switch is smaller by a factor 1.2. This is essentially because in our case, the additively homomorphic encryption has large ciphertexts. In particular, any additively homomorphic encryption satisfying the conditions of our construction with smaller elements will allow to gain in terms of bit complexity. Our instantiation, which is secure in the semi-honest model under classical assumptions can be extended to the malicious case. We believe that it is possible to improve our instantiation by deviating a bit more from the generic construction. Moreover, an interesting open problem is to design an encryption scheme which is homomorphic for the + in $\mathbf{F}_2$ without the factorization assumption. A consequence could be to have an ESP whose security relies only on a discrete logarithm related assumption. Designing a more efficient encrypted zero-test is also a direction which will allow a significant improvement in the protocol.

Acknowledgments: The authors would like to thank Geoffroy Couteau for fruitful discussions, carreful reading and constructive comments on the preliminary version of this work. We also express our thanks to Bruno Grenet, Romain Lebreton, Benoît Libert and Damien Stehlé for their feedbacks.

The authors are supported in part by the French ANR ALAMBIC project (ANR-16-CE39-0006), by the "Investments for the future" Programme IdEx Bordeaux - CPU (ANR-10-IDEX-03-02), and by ERC Starting Grant ERC-2013-StG-335086-LATTAC.

D.A. 2-Party Decryption : zero-knowledge

Definition D − 7. An encryption scheme Π supporting 2-party decryption is *zero-knowledge for* A if there exists an efficient simulator $\mathsf{Sim}^{2d} = (\mathsf{Sim}^{2d}_{\mathsf{Share}}, \mathsf{Sim}^{2d}_{\mathrm{A}})$ which simulates the sharing phase and the player A.

The subroutine $\mathsf{Sim}^{2d}_{\mathsf{Share}}$ takes as input a public key pk and outputs (pk', sk'_{B}) that simulates the public key obtained from the Share algorithm and Bob's share of the secret key.

The subroutine $\mathsf{Sim}^{2d}_{\mathrm{A}}$ takes as input a public key pk a ciphertext c, a plaintext m, possibly sk_{B} and a flow flow. It emulates honest player A's answer upon receiving the flow flow when running the protocol $\mathsf{2Dec}(pk, c; sk_{\mathrm{A}}; sk_{\mathrm{B}})$ without sk_{A}, and forcing the output to be m.

Then, for all $\lambda \in \mathbf{N}$, for any $(\mathsf{params} \leftarrow \mathsf{Setup}(1^{\lambda})$, for any pair of keys $(pk, sk) \leftarrow \Pi.\mathsf{KeyGen}(1^{\lambda}, \mathsf{params})$, for any shares $(pk, sk_{\mathrm{A}}, sk_{\mathrm{B}}) \leftarrow \mathsf{Share}(pk, sk)$ or for any *simulated* share $(pk', sk'_{\mathrm{B}}) \leftarrow \mathsf{Sim}^{2d}_{\mathsf{Share}}(pk)$, and for any adversary $\mathscr{D}$ playing the role of B, the advantage

$$\mathsf{Adv}^{zk}_{\mathrm{A},\Pi}(\mathscr{D}) = \left|\Pr[1 \leftarrow \mathscr{D}^{\mathsf{A}}(pk, sk_{\mathrm{B}})] - \Pr[1 \leftarrow \mathscr{D}^{\mathsf{Sim}^{2d}_{\mathrm{A}}()}(pk', sk'_{\mathrm{B}})]\right|$$

is negligible.

We define similarly that Π is *zero-knowledge for* B. It is *zero-knowledge* if it is zero-knowledge for A and B.

D.B. Proof of Theorem D – 5

Theorem D – 5. *The ESP between* Π_+ *and* $\Pi_\times^0$ *whose routines are described in Fig. D.6 and D.7 is zero-knowledge if* Π_+ *and* $\Pi_\times$ *are two compatible encryptions that are IND-CPA and whose 2-party decryption are zero-knowledge, and* EZT *is zero-knowledge.*

Proof. Once again, the proof consists in proving that after a share of the secret keys, both switching procedures are zero-knowledge for Alice and Bob. As both switches consist in a sequence of protocols that have been independently proved secure, the main issue in the proof consists in showing that their sequential combination is still secure. The reduction will get a pair $(\mathsf{C}, \bar{\mathsf{C}})$ of input and output of the whole switches, and the main idea is to construct such intermediate pairs for each independent subroutines using random ciphertexts.

ZK for Alice. Let us start with the proof that the ESP is zero-knowledge for Alice. We describe a simulator Sim whose behavior is indistinguishable from Alice's behavior in front of an adversarial Bob.
$\mathsf{Sim}_{\mathsf{Share}}$: The simulator receives the public key $(pk_+, pk_\times)$ and sets $\mathsf{Sim}_{\mathsf{Share}}$ as follows: it calls out the $\mathsf{Sim}^{2d}_{\mathsf{Share}}$ procedures of the zero-knowledge property of Alice for 2-party decryption of respectively Π_+ and $\Pi_\times$ with pk_+ and $pk_\times$ as input. In particular it gets $sk'_{\mathsf{B}} = (x^+_{\mathsf{B}}, x^\times_{\mathsf{B}})$ it feeds the adversary with. When Sim is requested for a switch, it receives a pair of twin ciphertexts $(\mathsf{C}, \bar{\mathsf{C}})$.

Game G_0**.** This game is the real game. The simulator simulates all the secrets in an honest way and gives his share to Bob. It plays honestly any switching protocols on an input $(\mathsf{C}, \bar{\mathsf{C}})$ using Alice's secret key.

Game G_1**.** Each time Sim is requested for a switch ($\mathsf{Switch}_{\times\to+}$ or $\mathsf{Switch}_{+\to\times}$) it is given as input $(\mathsf{C}, \bar{\mathsf{C}})$ and one of the two is an encryption of m under $\Pi^0_\times$, which contains an Π_+-encryption under pk' of the bit b. The simulation uses its knowledge of the secret key sk' to decrypt the bit b. This game is indistinguishable from the previous one.

Game G_2**.** A modification is done for the *additive to multiplicative* case. The setup and key generation are the same as in the previous game. When requested to participate to a $\mathsf{Switch}_{+\to\times}$, with $(\mathsf{C}, \bar{\mathsf{C}})$ as input, the simulator uses its knowledge of b to query the EZT's simulator for Alice with $(\mathsf{C}, \Pi_+.\mathsf{Encrypt}(pk^+, b))$ as input. By definition of the simulator for the EZT, this game is indistinguishable from the previous one.

Game G_3**.** After the simulation of the EZT procedure, Alice and Bob gets C^+_b. The simulation now uses the ReEnc_+'s simulator for Alice with this C^+_b and the ciphertext $\Pi_+.\mathsf{Encrypt}(pk', b)$ as input, once again thanks to the knowledge of b. Thanks to the zero-knowledge property of ReEnc_+, this game is indistinguishable from the previous one.

Game G_4**.** Now the simulation uses the simulator for the $\mathsf{Switch}_{+\to\times}$. As the simulation knows $\bar{\mathrm{C}}$, it can extract its first component which is a $\Pi_\times$-encryption of $m+b$. Therefore, it calls $\mathsf{Switch}_{+\to\times}$'s simulator for Alice with C^+_{m+b} (obtained by genuinely computing the Hom_+ after the re-encryption) and $\bar{\mathrm{C}}$'s first component. Because we proved that the $\mathsf{Switch}_{+\to\times}$ procedure is zero-knowledge in Theorem D−3, this game is indistinguishable from the previous one.

Game G_5**.** The final flow from the switching protocols is simply the forward of $\bar{\mathrm{C}}$ since it is a twin ciphertext of C: this game is indistinguishable from the previous one.

Game G_6**.** The modification now concerns the ***multiplicative to additive*** case. The simulation has as input $(\mathrm{C},\bar{\mathrm{C}})$ where C is an encryption of a message m under $\Pi^0_\times$ and $\bar{\mathrm{C}}$ is a twin ciphertext. Sim still knows the bit b. To simulate the switch, it uses the corresponding simulator for Alice with, as input the first component of C which is an encryption using $\Pi_\times$ of $m+b$ and $\Pi_+.\mathsf{Hom}_+(pk^+,\bar{\mathrm{C}},\Pi_+.\mathsf{Encrypt}(pk^+,b))$ which is an encryption $m+b$ under Π_+. Because of the zero-knowledge property of this switch proved in Theorem D−3, this game is indistinguishable from the previous one.

Game G_7**.** The simulation now simulates the EZT procedure: it feeds the corresponding simulator with the second component of C (which is an encryption of a random element under Π_+) and $\Pi_+.\mathsf{Encrypt}(pk^+,\bar{b})$, which is a valid input. The EZT being zero-knowledge, this game is indistinguishable from the previous.

Game G_8**.** The last step of the switch in the multiplicative to additive direction is the computation of the Π_+ encryption of a product. The simulation makes a call to the simulator of the $\mathsf{2Mul}_+$ protocol with as input: the output of the first switch, the output of the EZT and $\bar{\mathrm{C}}$. As this is a genuine input, this game is indistinguishable from the previous.

Game G_9**.** From now on, the simulation will not use its knowledge of b and of the secret key sk'. To do so, in the additive to multiplicative direction, the simulation will feed the EZT simulator with (C,C'), where C' is a ciphertext of a random element in $\mathscr{R}$ under pk, instead of an encryption of b (see Game G_2.). Thanks to the IND-CPA property of Π_+, this game is indistinguishable from the previous one.

Game G_{10}**.** The simulation runs the simulator for the re-encryption process with C^+_b and a ciphertext of random element in $\mathscr{R}$ under pk', instead of an encryption of b, and again, because Π_+ is IND-CPA, this game is indistinguishable from the previous one.

Game G_{11}**.** In the multiplicative to additive direction, the simulator of the first ESP is run with the first component of C and a ciphertext of a random element in $\mathscr{R}^*$ under $pk^\times$. Since $\Pi_\times$ is IND-CPA, this game is indistinguishable from the previous one.

Game G_{12}**.** The simulation now runs the EZT simulator with the second component of C and a ciphertext of a random element of $\mathscr{R}$ instead of an encryption of $\bar{b}$. Because Π_+ is IND-CPA, this game is indistinguishable from the previous.

Game G_{13}**.** The simulation now uses the procedure $\mathsf{Sim}_{\mathsf{Share}}$ to simulates Bob's keys. By the zero-knowledge property of the 2-party decryption, this game is indistinguishable from the previous one and the adversary is in an environment completely simulated by Sim.

ZK for Bob. The proof that the protocols are zero-knowledge for Bob follows the same lines. It is a bit simpler since Bob has less contribution in the additive to multiplicative direction and the switch the other way around is essentially symmetric. □

D.C. Uniform Sampling in a Cyclic Group of Unknown Order with a Folded Discrete Gaussian Distribution

Let $\sigma > 0$ a real number and let us denote by ρ_σ the Gaussian centered function: $\rho_\sigma(x) = \exp(-\pi \|x\|^2/\sigma^2)$ for $x \in \mathbf{R}^n$. For a lattice Λ, we note $\rho_\sigma(\Lambda) = \sum_{x\in\Lambda} \rho_\sigma(x)$ and define the probability mass function $\mathscr{D}_{\Lambda,\sigma}$ over Λ by

$$\forall x \in \Lambda, \mathscr{D}_{\Lambda,\sigma}(x) = \rho_\sigma(x)/\rho_\sigma(\Lambda).$$

The smoothing parameter was defined by Micciancio and Regev [MR07]. For a lattice Λ, and a real $\epsilon > 0$, the smoothing parameter $\eta_\epsilon(\Lambda)$ is the smallest s such that $\rho_{1/s}(\Lambda^\star \setminus \{0\}) \leqslant \epsilon$.

From [MR07, Lemma 3.3], we have :

Fact 1

$$\eta_\epsilon(\Lambda) \leqslant \sqrt{\frac{\ln(2n(1+1/\epsilon))}{\pi}}\lambda_n(\Lambda).$$

The following result is implicit in [MR07] and made explicit in [GPV08, Corollary 2.8].

Fact 2 Let Λ, Λ' be n-dimensional lattices of same rank, with $\Lambda' \subseteq \Lambda$. Then for any $\epsilon \in \mathbf{R}, 0 < \epsilon < 1/2$, any $\sigma \geqslant \eta_\epsilon(\Lambda')$, $(\mathscr{D}_{\Lambda,\sigma} \mod \Lambda')$ is within statistical distance at most 2ϵ of the uniform distribution over $(\Lambda \mod \Lambda')$.

Lemma D − 1. *Let* G *be a cyclic group of order* q*, generated by* g*. Consider the random variable* X *with values in* G *with uniform distribution:* $\Pr[X = h] = \frac{1}{q}$ *for all* h *in* G*, and* Y *the random variable with values in* G *defined as follows. Draw* y *from* $\mathscr{D}_{\mathbf{Z},\sigma}$*, with* $\sigma \geqslant q\sqrt{\frac{\ln(2(1+1/\epsilon))}{\pi}}$*, and define* $Y = g^y$*. Then,* $\Delta(X, Y) \leqslant 2\epsilon$*.*

Proof. Let X′ the random variable with values in $\{0, \dots, q-1\}$ with uniform distribution and Y′ defined by $Y' = (y \mod q)$ where y is drawn from $\mathscr{D}_{\mathbf{Z},\sigma}$. Clearly, $\Delta(X, Y) = \Delta(X', Y')$.

We apply the Fact 1 with $n = 1$ and $\Lambda' = q\mathbf{Z}$. As $\lambda_n(q\mathbf{Z}) = q$, we get

$$\eta_\epsilon(q\mathbf{Z}) \leqslant q\sqrt{\frac{\ln(2(1+1/\epsilon))}{\pi}}.$$

Then we apply Fact 2 with $\Lambda = \mathbf{Z}$ and $\Lambda' = q\mathbf{Z}$. For any $\sigma \geqslant q\sqrt{\frac{\ln(2(1+1/\epsilon))}{\pi}}$, $\Delta(X', Y') \leqslant 2\epsilon$. □

D.D. Sharing Secret over the Integers

As one of the protocols involved in our construction needs a group of unknown order, sampling the secret key exponents and their shares is an issue. We have to use secret sharing of integers in a public interval. This problem has already been addressed, for example in [DM10, Section 4].

We adapt here this solution to fit our special case of sharing between two parties. Let $s \in \{-\mathrm{N}, \dots, \mathrm{N}\}$ be an integer to be shared. Let λ be a security parameter. Let s_A (resp. s_B) be the share of Alice (resp. of Bob). The idea is to take s_A uniform in a very large interval in order to make the distribution $s_\mathrm{B} = s - s_\mathrm{A}$ almost independent of s. More precisely, the share s_A is drawn uniformly in $\{-2^\lambda \mathrm{N}, \dots, 2^\lambda \mathrm{N}\}$ and $s_\mathrm{B} := s - s_\mathrm{A}$.

Let us show that the scheme is private. Let $s' \in \{-\mathrm{N}, \dots, \mathrm{N}\}$ be another secret shared as $(s'_\mathrm{A}, s'_\mathrm{B})$ with s'_A uniform in $\mathrm{I} := \{-2^\lambda \mathrm{N}, \dots, 2^\lambda \mathrm{N}\}$ and $s'_\mathrm{B} := s' - s'_\mathrm{A}$. The scheme is private if s_A (resp. s_B) is indistinguishable from s'_A (resp. s'_B). For s_A and s'_A it is clear. The share s_B and s'_B follow the uniform distribution on I respectively translated by $-s$ and $-s'$. We next show that the statistical distance $\Delta(s_\mathrm{B}, s'_\mathrm{B})$ between the random variables s_B and s'_B is negligible. Denote $p = 1/(2^{\lambda+1}\mathrm{N} + 1) = \Pr[s_\mathrm{A} = z]$ for $z \in \mathrm{I}$. The statistical distance $\Delta(s_\mathrm{B}, s'_\mathrm{B})$ will be maximal if $|s - s'|$ is maximal, for example, wlog, if $s = \mathrm{N}$ and $s' = -\mathrm{N}$. Let $\mathrm{J} = \{z \in \mathbf{Z}, \Pr[s_\mathrm{B} = z] > \Pr[s'_\mathrm{B} = z]\}$. It is clear that $\mathrm{J} = \{-2^\lambda \mathrm{N} - \mathrm{N}, \dots, -2^\lambda \mathrm{N} + \mathrm{N} - 1\}$ and that for $z \in \mathrm{J}$, $\Pr[s_\mathrm{B} = z] = p$ and $\Pr[s'_\mathrm{B} = z] = 0$. As a result as $\Delta(s_\mathrm{B}, s'_\mathrm{B}) = \sum_{z \in \mathrm{J}} \Pr[s_b = z] - \Pr[s'_\mathrm{B} = z]$, one has $\Delta(s_\mathrm{B}, s'_\mathrm{B}) = 2\mathrm{N}p < 2\mathrm{N}/(2^{\lambda+1}\mathrm{N}) = 2^{-\lambda}$ which is negligible.

In Appendix D.C, we saw that Discrete Gaussian distribution can improve uniform sampling in certain cases. However for sharing over the integer, there is no gain: suppose that s_A is taken from $\mathscr{D}_{\mathbf{Z},\sigma}$ for some σ and $s_\mathrm{B} = s - s_\mathrm{A}$. As before, we compute the statistical distance between two shares of different secrets s and s', and this still reduces to computing the distance between $\mathscr{D}_{\mathbf{Z},\sigma} - s$ and $\mathscr{D}_{\mathbf{Z},\sigma} - s'$. Let $t = |s' - s|$. This statistical distance is the same than between $\mathscr{D}_1 := \mathscr{D}_{\mathbf{Z},\sigma}$ and $\mathscr{D}_2 := \mathscr{D}_{\mathbf{Z},\sigma} + t$. Let $\mathrm{J} \subset \mathbf{Z}$ be the subset of integers z such that $\mathscr{D}_1(z) > \mathscr{D}_2(z)$. Equality occurs when $z^2 = (z - t)^2$, i.e., when $z = t/2$. So $\mathrm{J} = \{z < t/2\}$. As a result $\Delta(\mathrm{D}_1, \mathrm{D}_2) = \sum_{z<t/2} \mathscr{D}_1(z) - \mathscr{D}_2(z) = \sum_{-t/2 \leqslant z < t/2} \mathscr{D}_1(z) \leqslant t/\rho_\sigma(\mathbf{Z})$, where the equality comes with the cancellation of the terms, and the inequality by upper bounded all the terms by the value in 0. Then, as $\rho_\sigma(\mathbf{Z}) > \sigma$, we eventually find that $\Delta(\mathrm{D}_1, \mathrm{D}_2) < t/\sigma = |s' - s|/\sigma < 2\mathrm{N}/\sigma$.

As a result the standard deviation must be chosen as $\sigma = 2^{\lambda+1}\mathrm{N}$ in order to make that distance negligible, so we obtain something similar to what we saw with the uniform distance.

Chapter E

Practical Fully Secure Unrestricted Inner Product Functional Encryption modulo p

Joint work with Fabien Laguillaumie and Ida Tucker
[CLT18]

Abstract. Functional encryption is a modern public-key cryptographic primitive allowing an encryptor to finely control the information revealed to recipients from a given ciphertext. Abdalla, Bourse, De Caro, and Pointcheval (PKC 2015) were the first to consider functional encryption restricted to the class of linear functions, i.e. inner products. Though their schemes are only secure in the selective model, Agrawal, Libert, and Stehlé (CRYPTO 16) soon provided adaptively secure schemes for the same functionality. These constructions, which rely on standard assumptions such as the Decision Diffie-Hellman (DDH), the Learning-with-Errors (LWE), and Paillier's Decision Composite Residuosity (DCR) problems, do however suffer of various practical drawbacks. Namely, the DCR based scheme only computes inner products modulo an RSA integer which is oversized for many practical applications, while the computation of inner products modulo a prime p either requires, for their (DDH) based scheme, that the inner product be contained in a sufficiently small interval for decryption to be efficient, or, as in the LWE based scheme, suffers of poor efficiency due to impractical parameters.

In this paper, we provide adaptively secure functional encryption schemes for the inner product functionality which are both efficient and allow for the evaluation of unbounded inner products modulo a prime p. Our constructions rely on new natural cryptographic assumptions in a cyclic group containing a subgroup where the discrete logarithm (DL) problem is easy which extend Castagnos and Laguillaumie's assumption

(RSA 2015) of a DDH group with an easy DL subgroup. Instantiating our generic construction using class groups of imaginary quadratic fields gives rise to the most efficient functional encryption for inner products modulo an arbitrary large prime p. One of our schemes outperforms the DCR variant of Agrawal et al.'s protocols in terms of size of keys and ciphertexts by factors varying between 2 and 20 for a 112-bit security.

E.1. Introduction

Traditional public key encryption (PKE) provides an all-or-nothing approach to data access. This somewhat restricting property implies that a receiver can either recover the entire message with the appropriate secret key, or learns nothing about the encrypted message. In many real life applications however, the encryptor may wish for a more subtle encryption primitive, allowing him to disclose distinct and restricted information on the encrypted data according to the receivers privileges. For instance, consider a cloud-based email service where users may want the cloud to perform spam filtering on their encrypted emails but learn nothing more about the contents of these emails. Here the user only wants the cloud to learn one bit indicating whether or not the message is spam, but nothing more.

Functional encryption (FE) [BSW11, O'N10] emerged from a series of refinements of PKE, starting with identity based encryption [Sha84], which was later extended to fuzzy identity-based encryption by Sahai and Waters [SW05]. This work also introduced attribute-based encryption, where a message is encrypted for all users that have a certain set of attributes. FE encompasses all three of these primitives, and goes further still, as it allows not only to devise policies regulating which users can decrypt, but also provides control over which piece or function of the data each user can recover. Specifically, FE allows for a receiver to recover a function $f(y)$ of the encrypted message y, without learning anything else about y. The primitive requires a trusted authority, which possesses a master secret key msk, to deliver secret keys sk_{f_i} – associated to specific functionalities f_i – to the appropriate recipients. The encryptor computes a single ciphertext associated to the plaintext $c = \mathsf{Encrypt}(y)$, from which any user, given a decryption key sk_{f_i}, can recover $f_i(y) = \mathsf{Decrypt}(sk_i, c)$.

There exist two main security definitions for FE, indistinguishability-based and a stronger simulation-based security. The former – which is the model we adopt throughout this paper – requires that no efficient adversary, having chosen plaintext messages y_0 and y_1, can guess, given the encryption of one of these, which is the underlying message with probability significantly greater than $1/2$. The adversary can query a key derivation oracle for functionalities f, with the restriction that $f(y_0) = f(y_1)$, otherwise one could trivially tell apart both ciphertexts. Though constructions for general FE have been put forth, these schemes are far from practical, and only allow for the adversary to request an a priori bounded number of secret keys [GKP+13b, SS10], or rely on non-standard and ill-understood cryptographic assumptions such as indistinguishability obfuscation or multilinear maps [ABSV15, BGJS16, GKP+13a, GVW12, Wat15, GGHZ16].

The problem thus arose of building efficient FE schemes for restricted classes of functions; such constructions could be of great use for many practical applications, while developing our understanding of FE.

Inner Product Functional Encryption (IPFE).

The restriction of FE to linear functions, or equivalently to the inner product functionality yields many interesting applications. Among other uses, linear functions allow for the computation of weighted averages and sums which are of use for statistical analysis on encrypted data, where the statistical analysis itself has sensitive information. As mentioned by Katz, Sahai and Waters in [KSW08], another application is the evaluation of polynomials over encrypted data. Agrawal, Libert and Stehlé [ALS16, Section 6] motivate FE for the computation of linear functions modulo a prime p by demonstrating that such a scheme can be turned into a bounded collusion FE scheme for all circuits[1]. And as a final example, Agrawal, Bhattacherjee, Phan, Stehlé and Yamada provide a generic transformation from FE for linear functions to trace-and-revoke systems in [ABP+17]. Naturally as they are performing linear algebra, their transformation requires the modulus to be prime and preferably quite large (of the order of 128 or 256 bits).

The primitive can succinctly be defined as follows: plaintexts are vectors $\vec{y} \in \mathscr{R}^\ell$, where $\mathscr{R}$ is a given ring. Function specific secret keys $sk_{\vec{x}}$ are derived from vectors $\vec{x} \in \mathscr{R}^\ell$ and allow to recover $\langle \vec{y}, \vec{x} \rangle \in \mathscr{R}$ but reveal no further information about $\vec{y}$. It is worth noting that due to the linearity of inner products, if the adversary requests decryption keys derived from independent vectors $\vec{x}_i$ for $i \in \{1, \dots, \ell\}$, it can recover $\vec{y}$ by resolving a simple system of linear equations resulting from $\langle \vec{y}, \vec{x}_i \rangle$ for $i \in \{1, \dots, \ell\}$.

This specific line of research was initiated by Abdalla, Bourse, De Caro and Pointcheval in 2015 [ABDP15]. They provided the first IPFE schemes which rely on standard assumptions such as learning with errors (LWE) and decision Diffie Hellman (DDH). However their schemes are only secure in the *selective setting*, i.e. the adversary must commit to challenge messages before having access to the schemes' public parameters. Though of great theoretical interest, such schemes are not sufficiently secure for practical applications, indeed selective security is often considered a first step towards proving full *adaptive security*. The first fully secure schemes were put forth by Agrawal, Libert and Stehlé [ALS16] under the LWE, DDH and Paillier's Decision Composite Residuosity (DCR, *cf.* [Pai99]) assumptions. Abdalla *et al.* in [ABDP16] also put forth a generic construction achieving adaptive security and provide instantiations from the DDH, DCR and LWE assumptions. However, their instantiation from Elgamal gives the same construction as the DDH based scheme of [ALS16], and their obtained schemes from LWE are restricted to the computation of inner products over the integers $\mathbf{Z}$, and are less efficient than those of [ALS16]. Finally Benhamouda *et al.* [BBL17, Bou17] provided generic constructions from hash proof systems to both chosen plaintext and chosen ciphertext

[1] We note however that this application of linear FE modulo a prime p can not be instantiated with our schemes, as we require p to be at least a 112-bit prime, whereas this application typically calls for small values of p (e.g. $p = 2$).

secure IPFE schemes. The resulting schemes are again restricted to the computation of inner products over the integers $\mathbf{Z}$ and the sizes of secret keys are larger than those of [ALS16] (see details at the end of the introduction).

These brilliant developments do however still suffer of practical drawbacks. Namely the computation of inner products modulo a prime p are restricted, in that they require that the inner product $\langle \vec{y}, \vec{x} \rangle$ be small for decryption to be efficient (as is the case for the schemes of [ABDP15], [ABDP16], and the DDH based scheme of [ALS16]). To our knowledge, the only scheme that allows for decryption of inner products of any size modulo a prime p is the LWE based scheme of [ALS16], which suffers of poor efficiency since the modulus should be exponentially large in the dimension of encrypted vectors while the size of ciphertexts is cubic in this dimension.

Our Contributions.

In this paper we put forth IPFE schemes which resolve the aforementioned issue. Our constructions allow for inner products over the integers and modulo a prime integer p, and rely on novel cryptographic assumptions defined in Subsection E.3.1. These are variants of the [CL15] assumption, which supposes the existence of a DDH *group with an easy* DL *subgroup*: a cyclic group $G = \langle g \rangle$ where the DDH assumption holds together with a subgroup $F = \langle f \rangle$ of G where the discrete logarithm problem is easy. For ease of notation we will hereafter simply refer to this assumption as the DDH assumption.

The first assumption we introduce relies on a *hard subgroup membership* (HSM) problem (according to Gjøsteen's terminology [Gjø05]), in order to somewhat generalise Paillier's DCR assumption, which follows on a long line of assumptions of distinguishing powers in $\mathbf{Z}/N\mathbf{Z}$. Known attacks for these require computing the groups' order which reduces to factoring N. In the [CL15] framework, the group G is cyclic of order ps where s is unknown and $\gcd(p,s) = 1$. We denote $G^p = \langle g_p \rangle$ the subgroup of p−th powers in G. In this setting one has $G = F \times G^p$. The assumption is that it is hard to distinguish the elements of G^p in G.

We then define the DDH-f assumption, which is *weaker* than both the DDH assumption of [CL15], and the aforementioned HSM assumption. Denoting $\mathcal{D}$ a distribution statistically close to the uniform distribution modulo ps, this assumption states that it is hard to distinguish distributions $\{(g^x, g^y, g^{xy}), x, y \hookleftarrow \mathcal{D}\}$ (i.e. Diffie-Hellman triplets in G) and $\{(g^x, g^y, g^{xy} f^u), x, y \hookleftarrow \mathcal{D}, u \hookleftarrow \mathbf{Z}/p\mathbf{Z}\}$.

We prove that this assumption is actually *equivalent* to the semantic security of the generic CL homomorphic encryption scheme of [CL15], an Elgamal variant in G where the messages are encoded in the exponent in the subgroup F. In fact, the DDH-f assumption is better suited to mask an element of F, thus providing clearer proofs.

These new assumptions allow us to construct generic, linearly homomorphic encryption schemes over $\mathbf{Z}/p\mathbf{Z}$ which are semantically secure under chosen plaintext attacks (ind-cpa), which we call HSM-CL and Modified CL (*cf.* Section E.3.2). The reductions between their semantic security and the underlying assumptions are given in Fig. E.1, where A → B indicates that assumption B holds if assumption A holds, i.e. A is a stronger assumption than B.

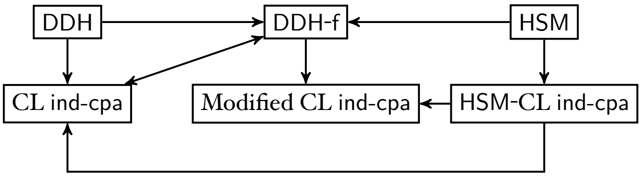


Figure E.1: Reductions between assumptions and ind-cpa security of CL variants

We then use the homomorphic properties of the above schemes to construct generic IPFE schemes over the integers and over $\mathbf{Z}/p\mathbf{Z}$, both from the weaker DDH-f assumption in Section E.4, and from the HSM assumption in Section E.5, somewhat generalising the scheme based on DCR of [ALS16]. Since the inner product is encoded in the exponent in the subgroup F, it can efficiently be recovered, whatever its size. We thereby present the first IPFE schemes which are both efficient and recover $\langle \vec{y}, \vec{x} \rangle \mod p$ whatever its size.

Our security proofs for the HSM based schemes follow a similar logic to those of [ALS16], analysing the entropy loss that occurs via queried keys, and demonstrating that there is enough residual entropy left for the challenge ciphertext to appear uniform to the adversary. However, significant difficulties occur for the schemes arising from the weaker DDH-f assumption. As in the DDH based scheme of [ALS16], we use a variant of Elgamal *à la* Cramer-Shoup. But unlike previous uses of this approach, the order of our group is unknown *and* may have small factors, so with constant probability an element may not be a generator. This calls for various subtleties: any element of the group can not be masked, however, if p is large enough, elements of the subgroup F of order p can be.

Moreover, in order to handle private key queries, instead of computing the global distribution of the keys given this information, we carefully simplify the description of the adversary's view, since merely restricting the adversary's view modulo p could potentially result in a loss of information.

We note that for our schemes over $\mathbf{Z}/p\mathbf{Z}$, vectors $\vec{x}_i$ from which keys are derived are in $\mathbf{Z}/p\mathbf{Z}$, whereas decryption keys are computed in $\mathbf{Z}$, so a lift of the $\vec{x}_i$ in $\mathbf{Z}$ must be done. Since lifting does not preserve linear dependencies, it is essential (as in [ALS16]) the key generation algorithm be stateful to lift vectors while maintaining linear dependencies. Without this restriction an adversary could learn a combination of the master key components which is singular modulo p but invertible over $\mathbf{Z}$, thus revealing the whole master key.

To instantiate our generic constructions we use class groups of imaginary quadratic fields. Although the devastating attack from [CL09] eliminates a whole family of protocols built from such groups, this attack applies to schemes whose security is based on factoring a discriminant while here this factorisation is public. Moreover [CL15] showed that designing with care discrete logarithm based cryptosystems within such groups is still possible and allows for efficient and versatile protocols (*Encryption switching protocols* for instance, *cf.* [CIL17]). The problem of computing a discrete logarithm in class groups of imaginary quadratic fields has been extensively studied since the 80's, and the complexity of best known subexponential algorithms is[2] $\mathcal{O}(\mathrm{L}_{1/2})$ (*cf.* [BJS10]) as opposed to $\mathcal{O}(\mathrm{L}_{1/3})$ (*cf.* [Adl94]) for the discrete logarithm problem in finite field or factoring. In particular this implies that our keys can be chosen shorter and corroborates the above claim that the assumptions on which we rely are indeed weak.

In terms of efficiency, we show in Section E.6 that for a security parameter of $\lambda = 112$ we outperform Paillier's variant of [ALS16] on all possible sizes by factors varying between 2 and 20.

Relation to Hash Proof Systems.

Hash proof systems (HPS) were introduced in [CS02] as a generalisation of the techniques used in [CS98]. Consider a set of words $\mathcal{X}$, an NP language $\mathcal{L} \subset \mathcal{X}$ such that $\mathcal{L} = \{x \in \mathcal{X} \mid w : (x, w) \in \mathsf{R}\}$ where R is the relation defining the language, $\mathcal{L}$ is the language of true statements in $\mathcal{X}$, and for $(x, w) \in \mathsf{R}$, w is a witness for $x \in \mathcal{L}$. A HPS defines a key generation algorithm KeyGen which outputs a secret hashing key hk and a public projection key hp such that hk defines a hash function $\mathscr{H}_{\mathsf{hk}} : \mathcal{X} \mapsto \Pi$, and hp allows for the (public) evaluation of the hash function on words $x \in \mathcal{L}$, i.e. $\mathscr{H}_{\mathsf{hp}}(x, w) = \mathscr{H}_{\mathsf{hk}}(x)$ for $(x, w) \in \mathsf{R}$. The *smoothness* property requires that for any $x \notin \mathcal{L}$, the value $\mathscr{H}_{\mathsf{hk}}(x)$ be uniformly distributed knowing hp.

The DDH and DCR assumptions can be used to instantiate smooth HPS's where the languages $\mathcal{L} \subset \mathcal{X}$ define hard subset membership problems. Such HPS's, endowed with homomorphic properties over the key space, underly the IPFE schemes of [ALS16]. In fact Benhamouda, Bourse, and Lipmaa in [BBL17], and Bourse, in his thesis [Bou17], present a generic construction from a key homomorphic HPS (with a few other required properties) to an IPFE scheme in $\mathbf{Z}$ which is secure under chosen plaintext attacks. They instantiate it from DDH and from DCR but leave out LWE due to the complexity of the resulting scheme, as simpler constructions can be attained without using HPSs.

We note that though our constructions resemble the above – one can deduce new subset membership problems from the assumptions in Subsection E.3.1 and associated HPS's – our proof techniques are very different to those of [Bou17]: so as to achieve adaptive security, their game challenger must *guess* the difference between challenge ciphertexts prior to generating the public/private key pair. If the hash key is *not* sampled uniformly at random from the key space (as in our constructions), then in order to maintain a level of security equivalent to that of the HPS the size of the secret keys increases substantially. Indeed, to encrypt vectors of dimension ℓ whose coordinates are bounded by Y, their proof techniques cause an additional $\ell\log(\mathrm{Y})$-bit term to appear in each coordinate of the secret key, whereas in our constructions over $\mathbf{Z}$, the bit length

[2] L_α is a shortcut to denote $\mathrm{L}_{\alpha,c}(x) = \exp(c \log(x)^\alpha \log(\log(x))^{1-\alpha})$

of the coordinates is independent of ℓ. As a consequence, this approach leads to less efficient schemes.

Our goal has been to build *practical* IPFE schemes, therefore we avoid this genericity and the key blow up it entails, carefully evaluating the information leaked to the adversary by the public key, the secret key queries and by the challenge ciphertext, thus demonstrating that the challenge bit β remains statistically hidden. This style of proof is closer to those of [ALS16], it allows us to obtain constructions for IPFE over $\mathbf{Z}$ that are substantially more efficient than those of [BBL17, Bou17], and constructions for IPFE modulo a prime p that do not restrict the size of the inputs or of the resulting inner product, which are the most efficient such schemes to date.

E.2. Background

Notations.

We denote sets by uppercase letters, vectors by bold lowercase letters, and the inner product of vectors $\vec{x}$ and $\vec{y}$ is denoted $\langle\vec{x},\vec{y}\rangle$. For a distribution $\mathscr{D}$, we write $d \hookleftarrow \mathscr{D}$ to refer to d being sampled from $\mathscr{D}$. We overload the notation as $b \hookleftarrow \mathrm{B}$ to say that b is sampled uniformally at random in the set B. For an integer x, we denote its size by $|x|$, and by $[x]$ the set of integers $\{1, \dots, x\}$. For any $\vec{c} \in \mathbf{R}^\ell$, real $\sigma > 0$, and ℓ-dimensional lattice Λ, $\mathscr{D}_{\Lambda,\sigma,\vec{c}}$ will denote the usual discrete Gaussian distribution over Λ.

Definition of Inner Product Functional Encryption.

This is a special case of functional encryption, as first formalised by Boneh, Sahai and Waters in [BSW11]. To start with, we provide the definition of a *functionality*.

Definition E – 1 (Functionality)**.** A functionality F defined over $(\mathscr{K}, \mathscr{Y})$ is a function $\mathrm{F} : \mathscr{K} \times \mathscr{Y} \to \Sigma \cup \{\perp\}$, where $\mathscr{K}$ is a key space, $\mathscr{Y}$ is a message space and Σ is an output space, which does not contain the special symbol $\perp$.

In this article, we consider the inner product functionality, which means that decrypting the encryption of a vector $\mathbf{y}$ with a key associated to a vector $\mathbf{x}$ will reveal only $\langle\mathbf{x},\mathbf{y}\rangle$. More precisely, we consider the function $\mathrm{F} : (\mathbf{Z}/p\mathbf{Z})^\ell \times (\mathbf{Z}/p\mathbf{Z})^\ell \to \mathbf{Z}/p\mathbf{Z} \cup \{\perp\}$ such that $\mathrm{F}(\mathbf{x},\mathbf{y}) = \langle\mathbf{x},\mathbf{y}\rangle$. The syntax of a functional encryption scheme is described below.

Definition E – 2 (Functional encryption scheme)**.** Let λ be a positive integer. A functional encryption scheme for a functionality F over $(\mathscr{K}, \mathscr{Y})$ is a tuple (Setup, KeyDer, Encrypt, Decrypt) of algorithms with the following specifications:

- Setup on input a security parameter 1^λ, outputs a master key pair (mpk, msk);
- KeyDer on input the master secret key msk and a key $\mathrm{K} \in \mathscr{K}$, outputs a key sk_K;
- Encrypt on input the master public key mpk and a message $\mathrm{Y} \in \mathscr{Y}$, outputs a ciphertext C;

- Decrypt takes as input the master public key mpk, a key sk_{K} and a ciphertext C and outputs $v \in \Sigma \cup \{\perp\}$.

For correctness, we require that for all $(mpk, msk) \leftarrow \mathsf{Setup}(1^\lambda)$, all keys $\mathrm{K} \in \mathscr{K}$ and all messages $\mathrm{Y} \in \mathscr{Y}$, if $sk_{\mathrm{K}} \leftarrow \mathsf{KeyDer}(msk, \mathrm{K})$ and $\mathrm{C} \leftarrow \mathsf{Encrypt}(mpk, \mathrm{Y})$, with overwhelming probability it holds that, if $v \leftarrow \mathsf{Decrypt}(mpk, sk_{\mathrm{K}}, \mathrm{C})$ then $v = \mathrm{F}(\mathrm{K}, \mathrm{Y})$ whenever $\mathrm{F}(\mathrm{K}, \mathrm{Y}) \neq \perp$.

Security.

We define below the security notion for functional encryption which states that given the ciphertext of a message Y, the only information obtained from the secret key sk_{K} is the evaluation of the function $f(\mathrm{K}, \mathrm{Y})$. More precisely, no adversary can distinguish an encryption of Y_0 from an encryption of Y_1 even with the knowledge of secret keys sk_{K} chosen adaptatively but satisfying $\mathrm{F}(\mathrm{K}, \mathrm{Y}_0) = \mathrm{F}(\mathrm{K}, \mathrm{Y}_1)$. The following definition is that of *adaptive* security, meaning that the adversary has access to the systems' public parameters, and can perform a series of secret key requests *before* choosing Y_0 and Y_1. We consider an indistinguishability-based definition instead of the simulation-based security definition of Boneh, Sahai and Waters from [BSW11]. This adaptive indistinguishability notion is easier to handle, and it is also the strongest adaptive notion of security that can be achieved for numerous interesting functionalities. In particular, it has been demonstrated in [BSW11, AGVW13, BO13] that the strong simulation-based definition cannot be met in the standard model, while O'Neill showed in [O'N10] that indistinguishability-based security is equivalent to non-adaptive simulation-based security for a class of functions that includes the inner product. Moreover, De Caro *et al.* [DIJ+13] describe a method to transform an FE achieving an indistinguishability-based security notion into an FE attaining a certain simulation-based security.

Definition E – 3 (Indistinguishability-based security)**.** A functional encryption scheme $\mathsf{FE} = (\mathsf{Setup}, \mathsf{KeyDer}, \mathsf{Encrypt}, \mathsf{Decrypt})$ provides semantic security under chosen plaintext attacks (ind-fe-cpa) if no PPT adversary $\mathscr{A}$ has non-negligible advantage denoted $\mathsf{Adv}_{\mathscr{A}}(\lambda)$, under the constraints that $\mathscr{A}$'s secret-key queries before and after its choice of challenge messages Y_0 and Y_1 satisfy $\mathrm{F}(\mathrm{K}, \mathrm{Y}_0) = \mathrm{F}(\mathrm{K}, \mathrm{Y}_1)$ for all K in the set of key queries. The advantage is defined as follows:

$$\begin{aligned}\mathsf{Adv}_{\mathscr{A}}(\lambda) = \Big|\Pr\Big[\beta = \beta' : mpk, msk \leftarrow \mathsf{Setup}(1^\lambda), \mathrm{Y}_0, \mathrm{Y}_1 \leftarrow \mathscr{A}^{\mathsf{KeyDer}(msk,\cdot)}(mpk),\\ \beta \xleftarrow{\$} \{0,1\}, \mathrm{C}^\star \leftarrow \mathsf{Encrypt}(mpk, \mathrm{Y}_\beta), \beta' \leftarrow \mathscr{A}^{\mathsf{KeyDer}(msk,\cdot)}(\mathrm{C}^\star)\Big] - \frac{1}{2}\Big|.\end{aligned}$$

Backgound on Lattices.

We here recall some definitions and basic results about Gaussian distributions. These are useful for our security proofs, in which we need to evaluate the distribution of an inner product when one of the two vectors follow a Gaussian distribution. We also recall an important result from [GPV08] which explains the conditions for a Gaussian distribution over a lattice which is reduced modulo a sublattice to be close to a uniform distribution, which is also a crucial point of our proofs.

Definition E – 4 (Gaussian Function)**.** For any $\sigma > 0$ define the Gaussian function on $\mathbf{R}^\ell$ centred at $\vec{c}$ with parameter σ:

$$\forall \vec{x} \in \mathbf{R}^\ell, \rho_{\sigma,\vec{c}}(\vec{x}) = \exp(-\pi\|\vec{x} - \vec{c}\|^2/\sigma^2).$$

If $\sigma = 1$ (resp. $\vec{c} = \vec{0}$), then the subscript σ (resp. $\vec{c}$) is omitted.

Definition E – 5 (Discrete Gaussians)**.** For any $\vec{c} \in \mathbf{R}^\ell$, real $\sigma > 0$, and ℓ-dimensional lattice Λ, define the discrete Gaussian distribution over Λ as:

$$\forall \vec{x} \in \Lambda, \quad \mathscr{D}_{\Lambda,\sigma,\vec{c}}(\vec{x}) = \rho_{\sigma,\vec{c}}(\vec{x})/\rho_{\sigma,\vec{c}}(\Lambda),$$

where $\rho_{\sigma,\vec{c}}(\Lambda) = \sum_{\vec{x}\in\Lambda} \rho_{\sigma,\vec{c}}(\vec{x})$.

Lemma E – 1. *Let $\vec{x} \in \mathbf{R}^\ell \setminus \{\vec{0}\}, \vec{c} \in \mathbf{R}^\ell, \sigma \in \mathbf{R}$ with $\sigma > 0$ and $\sigma' = \sigma/\|\vec{x}\|_2, c' = \frac{\langle \vec{c},\vec{x}\rangle}{\langle \vec{x},\vec{x}\rangle}$. A random variable K is distributed according to $\mathscr{D}_{\mathbf{Z},\sigma',c'}$ if and only if $V := K\vec{x}$ is distributed according to $\mathscr{D}_{\vec{x}\mathbf{Z},\sigma,\vec{c}}$.*

In dimension 1, Lemma E – 1 implies that if $x \in \mathbf{R}$, then $V = Kx$ is distributed according to $\mathscr{D}_{x\mathbf{Z},\sigma,c}$ if and only if K is distributed according to $\mathscr{D}_{\mathbf{Z},\sigma/|x|,c/x}$. The following lemma allows to evaluate the distribution of the inner product resulting from a constant vector $\vec{x}$, and a vector with coordinates sampled from a Gaussian distribution over the lattice $\vec{x} \cdot \mathbf{Z}$.

Lemma E – 2. *Let $\vec{x} \in \mathbf{R}^\ell$ with $\vec{x} \neq \vec{0}, \vec{c} \in \mathbf{R}^\ell, \sigma \in \mathbf{R}$ with $\sigma > 0$. Let V be a random variable distributed according to $\mathscr{D}_{\vec{x}\cdot\mathbf{Z},\sigma,\vec{c}}$. Then the random variable S defined as $S = \langle \vec{x}, V\rangle$ is distributed according to $\mathscr{D}_{\|\vec{x}\|_2^2\cdot\mathbf{Z},\sigma\cdot\|\vec{x}\|_2,\langle\vec{c},\vec{x}\rangle}$.*

Proofs of Lemmas E – 1 and E – 2 are provided in Aux. Material E.A.

Lemma E – 3 ([GPV08])**.** *Let $\Lambda'_0 \subset \Lambda_0 \subset \mathbf{R}^\ell$ be two lattices with the same dimension. Let $\epsilon \in (0, 1/2)$. Then for any $c \in \mathbf{R}^\ell$ and any $\sigma \geq \eta_\epsilon(\Lambda'_0)$, the distribution $D_{\Lambda_0,\sigma,c} \mod \Lambda'_0$ is within statistical distance 2ϵ from the uniform distribution over Λ_0/Λ'_0. The value $\eta_\epsilon(\Lambda'_0)$ is the* smoothing parameter *of the lattice Λ'_0, as defined in [MR04b].*

E.3. Variants of CL: assumptions and ind-cpa schemes

In [CL15], Castagnos and Laguillaumie introduced the framework of a DDH *group with an easy* DL *subgroup*: a cyclic group G where the DDH assumption holds together with a subgroup F of G where the discrete logarithm problem is easy. Within this framework, they designed a linearly homomorphic variant of Elgamal, described in Aux. Material E.B, and denoted CL throughout the rest of this paper. Moreover, they gave an instantiation using class groups of quadratic fields which allows the computation of linear operations modulo a prime p.

Their protocol is similar to the one of Bresson *et. al.* [BCP03] whose ind-cpa security relies on the DDH assumption in $(\mathbf{Z}/N^2\mathbf{Z})^\times$, where $N = pq$, using the arithmetic ideas of Paillier's encryption [Pai99]. Another encryption scheme based on Elgamal over $(\mathbf{Z}/N^2\mathbf{Z})^\times$ was proposed by Camenisch and Shoup in [CS03]. Its ind-cpa security relies on the Decision Composite Residuosity assumption (DCR), which consists in distinguishing the N−th powers in $(\mathbf{Z}/N^2\mathbf{Z})^\times$.

In the following subsection, we recall the framework of [CL15] and then generalise the DCR assumption to fit this framework of a DDH *group with an easy* DL *subgroup* with a hard subgroup membership problem (following [Gjø05]'s terminology). We also introduce a new DDH-like assumption which is weaker than DDH in G. Then, in Subsection E.3.2, we give generic encryption schemes whose ind-cpa security are based on these assumptions. In particular we give a generalisation of the scheme of [CS03] in a DDH *group with an easy* DL *subgroup*, and a modification of CL *à la* Cramer-Shoup. Finally, in Subsection E.3.3, we discuss the relations between these assumptions.

E.3.1. Algorithmic assumptions

To start with, we explicitly define the generator GenGroup used in the framework of a DDH group with an easy DL subgroup introduced in [CL15], with a few modifications as discussed below.

Definition E − 6 (Generator for a DDH group with an easy DL subgroup)**.** Let GenGroup be a pair of algorithms (Gen, Solve). The Gen algorithm is a group generator which takes as inputs two parameters λ and μ and outputs a tuple $(p, \tilde{s}, g, f, g_p, G, F, G^p)$. The set $(G, \cdot)$ is a cyclic group of order ps where s is an integer, p is a μ-bit prime, and $\gcd(p, s) = 1$. The algorithm Gen only outputs an upper bound $\tilde{s}$ of s. The set $G^p = \{x^p, x \in G\}$ is the subgroup of order s of G, and F is the subgroup of order p of G, so that $G = F \times G^p$. The algorithm Gen outputs f, g_p and $g = f \cdot g_p$ which are respective generators of F, G^p and G. Moreover, the DL problem is easy in F, which means that the Solve algorithm is a deterministic polynomial time algorithm that solves the discrete logarithm problem in F:

$$\Pr\Big[x = x^\star : (p, \tilde{s}, g, f, g_p, G, F, G^p) \leftarrow \mathsf{Gen}(1^\lambda, 1^\mu), x \hookleftarrow \mathbf{Z}_p, X = f^x,$$
$$x^\star \leftarrow \mathsf{Solve}(p, \tilde{s}, g, f, g_p, G, F, G^p, X)\Big] = 1.$$

Remark E − 1. In practice the size of s is chosen so that computing discrete logarithms in G^p takes time $\mathcal{O}(2^\lambda)$.

We note that this definition differs slightly from the original definition of [CL15]. First, we impose F to be of prime order p as our agenda is to use the instantiation with class groups of quadratic fields in order to have $\mathbf{Z}/p\mathbf{Z}$ as the message space. This means that the generic constructions do not encompass the schemes built from Paillier where the message space is $\mathbf{Z}/N\mathbf{Z}$, with $N = pq$. If it is possible to use $N = pq$ as the order of

F, the proofs have to rely on factoring assumptions to take care of the non-zero non-invertible elements of $\mathbf{Z}/N\mathbf{Z}$. As a consequence, this restriction simplifies the proofs, since an element of $\mathbf{Z}/p\mathbf{Z}$ is invertible if and only if it is non-zero.

Another modification is outputing the element g_p that generates G^p to define the HSM assumption below, and to set $g = f \cdot g_p$. In practice, the instantiation of [CL15] with class groups of quadratic fields already computes such an element g_p and thus defines the generator g of G. Note that this explicit definition of g is only needed for the proof of Theorem E−4 for the relation between the HSM and the DDH assumptions (*cf.* Def. E−7, E−8 and E−9 of Aux. Material E.B respectively). A last modification is that Gen only outputs an upper bound $\tilde{s}$ of s and not n. This is more accurate than the original definition as n is not used in the applications and actually, the instantiation does not compute n as it is a class number that requires sub-exponential time (with complexity $\mathcal{O}(L_{1/2})$) to be computed. This implies that in the following assumptions, exponents are sampled from distributions statistically close to uniform distributions. We discuss this in Remark E−2.

Now, following Gjøsteen's terminology ([Gjø05]) we define a *hard subgroup membership* (HSM) problem, in order to somehow generalise Paillier's DCR assumption: in Def. E−6, one has $G = F \times G^p$. The assumption is that it is hard to distinguish the elements of G^p in G.

Definition E − 7 (HSM assumption)**.** Let GenGroup = (Gen, Solve) be a generator for DDH groups with an easy DL subgroup. Using the notations introduced in Def E−6, the HSM assumption requires that the HSM problem is hard in G even with access to the Solve algorithm. Let $\mathcal{D}$ (resp. $\mathcal{D}_p$) be a distribution over the integers such that the distribution $\{g^x, x \hookleftarrow \mathcal{D}\}$ (resp. $\{g_p^x, x \hookleftarrow \mathcal{D}_p\}$) is at distance less than $2^{-\lambda}$ from the uniform distribution in G (resp. in G^p). Let $\mathcal{A}$ be an adversary for the HSM problem, its advantage is defined as:

$$\begin{aligned}\mathsf{Adv}_{\mathcal{A}}^{\mathsf{HSM}}(\lambda,\mu) = \Big| 2 \cdot \Pr\big[b = b^\star : (p, \tilde{s}, g, f, g_p, G, F, G^p) \leftarrow \mathsf{Gen}(1^\lambda, 1^\mu),& \\ x \hookleftarrow \mathcal{D}, x' \hookleftarrow \mathcal{D}_p, b \hookleftarrow \{0,1\}, Z_0 = g^x, Z_1 = g_p^{x'},& \\ b^\star \leftarrow \mathcal{A}(p, \tilde{s}, g, f, g_p, G, F, G^p, Z_b, \mathsf{Solve}(.))\big] - 1 \Big|&\end{aligned}$$

The HSM problem is said to be hard in G if for all probabilistic polynomial time attacker $\mathcal{A}$, $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{HSM}}(\lambda,\mu)$ is negligible.

Remark E − 2. In contrast to the traditional formulation of DDH or DCR, we can not sample uniformly elements in G^p or G either by a direct construction or using the generators and sampling uniformly exponents modulo the group order as the order s (resp. ps) of G^p (resp. of G) is unknown. As a result we use the upper bound $\tilde{s}$ of s in order to instantiate the distributions $\mathcal{D}_p$ and $\mathcal{D}$ of the above definitions. Choosing distributions $\mathcal{D}$ and $\mathcal{D}_p$ with induced distributions statistically close to the uniform distributions in G and G^p allows for more flexibility in our upcoming proofs, which is of interest, since it is easy to see that the DDH and HSM assumptions do not depend on the choice of the distribution.

In practice, we will instantiate $\mathcal{D}_p$ and $\mathcal{D}$ thanks to the following lemma, whose proof is in Aux. Material E.C. We use folded gaussians as they provide better efficiency than folded uniforms, and allow us to apply Lemma E−3 in our security proofs.

Lemma E − 4. *The distributions $\mathcal{D}_p$ and $\mathcal{D}$ can be implemented from the output of* Gen *as follows:*

1. *One can choose $\mathcal{D}$ to be the uniform distribution on $\{0, \dots, 2^{\lambda-2} \cdot \tilde{s} \cdot p\}$.*
2. *Alternatively, choosing $\mathcal{D} = \mathcal{D}_{\mathbf{Z},\sigma}$ with $\sigma = \tilde{s} \cdot p \cdot \sqrt{\lambda}$ allows for more efficient constructions as the sampled elements will tend to be smaller.*
3. *Likewise, one can choose $\mathcal{D}_p = \mathcal{D}_{\mathbf{Z},\sigma'}$ with $\sigma' = \tilde{s} \cdot \sqrt{\lambda}$*
4. *One can also, less efficiently, define $\mathcal{D}_p = \mathcal{D}$.*
5. *Conversely, one can also define $\mathcal{D}$ from $\mathcal{D}_p$ and the uniform distribution modulo p: the distribution $\{g_p^x \cdot f^a, x \hookleftarrow \mathcal{D}_p, a \hookleftarrow \mathbf{Z}_p\}$ is statistically close to the uniform distribution in G.*

Finally, we introduce a new assumption called DDH-f that we prove to be weaker than DDH. The security of our first IPFE relies on this assumption. Roughly speaking, it means that it is hard to distinguish the distributions

$$\{(g^x, g^y, g^{xy}), x, y \hookleftarrow \mathcal{D}\} \text{ and } \{(g^x, g^y, g^{xy} f^u), x, y \hookleftarrow \mathcal{D}, u \hookleftarrow \mathbf{Z}/p\mathbf{Z}\}.$$

In other words, as $g = f \cdot g_p$, we have on the left, a Diffie-Hellman triplet in G, and on the right, a triplet whose components in G^p form a Diffie-Hellman triplet, and whose components in F form a random triplet: (f^x, f^y, f^{xy+u}) (as noted in the previous remark, $\mathcal{D}$ induces distributions statistically close to the uniform in G^p and F).

We will see in the next subsection that the security of the CL encryption scheme is actually *equivalent* to this assumption and that this assumption is *weaker* than the DDH assumption and the HSM assumption (see Theorem E−4). As a side effect, using this assumption will simplify the forthcoming proofs as it is tightly related to the ind-cpa security of the underlying encryption scheme.

We note that DDH-f can be seen as an instance of the Extended-DDH (EDDH) problem as defined by Hemenway and Ostrovsky in [HO12]. They demonstrate that QR and DCR imply two different instantiations of EDDH, our implication from HSM to DDH-f somewhat generalises their proof since DDH-f is a more generic assumption than either of the hardness assumptions obtained from their reductions.

Definition E − 8 (DDH-f assumption)**.** Let GenGroup = (Gen, Solve) be a generator for DDH groups with an easy DL subgroup. Using the notations of Def E−6, the

DDH-f assumption requires that the DDH-f problem is hard in G even with access to the Solve algorithm. Let $\mathcal{D}$ be a distribution over the integers such that the distribution $\{g^x, x \hookleftarrow \mathcal{D}\}$ is at distance less than $2^{-\lambda}$ from the uniform distribution in G. Let $\mathcal{A}$ be an adversary for the DDH-f problem, its advantage is defined as:

$$\begin{aligned}\mathsf{Adv}_{\mathcal{A}}^{\mathsf{DDH\text{-}f}}(\lambda,\mu) = \Big|2\cdot \Pr\Big[b = b^\star : (p,\tilde{s},g,f,g_p,G,F,G^p) \leftarrow \mathsf{Gen}(1^\lambda,1^\mu),\\ x,y \hookleftarrow \mathcal{D}, u \hookleftarrow \mathbf{Z}/p\mathbf{Z}, X = g^x, Y = g^y, b \hookleftarrow \{0,1\}, Z_0 = g^{xy}, Z_1 = g^{xy}f^u,\\ b^\star \leftarrow \mathcal{A}(p,\tilde{s},g,f,g_p,G,F,G^p,X,Y,Z_b,\mathsf{Solve}(.))\Big] - 1\Big|.\end{aligned}$$

The DDH-f problem is said to be hard in G if for all probabilistic polynomial time attacker $\mathcal{A}$, $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{DDH\text{-}f}}\lambda,\mu)$ is negligible.

E.3.2. Some variants of the CL generic encryption scheme

The original Castagnos-Laguillaumie encryption scheme.

Castagnos and Laguillaumie put forth in [CL15] a generic construction for a linearly homomorphic encryption scheme over $\mathbf{Z}/p\mathbf{Z}$ based on a cyclic group with a subgroup of order p where the DL problem is easy, as given by the GenGroup generator of Def. E−6. Its description is provided in Fig. E.7 of Aux. Material E.B. They prove that this scheme is ind-cpa under the DDH assumption as defined in Def. E−9 of Aux. Material E.B. We demonstrate below that we can be more precise and prove that the security of this scheme is equivalent to the DDH-f assumption of Def. E−8: the key idea is to perform a one-time pad in F, instead of in the whole group G.

Theorem E−1. *The* CL *encryption scheme is semantically secure under chosen plaintext attacks* (ind-cpa) *if and only if the* DDH-f *assumption holds.*

Proof (sketch). Suppose that the DDH-f assumption holds. Let us consider the ind-cpa game, with a public key, $h = g^x$, $x \hookleftarrow \mathcal{D}$, and a challenge ciphertext $(c_1,c_2) = (g^r, f^{m_\beta}h^r)$ with $r \hookleftarrow \mathcal{D}$ and $\beta \hookleftarrow \{0,1\}$, $m_0,m_1 \in \mathbf{Z}/p\mathbf{Z}$. We can replace $(h,c_1,h^r) = (g^x,g^r,g^{xr})$ by $(g^x,g^r,g^{xr}f^u) = (g^x,g^r,h^rf^u)$ with $u \hookleftarrow \mathbf{Z}/p\mathbf{Z}$. As a result $c_2 = h^rf^{u+m_\beta}$. For the adversary, the value of r modulo n is fixed by $c_1 = g^r$ as g is a generator, so the value of h^r is fixed. As a result from c_2 an unbounded adversary can infer $u + m_\beta \in \mathbf{Z}/p\mathbf{Z}$ but as u is uniformly distributed in $\mathbf{Z}/p\mathbf{Z}$, he will have no information on β.

Conversely, we construct an ind-cpa adversary from a distinguisher for the DDH-f problem. Choose $m_0 \in \mathbf{Z}_p$ and $m_1 := m_0 + u$ with $u \hookleftarrow \mathbf{Z}/p\mathbf{Z}$. From the public key and the challenge ciphertext, construct the triplet

$$(h, c_1, c_2/f^{m_0}) = (g^x, g^r, g^{xr}f^{m_\beta - m_0}).$$

This gives a DH triplet if and only $\beta = 0$ and the exponent of f is uniformly distributed in $\mathbf{Z}/p\mathbf{Z}$ if and only $\beta = 1$. As a result, one can use the output of a distinguisher for the DDH-f problem to win the ind-cpa game. □

A linearly homomorphic encryption scheme from HSM.

As noted in the introduction of this section, the CL scheme was inspired by the scheme of [BCP03]. We here present a slight modification to this scheme so that it relies on the HSM assumption of Def. E−7 in order to somewhat generalise the approach of the Camenisch and Shoup scheme of [CS03]. This ind-cpa scheme will be the basis of the functional encryption scheme for inner product of Section E.5.

Setting the parameters.

We use the output $(p, \tilde{s}, g, f, g_p, \mathrm{G}, \mathrm{F}, \mathrm{G}^p)$ of the GenGroup generator of Def. E−6. We ignore the generator g (which is useless here). Following Lemma E−4, Item 3, we require $\sigma' > \tilde{s}\sqrt{\lambda}$ to ensure that $\{g_p^r, r \hookleftarrow \mathscr{D}_{\mathbf{Z},\sigma'}\}$ is at distance less than $2^{-\lambda}$ from the uniform distribution in G^p. The plaintext space is $\mathbf{Z}/p\mathbf{Z}$, where p is a μ bit prime, with $\mu \geq \lambda$. The scheme is depicted in Fig. E.2(a) and the standard proof of the following theorem is provided in Aux. Material E.D for completeness.

Theorem E − 2. *The scheme described in Fig. E.2(a) is semantically secure under chosen plaintext attacks (*ind-cpa*) under the* HSM *assumption.*

Enhanced variant of the CL encryption scheme.

We here put forth an enhanced version of the CL homomorphic encryption scheme. We modify the original CL scheme by adding a key *à la* Cramer-Shoup (*cf.* [CS98]). The security of this scheme also relies on the DDH-f assumption. This ind-cpa scheme will be the basis of the functional encryption scheme for inner product of Section E.4.

This modification to the CL encryption scheme incurs some challenges: let us consider the vanilla Elgamal scheme defined over a cyclic group of prime order q, generated by an element g. The modification leading to the Cramer-Shoup encryption scheme uses a second generator h and creates a key $\eta = g^x h^y$ where $x, y \hookleftarrow \mathbf{Z}/q\mathbf{Z}$. Then η^r, with $r \hookleftarrow \mathbf{Z}/q\mathbf{Z}$ is used to mask the plaintext message. In the proof under the DDH assumption, one replaces the DH triplet (h, g^r, h^r) built from the public key and the ciphertext by a random triplet and proves that the mask η^r is then uniformly distributed and acts as a one-time pad for the plaintext, even with the knowledge of η. However, the triplet (h, g^r, h^r) is indeed a DH triplet, because if h is a generator, $h = g^\alpha$ with $\alpha \in (\mathbf{Z}/q\mathbf{Z})^*$. As a result, α is almost uniformly distributed in $\mathbf{Z}/q\mathbf{Z}$ (an element $\alpha \hookleftarrow \mathbf{Z}/q\mathbf{Z}$ is such that $\alpha \neq 0$ with overwhelming probability if q is large). The same happens when considering a composite group order N' where N' is an RSA integer as in [Luc02], for instance, under the factoring assumption.

In our case, we use the GenGroup generator of Def. E−6, *i.e.*, a cyclic group G of composite order $n = ps$ generated by g, where s is unknown and may have some small factors. As a result, a random element $h = g^\alpha$, with $\alpha \hookleftarrow \mathscr{D}_{\mathbf{Z},\sigma}$ may not be a generator with constant probability. Consequently, the padding η^r where $r \hookleftarrow \mathscr{D}_{\mathbf{Z},\sigma}$ and $\eta = g^x h^y$, with $x, y \hookleftarrow \mathscr{D}_{\mathbf{Z},\sigma}$ may not be uniformly distributed in G knowing η. However, we can still adapt the proof: we only need η^r to act as a one-time pad in the subgroup $\mathrm{F} = \langle f \rangle$

```
Algorithm KeyGen($1^\lambda, 1^\mu$)
  1. $(p, \tilde{s}, f, g_p, G, F, G^p) \leftarrow \mathsf{Gen}(1^\lambda, 1^\mu)$
  2. Pick $x \hookleftarrow \mathscr{D}_{\mathbf{Z},\sigma'}$ and $h = g_p^x$
  3. Set $pk = (\tilde{s}, g_p, f, p, h)$
  4. Set $sk = x$
  5. Return $(pk, sk)$

Algorithm Encrypt($pk, m$)
  1. Pick $r \hookleftarrow \mathscr{D}_{\mathbf{Z},\sigma'}$
  2. Return $(g_p^r, f^m h^r)$

Algorithm Decrypt($sk, (c_1, c_2)$)
  1. Compute $M = c_2 / c_1^x$
  2. Return Solve(M)
```

(a) HSM-CL

```
Algorithm KeyGen($1^\lambda, 1^\mu$)
  1. $(p, \tilde{s}, g, f, G, F) \leftarrow \mathsf{Gen}(1^\lambda, 1^\mu)$
  2. Pick $x, y, \alpha \hookleftarrow \mathscr{D}_{\mathbf{Z},\sigma}$
  3. Compute $h = g^\alpha$
  4. Compute $\eta = g^x h^y$
  5. Set $pk = (g, h, \eta)$
  6. Set $sk = (x, y)$
  7. Return $(pk, sk)$

Algorithm Encrypt($pk, m$)
  1. Pick $r \hookleftarrow \mathscr{D}_{\mathbf{Z},\sigma}$
  2. Return $(g^r, h^r, \eta^r f^m)$

Algorithm Decrypt($sk, (c_1, c_2, c_3)$)
  1. Compute $M = c_3 / (c_1^x c_2^y)$
  2. Return Solve(M)
```

(b) Modified CL

Figure E.2: Description of our variants of the CL encryption

of G of order p, since our plaintext message $m \in \mathbf{Z}/p\mathbf{Z}$ is encoded as $f^m \in F$. Supposing that p is a μ-bit prime, with $\mu \geq \lambda$ is sufficient to prove this. As the exponents are taken close to uniform modulo n and $n = p \cdot s$ with $\gcd(p, s) = 1$, they behave independently and close to uniform modulo p and modulo s. As we are interested only in what happens modulo p, we can ignore the behavior modulo s and get ind-cpa security under the DDH-f assumption. Note that the use of this assumption instead of the stronger DDH assumption greatly simplifies the proof.

Setting the parameters.

We use the output $(p, \tilde{s}, g, f, g_p, G, F, G^p)$ of the generator GenGroup of Def. E−6. As in the original CL scheme (*cf.* Aux. Material E.B), we ignore the group G^p and its generator (which are useless here). Following Lemma E−4, Item 2, we require $\sigma > p\tilde{s}\sqrt{\lambda}$ to ensure that $\{g^r, r \hookleftarrow \mathscr{D}_{\mathbf{Z},\sigma}\}$ is at distance less than $2^{-\lambda}$ from the uniform distribution in G. The plaintext space is $\mathbf{Z}/p\mathbf{Z}$, where p is a μ bit prime, with $\mu \geq \lambda$. The scheme is depicted in Fig. E.2(b).

Theorem E − 3. *The scheme described in Fig. E.2(b) is semantically secure under chosen plaintext attacks (*ind-cpa*) under the* DDH-f *assumption.*

The proof is provided in Aux. Material E.E for completeness.

E.3.3. Relations between the assumptions

One can establish direct reductions from the underlying problems of the DDH, DDH-f and HSM assumptions. However it is somewhat easier to use intermediate results on the ind-cpa security of the schemes defined in the previous subsection to demonstrate these reductions.

We proved in Theorem E − 1 that the original CL cryptosystem is ind-cpa if and only if the DDH-f assumption holds. In [CL15], it was proven that this scheme is ind-cpa under the DDH assumption. As a result, DDH-f is a weaker assumption than DDH. Furthermore, it is easy to see that if the HSM scheme of Fig. E.2(a) is ind-cpa then the original CL cryptosystem is ind-cpa: from a public key $h = g_p^x$, $x \hookleftarrow \mathscr{D}_{\mathbf{Z},\sigma'}$ and a ciphertext $c = (c_1, c_2) = (g_p^r, f^m \cdot h^r)$, $r \hookleftarrow \mathscr{D}_{\mathbf{Z},\sigma'}$ for the HSM scheme, one can chose $a, b \hookleftarrow \mathbf{Z}/p\mathbf{Z}$ and construct $h' = h \cdot f^a$, and the ciphertext $c' = (c_1', c_2') = (c_1 \cdot f^b, c_2 \cdot f^{ab})$. According to Lemma E − 4, Item 5, h' and c_1' are statistically indistinguishable from the uniform distribution in G. Furthermore, $h' = g_p^x f^a = g^\alpha$ where α is defined modulo n from the Chinese remainder theorem, such that $\alpha \equiv x \pmod s$ and $\alpha \equiv a \pmod p$. Likewise, $c_1' = g_p^r f^b = g^\beta$ for some β defined equivalently. Finally, one has $c_2'/f^m = g_p^{xr} f^{ab} = g_p^{\alpha\beta \bmod s} f^{\alpha\beta \bmod p} = g^{\alpha\beta}$. As a result, $(h', c_1', c_2'/f^m)$ is a DH triplet in G, so h', c' are a public key and a ciphertext for m for the CL cryptosystem. As a result, an ind-cpa attacker against the cryptosystem based on HSM can be built from an ind-cpa attacker against CL. Now, if the HSM assumption holds, from Theorem E − 2, the HSM scheme is ind-cpa, so the CL scheme is also ind-cpa and the DDH-f assumption holds.

We can sum up these results with the following theorem (see also Fig. E.1).

Theorem E − 4. *The* DDH *assumption implies the* DDH-f *assumption. Furthermore, the* HSM *assumption implies the* DDH-f *assumption.*

E.4. Inner product FE relying on the DDH-f assumption

In this section, we build an IPFE scheme from the DDH-f assumption (*cf.* Def. E − 8). As proven in Theorem E − 4, this assumption is weaker than both the DDH and the HSM assumptions and yields simple proofs as it is suited to deal with the encoding of the message into a subgroup of prime order p. We use the formalism of a cyclic group with an easy DL subgroup. Our approach is based on the enhanced variant of the CL scheme, described in Fig. E.2(b). The resulting scheme over $\mathbf{Z}/p\mathbf{Z}$ can be viewed as an adaptation of the DDH scheme of [ALS16] to this setting, thereby removing the restriction on the size of the inner product.

The proof technique somewhat differs from the general approach of [ALS16]. We start from the ind-cpa proof of the enhanced variant of CL and then deal with the information provided by the key queries. Instead of computing the global distribution of the keys given this information, in order to make the proof go through, we have to carefully simplify the description of the adversary's view. A technical point is that even if we are only interested in what happens modulo p, as the plaintexts are defined in $(\mathbf{Z}/p\mathbf{Z})^\ell$, we cannot restrict the adversary's view modulo p: this could potentially result in a loss of information, as the key queries are defined in $\mathbf{Z}$.

We first present an FE scheme for inner products over $\mathbf{Z}$ (Section E.4.1) and then consider a scheme for inner products over $\mathbf{Z}/p\mathbf{Z}$ (Section E.4.2).

E.4.1. DDH-f-based FE for inner product over $\mathbf{Z}$

Setting the parameters.

As in the ind-cpa scheme of Fig. E.2(b), we use the output $(p, \tilde{s}, g, f, g_p, \mathrm{G}, \mathrm{F}, \mathrm{G}^p)$ of the GenGroup generator of Def. E−6. We ignore the group G^p and its generator g_p (which are useless here). We require that p is a μ-bit prime, with $\mu \geq \lambda$.

From Lemma E−4, Item 2, choosing $\sigma > \tilde{s} \cdot p \cdot \sqrt{\lambda}$ suffices to ensure that the distribution $\{g^x, x \leftarrow \mathscr{D}_{\mathbf{Z},\sigma}\}$ is at distance less than $2^{-\lambda}$ from the uniform distribution in G, however for security we must take a larger $\sigma > \tilde{s} \cdot p^{3/2} \cdot \sqrt{2\lambda}$ (*cf.* proof of Theorem E−5). The Encrypt algorithm operates on plaintext messages $\vec{y} \in \mathbf{Z}^\ell$ and the key derivation algorithm derives keys from vectors $\mathbf{x} \in \mathbf{Z}^\ell$. Norm bounds X and Y are chosen such that $\mathrm{X}, \mathrm{Y} < (p/2\ell)^{1/2}$ so as to ensure decryption correctness. Indeed key vectors $\mathbf{x}$ and message vectors $\mathbf{y}$ are assumed to be of bounded norm $\|\mathbf{x}\|_\infty \leq \mathrm{X}$ and $\|\mathbf{y}\|_\infty \leq \mathrm{Y}$, respectively. The decryption algorithm recovers $\langle \mathbf{x}, \mathbf{y} \rangle \mod p$ (using a centered modulus), which is exactly $\langle \mathbf{x}, \mathbf{y} \rangle$ over the integers, thanks to the Cauchy–Schwarz inequality and the norm bounds, since $\mathrm{X} \cdot \mathrm{Y} < p/2\ell$.

Construction.

Fig. E.3 depicts the functional encryption scheme for inner products in $\mathbf{Z}$ which is secure under the DDH-f assumption (*cf.* Def. E−8).

Correctness.

We have

$$\begin{aligned}\prod_{i\in[\ell]} \mathrm{E}_i^{x_i}/(\mathrm{C}^{s_{\mathbf{x}}} \cdot \mathrm{D}^{t_{\mathbf{x}}}) &= \prod_{i\in[\ell]} (f^{y_i}(g^{s_i} \cdot h^{t_i})^r)^{x_i}/((g^r)^{\langle \mathbf{x},\mathbf{s}\rangle} \cdot (h^r)^{\langle \mathbf{x},\mathbf{t}\rangle})\\ &= (f^{\sum_{i=1}^{\ell} y_i x_i})(g^{r\sum_{i=1}^{\ell} s_i x_i})(h^{r\sum_{i=1}^{\ell} t_i x_i})/(g^{r\langle \mathbf{x},\mathbf{s}\rangle} \cdot h^{r\langle \mathbf{x},\mathbf{t}\rangle})\\ &= f^{\langle x,y\rangle}.\end{aligned}$$

Algorithm $\mathsf{Setup}(1^\lambda, 1^\mu, 1^\ell)$

```
1. $(p, \tilde{s}, g, f, G, F) \leftarrow \mathsf{Gen}(1^\lambda, 1^\mu)$
2. Pick $\alpha \hookleftarrow \mathcal{D}_{\mathbf{Z},\sigma}$
3. Compute $h = g^\alpha$
4. Pick $\vec{s}, \vec{t} \hookleftarrow \mathcal{D}_{\mathbf{Z}^\ell,\sigma}$
5. For $1 \leq i \leq \ell$ :
6.     Compute $h_i = g^{s_i} h^{t_i}$
7. Return $msk = (\vec{s}, \vec{t})$
   and $mpk = (\tilde{s}, g, h, f, p, \{h_i\}_{i\in[\ell]})$
```

Algorithm $\mathsf{KeyDer}(msk, \vec{x})$

```
1. Compute in $\mathbf{Z}$:
   $sk_{\mathbf{x}} = (s_{\mathbf{x}}, t_{\mathbf{x}}) = (\langle \mathbf{x}, \mathbf{s} \rangle, \langle \mathbf{x}, \mathbf{t} \rangle)$
2. Return $sk_{\vec{x}} = (s_{\mathbf{x}}, t_{\mathbf{x}})$
```

Algorithm $\mathsf{Encrypt}(mpk, \vec{y})$

```
1. Pick $r \hookleftarrow \mathcal{D}_{\mathbf{Z},\sigma}$
2. Set $C = g^r$ and $D = h^r$
3. For $1 \leq i \leq \ell$ :
4.     Compute $E_i = f^{y_i} h_i^r$
5. Return $C_{\mathbf{y}} = (C, D, \{E_i\}_{i\in[\ell]})$
```

Algorithm $\mathsf{Decrypt}(mpk, C_{\mathbf{y}}, sk_{\vec{x}})$

```
1. Set $C_{\mathbf{x}} = (\prod_{i\in[\ell]} E_i^{x_i}) / (C^{s_{\mathbf{x}}} \cdot D^{t_{\mathbf{x}}})$
2. $\mathsf{sol} = \mathsf{Solve}(C_{\mathbf{x}})$
3. If $\mathsf{sol} \geq p/2$ :
4.     Return $(\mathsf{sol} - p)$
5. Else:
6.     Return $\mathsf{sol}$
```

Figure E.3: FE scheme for inner product over $\mathbf{Z}$ under the DDH-f assumption.

Applying the Solve algorithm to $C_{\mathbf{x}}$ yields $\langle \vec{x}, \vec{y} \rangle \mod p$, which, thanks to the norm bounds, is either $\langle \mathbf{x}, \mathbf{y} \rangle$ or $\langle \mathbf{x}, \mathbf{y} \rangle + p$. Since the absolute value of $\langle \mathbf{x}, \mathbf{y} \rangle$ is lower than $p/2$, if $\mathsf{sol} < p/2$ then $\langle \mathbf{x}, \mathbf{y} \rangle = \mathsf{sol}$ in $\mathbf{Z}$, otherwise $\langle \mathbf{x}, \mathbf{y} \rangle = (\mathsf{sol} - p)$.

Theorem E − 5. *Under the* DDH-f *assumption, the functional encryption scheme for inner products over* $\mathbf{Z}$ *of Fig. E.3 provides full security* (ind-fe-cpa).

Proof. The proof proceeds as a sequence of games, starting in Game 0 with the real ind-fe-cpa game and ending in a game where the ciphertext statistically hides the random bit β chosen by the challenger from the adversary's point of view. The beginning of the proof is similar to the proof of Theorem E − 3 on ind-cpa security. Then we take into account the fact that the adversary $\mathcal{A}$ has access to a key derivation oracle. For each Game i, we denote S_i the event $\beta = \beta'$.
Game 0 ⇒ Game 1: In Game 1 the challenger, who has access to the master secret key *msk*, computes the ciphertext using *msk* instead of *mpk*. The resulting ciphertext has exactly the same distribution therefore:

$$\Pr[S_0] = \Pr[S_1].$$

Game 1 ⇒ Game 2: In Game 1, the tuple $(h = g^\alpha, C = g^r, D = h^r = g^{\alpha r})$, where

Game 1

```
1. $mpk, msk \leftarrow \mathsf{Setup}(1^\lambda, 1^\mu, 1^\ell)$
2. $\mathbf{y}_0, \mathbf{y}_1 \leftarrow \mathscr{A}^{\mathsf{KeyDer}(msk,\cdot)}(mpk)$
3. Pick $\beta \hookleftarrow \{0,1\}$
4. Pick $r \hookleftarrow \mathscr{D}_{\mathbf{Z},\sigma}$
5. Compute $\mathrm{C} = g^r, \mathrm{D} = h^r$
6. For $1 \leq i \leq \ell$:
7.     Compute $\mathrm{E}_i = f^{y_{\beta,i}}\mathrm{C}^{s_i}\mathrm{D}^{t_i}$
8. $\mathrm{C}_{\mathbf{y}} = (\mathrm{C}, \mathrm{D}, \{\mathrm{E}_i\}_{i\in[\ell]})$
9. $\beta' \leftarrow \mathscr{A}^{\mathsf{KeyDer}(msk,\cdot)}(\mathrm{C}_{\mathbf{y}})$
10. Return $(\beta = \beta')$
```

Game 2

```
1. $mpk, msk \leftarrow \mathsf{Setup}(1^\lambda, 1^\mu, 1^\ell)$
2. $\mathbf{y}_0, \mathbf{y}_1 \leftarrow \mathscr{A}^{\mathsf{KeyDer}(msk,\cdot)}(mpk)$
3. Pick $\beta \hookleftarrow \{0,1\}$
4. Pick $r \hookleftarrow \mathscr{D}_{\mathbf{Z},\sigma}$ and $u \hookleftarrow \mathbf{Z}/p\mathbf{Z}$
5. Compute $\mathrm{C} = g^r, \mathrm{D} = h^r f^u$
6. For $1 \leq i \leq \ell$:
7.     Compute $\mathrm{E}_i = f^{y_{\beta,i}}\mathrm{C}^{s_i}\mathrm{D}^{t_i}$
8. $\mathrm{C}_{\mathbf{y}} = (\mathrm{C}, \mathrm{D}, \{\mathrm{E}_i\}_{i\in[\ell]})$
9. $\beta' \leftarrow \mathscr{A}^{\mathsf{KeyDer}(msk,\cdot)}(\mathrm{C}_{\mathbf{y}})$
10. Return $(\beta = \beta')$
```

$\alpha, r \hookleftarrow \mathscr{D}_{\mathbf{Z},\sigma}$, is a DH triplet since choosing $\sigma > p^{3/2} \cdot \tilde{s} \cdot \sqrt{2\lambda} > p \cdot \tilde{s} \cdot \sqrt{\lambda}$ ensures that the induced distribution is at distance less than $2^{-\lambda}$ of the uniform distribution in G. In Game 2, the challenger samples a random $u \hookleftarrow \mathbf{Z}/p\mathbf{Z}$ and computes $\mathrm{D} = h^r f^u$. Both games are indistinguishable under the DDH-f assumption:

$$|\Pr[\mathrm{S}_2] - \Pr[\mathrm{S}_1]| = \mathsf{Adv}_{\mathscr{B}}^{\mathsf{DDH\text{-}f}}(\lambda, \mu).$$

Now in Game 2 the challenge ciphertext is:

$$(\mathrm{C} = g^r,\ \mathrm{D} = h^r f^u,\ \{\mathrm{E}_i = f^{y_{\beta,i}} \cdot \mathrm{C}^{s_i} \cdot \mathrm{D}^{t_i} = f^{y_{\beta,i}+ut_i} h_i^r\}_{i\in[\ell]}).$$

Lemma E − 5. *In Game* 2 *the ciphertext* $(\mathrm{C}, \mathrm{D}, \mathrm{E}_1, \ldots, \mathrm{E}_\ell) \in \mathrm{G}^{\ell+2}$ *statistically hides* β *such that* $|\Pr[\mathrm{S}_2] - 1/2| \leq 2^{-\lambda}$.

Intuition. Following the proof methodology of [ALS16], we first delimit the information that is leaked in the challenge ciphertext by only considering the dimension in which both potential challenge ciphertexts differ. Indeed, denoting $\mathbf{z}_\beta = \mathbf{y}_\beta + u \cdot \vec{t} \bmod p$, then projecting $\mathbf{z}_\beta$ onto the subspace generated by $\mathbf{y}_0 - \mathbf{y}_1$ encapsulates all the information revealed by the challenge ciphertext.

Next, we consider the distribution of the projection of the secret key component $\mathbf{t}$ on the subspace generated by $\mathbf{y}_0 - \mathbf{y}_1$, conditionally on the adversary's view (i.e. on the information leaked by private key queries and the public key). This amounts to a distribution over a one dimensional lattice Λ_0. We then reduce this distribution modulo a sub-lattice Λ_0' such that $\Lambda_0/\Lambda_0' \simeq \mathbf{Z}/n\mathbf{Z}$, and using Lemma E − 3 we demonstrate that for an appropriate choice of the standard deviation σ (which defines $\mathscr{D}_{\mathbf{Z}^\ell,\sigma}$, from which

$\vec{t}$ is sampled), the projection of $\vec{t}$ on the subspace generated by $\mathbf{y}_0 - \mathbf{y}_1$ is statistically close to the uniform distribution over $\mathbf{Z}/n\mathbf{Z}$. As a result, $\langle \mathbf{y}, \vec{t} \rangle$ modulo p is also close to the uniform distribution over $\mathbf{Z}/p\mathbf{Z}$, and thus $\mathbf{y}_\beta$ (and therefore β) is statistically hidden in $\mathbf{z}_\beta$.

Proof of Lemma E−5. The ciphertext component $\mathrm{C} = g^r$ information theoretically reveals $r \mod n$. Furthermore, $\forall i \in [\ell]$, E_i information theoretically reveals $y_{\beta,i} + ut_i \mod p$ as the value of h_i^r is fixed from C and the public key. Therefore the challenge ciphertext information theoretically reveals $\vec{z}_\beta = \vec{y}_\beta + u \cdot \vec{t} \mod p$.

Throughout the rest of this proof we demonstrate that $\vec{y}_\beta$ is statistically hidden mod p, thanks to the distribution of $\vec{t}$ conditioned on $\mathscr{A}$'s view.

We denote $\vec{x}_i$ $\mathscr{A}$'s ith query to the key derivation oracle. It must hold that, for all i, $\langle \vec{x}_i, \vec{y}_0 \rangle = \langle \vec{x}_i, \vec{y}_1 \rangle$. Let $d \neq 0$ be the gcd of the coefficients of $\vec{y}_1 - \vec{y}_0$ and define

$$\vec{y} = (y_1, \dots, y_\ell) = 1/d \cdot (\vec{y}_1 - \vec{y}_0) \in \mathbf{Z}^\ell.$$

It holds that $\langle \vec{x}_i, \vec{y} \rangle = 0$ over $\mathbf{Z}$ for all i. Therefore if we consider the lattice

$$\vec{y}^\perp = \{\vec{x} \in \mathbf{Z}^\ell : \langle \vec{x}, \vec{y} \rangle = 0\},$$

all the queries $\vec{x}_i$ must belong to that lattice. W.l.o.g., we assume the n_0 first coordinates of $\vec{y}$ are zero (for some n_0), and all remaining entries are non-zero. Further, the rows of the following matrix form a basis of $\vec{y}^\perp$:

$$\mathbf{X}_{\mathsf{top}} = \left[\begin{array}{c|ccccc} \mathbf{I}_{n_0} & & & & & \\ \hline & -y_{n_0+2} & y_{n_0+1} & & & \\ & & -y_{n_0+3} & y_{n_0+2} & & \\ & & & \ddots & \ddots & \\ & & & & -y_\ell & y_{\ell-1} \end{array}\right] \in \mathbf{Z}^{(\ell-1)\times\ell}.$$

We define the matrix:

$$\mathbf{X} = \left[\begin{array}{c} \mathbf{X}_{\mathsf{top}} \\ \hline \vec{y}^{\mathsf{T}} \end{array}\right] \in \mathbf{Z}^{\ell\times\ell}. \tag{E.1}$$

We claim that $\mathbf{X}$ is invertible mod p. The proof follows the same reasoning as [ALS16, Proof of Theorem 2].

Now since $\mathbf{X}$ is invertible over $\mathbf{Z}/p\mathbf{Z}$ and does not depend on $\beta \in \{0,1\}$, it suffices to show that $\mathbf{X} \cdot \mathbf{z}_\beta \in (\mathbf{Z}/p\mathbf{Z})^\ell$ is statistically independent of β. Moreover by construction $\mathbf{X}_{\mathsf{top}} \cdot \vec{y}_0 = \mathbf{X}_{\mathsf{top}} \cdot \vec{y}_1$ (over the integers), so $\mathbf{X}_{\mathsf{top}} \cdot \mathbf{z}_\beta$ is clearly independent of β and we only need to worry about the last row of $\mathbf{X} \cdot \mathbf{z}_\beta \mod p$, i.e. the information about β leaked by the challenge ciphertext is contained in:

$$\langle \vec{y}, \vec{z}_\beta \rangle = \langle \vec{y}, \vec{y}_\beta \rangle + u \cdot \langle \vec{y}, \vec{t} \rangle \mod p. \tag{E.2}$$

We hereafter prove that, from $\mathscr{A}$'s perspective, $\langle \vec{y}, \vec{t} \rangle$ follows a distribution statistically close to the uniform distribution mod p, thus proving that β is statistically hidden: since

u is sampled uniformly at random from $\mathbf{Z}/p\mathbf{Z}$, $u \neq 0 \mod p$ with all but negligible probability as p is a μ-bit prime, with $\mu \geq \lambda$. To this end, we analyse the information that the adversary gains on $\vec{t}$ mod n. From this, we will prove that the distribution of $\langle \vec{y}, \vec{t} \rangle$ is close to uniform mod n, and thus, mod p.

As in the proof of Theorem E−3, the adversary learns $\vec{z} = \vec{s} + \alpha\vec{t}$ mod n from the public key as $\forall\, i \in [\ell], h_i = g^{s_i} h^{t_i}$. Knowing $\vec{z}$, the joint distribution of $(\vec{s}, \vec{t})$ mod n is:

$$(\vec{z} - \alpha\vec{t} \mod n, \vec{t} \mod n) \text{ where } \vec{t} \hookleftarrow \mathscr{D}_{\mathbf{Z}^{\ell},\sigma}.$$

As a result, knowing $\vec{z}$ does not give more information on $\vec{t}$ mod n to $\mathscr{A}$.

One may assume that through its secret key queries, the information learned by $\mathscr{A}$ is completely determined by $\mathbf{X}_{\mathsf{top}} \cdot \vec{s}$ and $\mathbf{X}_{\mathsf{top}} \cdot \vec{t} \in \mathbf{Z}^{(\ell-1)}$, as all the queried vectors $\vec{x}_i$ can be obtained as linear combinations of the rows of $\mathbf{X}_{\mathsf{top}}$.

The value of $\mathbf{X}_{\mathsf{top}} \cdot \vec{s}$ does not give $\mathscr{A}$ more information on $\vec{t}$ mod n than what he obtains from $\mathbf{X}_{\mathsf{top}} \cdot \vec{t}$. Indeed the remainder of the Euclidean division of $\mathbf{X}_{\mathsf{top}} \cdot \vec{s}$ by n can be deduced from $\vec{z}$ and $\mathbf{X}_{\mathsf{top}} \cdot \vec{t}$; while the quotient is independent of $\vec{t}$ mod n and $\mathbf{X}_{\mathsf{top}} \cdot \vec{t}$, as $\vec{s}$ and $\vec{t}$ are sampled independently and $\vec{z}$ only brings a relation modulo n. It is thus sufficient to analyse the distribution of $\vec{t}$ mod n knowing $\mathbf{X}_{\mathsf{top}} \cdot \vec{t}$.

Let $\vec{t}_0 \in \mathbf{Z}^{\ell}$ be an arbitrary vector such that $\mathbf{X}_{\mathsf{top}} \cdot \vec{t}_0 = \mathbf{X}_{\mathsf{top}} \cdot \vec{t}$. Knowing $\mathbf{X}_{\mathsf{top}} \cdot \vec{t}$, the distribution of $\vec{t}$ is $\vec{t}_0 + \mathscr{D}_{\Lambda,\sigma,-\vec{t}_0}$ where $\Lambda = \{\mathbf{t} \in \mathbf{Z}^{\ell} : \mathbf{X}_{\mathsf{top}} \cdot \mathbf{t} = \mathbf{o}\}$. This lattice has dimension 1 and contains $\vec{y} \cdot \mathbf{Z}$. In fact, as $\gcd(y_1, \ldots, y_\ell) = 1$, one has $\vec{y} \cdot \mathbf{Z} = \Lambda$ (there exits $\vec{y}' \in \mathbf{Z}^{\ell}$ such that $\Lambda = \vec{y}' \cdot \mathbf{Z}$ and $\vec{y} = \alpha\vec{y}'$ so α must divide $\gcd(y_1, \ldots, y_\ell) = 1$). Therefore, applying Lemma E−2, we see that conditioned on $\mathbf{X}_{\mathsf{top}} \cdot \vec{t}$, $\langle \vec{y}, \vec{t} \rangle$ is distributed according to

$$\langle \vec{y}, \vec{t}_0 \rangle + \mathscr{D}_{\|\vec{y}\|_2^2 \mathbf{Z}, \|\vec{y}\|_2 \sigma, -\langle \vec{t}_0, \vec{y} \rangle}.$$

Now consider the distribution obtained by reducing $\mathscr{D}_{\|\vec{y}\|_2^2 \mathbf{Z}, \|\vec{y}\|_2 \sigma, -\langle \vec{t}_0, \vec{y} \rangle}$ over $\Lambda_0 = \|\vec{y}\|_2^2 \cdot \mathbf{Z}$ modulo the sublattice $\Lambda_0' = n \cdot \|\vec{y}\|_2^2 \cdot \mathbf{Z}$. In order to apply Lemma E−3 we need $\|\vec{y}\|_2 \cdot \sigma > \eta_\epsilon(\Lambda_0')$, which – applying a bound on the smoothing parameter from [MR07] for $\epsilon = 2^{-\lambda-1}$ – is guaranteed by choosing

$$\|\vec{y}\|_2 \cdot \sigma > \lambda_1(\Lambda_0') \cdot \sqrt{\lambda}.$$

Moreover since $\lambda_1(\Lambda_0') = n \cdot \|\vec{y}\|_2^2$, we require

$$\|\vec{y}\|_2 \cdot \sigma > p \cdot \tilde{s} \cdot \|\vec{y}\|_2^2 \cdot \sqrt{\lambda},$$

thus

$$\sigma > p \cdot \tilde{s} \cdot \|\vec{y}\|_2 \cdot \sqrt{\lambda}.$$

Now from the norm bounds on $\vec{y}_0$ and $\vec{y}_1$ it holds that $\|\vec{y}\|_2 < \sqrt{2p}$, so choosing

$$\sigma > p^{3/2} \cdot \tilde{s} \cdot \sqrt{2\lambda}$$

suffices to ensure that from $\mathscr{A}'s$ view, $\langle\vec{y},\vec{t}\rangle$ modulo n is within distance $2^{-\lambda}$ from the uniform distribution over $\Lambda_0/\Lambda_0' \simeq \mathbf{Z}/n\mathbf{Z}$. As a result, $\langle\vec{y},\vec{t}\rangle$ modulo p is also close to the uniform distribution over $\mathbf{Z}/p\mathbf{Z}$.

We have therefore demonstrated that in eq. (E.2) with overwhelming probability the term $\langle\vec{y},\vec{y}_\beta\rangle$ is statistically hidden modulo p and $|\mathrm{Pr}[\mathsf{S}_2] - 1/2| \leq 2^{-\lambda}$. □

Combining the different transition probabilities provides a bound for $\mathscr{A}$'s advantage, thus concluding the proof: $\mathsf{Adv}_{\mathscr{A}}^{\text{ind-fe-cpa}}(\lambda,\mu) \leq \mathsf{Adv}_{\mathscr{B}}^{\mathsf{DDH\text{-}f}}(\lambda,\mu) + 2^{-\lambda}$. □

E.4.2. DDH-f-based FE for inner product over $\mathbf{Z}/p\mathbf{Z}$

As in the LWE and Paillier-based IPFE modulo p put forth in [ALS16], the main problem encountered here is that private key queries are performed over the integers. An adversary may therefore query keys for vectors that are linearly dependant over $(\mathbf{Z}/p\mathbf{Z})^\ell$ but independent over $\mathbf{Z}^\ell$. To solve this issue we require as in [ALS16] that the authority distributing private keys keeps track of all previously revealed private keys.

Setting the parameters.

As in the previous construction, we use the output $(p,\tilde{s},f,g_p,G,F,G^p)$ of the GenGroup generator of Def. E − 6, with p a μ bit prime, and with $\mu \geq \lambda$. We sample the coordinates of the secret key from $\mathscr{D}_{\mathbf{Z}^\ell,\sigma}$. Choosing $\sigma > \tilde{s}\cdot p^\ell\cdot\sqrt{\lambda}\cdot(\sqrt{\ell})^{\ell-1}$ suffices for security to hold (*cf.* proof of Theorem E − 6), and ensures the distribution $\{g^x, x \leftarrow \mathscr{D}_{\mathbf{Z},\sigma}\}$ is at distance less than $2^{-\lambda}$ from the uniform distribution in G (*cf.* Lemma E − 4, Item 2). The Encrypt algorithm encrypts plaintexts $\vec{y} \in (\mathbf{Z}/p\mathbf{Z})^\ell$ and the key derivation algorithm derives keys from vectors $\mathbf{x} \in (\mathbf{Z}/p\mathbf{Z})^\ell$.

Construction.

Algorithms Setup and Encrypt proceed exactly as in the construction for inner products over $\mathbf{Z}$ under DDH-f (*cf.* Fig. E.3). Algorithms KeyDer and Decrypt, which differ from those of the previous construction, are defined in Fig. E.4. Again, correctness follows from the linearity of the inner product.

Theorem E − 6. *Under the* DDH-f *assumption, the functional encryption scheme for inner products over* $\mathbf{Z}/p\mathbf{Z}$ *of Fig.* E.4 *provides full security* (ind-fe-cpa)*.*

Proof. The proof proceeds similarly to that of Theorem E − 5, only we must define the matrix $\mathbf{X}_{\mathsf{top}}$ differently, as we can no longer guarantee that it is invertible modulo p. So we here follow the same steps as in the previous proof up until the definition of Game 2. The only difference being that the adversary $\mathscr{A}$ queries the *stateful* key derivation algorithm. We denote Game i' the variant of Game i in which the key derivation algorithm is stateful. From the proof of Theorem E − 5, it holds that $|\,\mathrm{Pr}[\mathsf{S}_2'] - \mathrm{Pr}[\mathsf{S}_0']| = \mathsf{Adv}_{\mathscr{B}}^{\mathsf{DDH\text{-}f}}(\lambda,\mu)$.

Algorithm KeyDer($msk, \vec{x}, st$)

Answering the j^{th} key request $sk_{\vec{x}}$ where $\vec{x} \in (\mathbf{Z}/p\mathbf{Z})^\ell$. At any time the internal state st contains at most ℓ tuples $(\vec{x}_i, \bar{\mathbf{x}}_i, z_{\mathbf{x}_i})$ where $(\bar{\mathbf{x}}_i, z_{\mathbf{x}_i})$ are previously queried secret keys and the $\vec{x}_i$'s are corresponding vectors.

```
If $\vec{x}$ is linearly independent of the $\vec{x}_i$'s modulo $p$ :
    Set $\bar{\mathbf{x}} \in \{0, \dots, p-1\}^\ell$ with $\bar{\mathbf{x}} = \vec{x} \mod p$
    $z_{\mathbf{x}} = (s_{\mathbf{x}}, t_{\mathbf{x}}) = (\langle \bar{\mathbf{x}}, \vec{s} \rangle, \langle \bar{\mathbf{x}}, \mathbf{t} \rangle) \in \mathbf{Z} \times \mathbf{Z}$
    $st = (st, (\vec{x}, \bar{\mathbf{x}}, z_{\mathbf{x}}))$
If $\exists \{k_i\}_{1 \le i \le j-1} \in \mathbf{Z}^{j-1}$ s.t. $\vec{x} = \sum_{i=1}^{j-1} k_i \vec{x}_i \in (\mathbf{Z}/p\mathbf{Z})^\ell$ then:
    $\bar{\mathbf{x}} = \sum_{i=1}^{j-1} k_i \bar{\mathbf{x}}_i \in \mathbf{Z}^\ell$
    $z_{\mathbf{x}} = (\sum_{i=1}^{j-1} k_i s_{\vec{x}_i}, \sum_{i=1}^{j-1} k_i t_{\vec{x}_i}) \in \mathbf{Z} \times \mathbf{Z}$
Return $sk_{\vec{x}} = (\bar{\mathbf{x}}, z_{\mathbf{x}})$
```

Algorithm Decrypt($mpk, \mathsf{C}_{\mathbf{y}}, sk_{\vec{x}}$)

```
Parse $(\bar{\mathbf{x}} = (\bar{x}_1, \dots, \bar{x}_\ell);\ z_{\mathbf{x}} = (s_{\mathbf{x}}, t_{\mathbf{x}})) = sk_{\vec{x}}$
Compute $\mathsf{C}_{\mathbf{x}} = (\prod_{i \in [\ell]} \mathsf{E}_i^{\bar{x}_i}) / (\mathsf{C}^{s_{\mathbf{x}}} \cdot \mathsf{D}^{t_{\mathbf{x}}})$
Return Solve($\mathsf{C}_{\vec{x}}$)
```

Figure E.4: Stateful FE scheme for inner products over $\mathbf{Z}/p\mathbf{Z}$ from DDH-f.

As in the original Game 2, here in Game 2′ the challenge ciphertext information theoretically reveals $\vec{z}_\beta = \vec{y}_\beta + u \cdot \vec{t} \mod p$

We define $\vec{y} = (y_1, \dots, y_\ell) = \vec{y}_1 - \vec{y}_0 \in (\mathbf{Z}/p\mathbf{Z})^\ell$; and, assuming $\mathcal{A}$ has performed j private key queries, for $1 \le i \le j$, we denote $\vec{x}_i \in (\mathbf{Z}/p\mathbf{Z})^\ell$ the vectors for which keys have been derived.

We want to demonstrate that from $\mathcal{A}$'s view, the bit β is statistically hidden in Game 2′. However we cannot use the same matrix $\mathbf{X}_{\mathsf{top}}$ as in the proof of Theorem E−5; indeed, if we define $\vec{\mathrm{X}}$ as in eq. (E.1) we cannot guarantee that $\vec{\mathrm{X}}$ is invertible modulo p, since $\det(\vec{\mathrm{X}}\vec{\mathrm{X}}^{\mathrm{T}})$ could be a multiple of p. Therefore, so as to ensure that the queried vectors $\vec{x}_i$ do not in some way depend on β, we prove via induction that after the j first private key queries (where $j \in \{0, \dots, \ell - 1\}$), $\mathcal{A}$'s view remains statistically independent of β, thus proving that the challenge ciphertext in Game 2′ statistically hides β such that $|\Pr[S_2'] - 1/2| \le 2^{-\lambda}$. The induction proceeds on the value of j.

Recall that Game 2 and Game 2′ are identical but for the key derivation algorithm.

Therefore if the adversary can make no calls to its key derivation oracle, the indistinguishability of ciphertexts in Game $2'$ follows immediately from that in Game 2, demonstrated in the proof of Theorem E−5, thus the induction hypothesis holds for $j = 0$. Now consider $j \in \{0, \dots, \ell - 1\}$. From the induction hypothesis one may assume that at this point the state $st = \{(\vec{x}_i, \overline{\mathbf{x}}_i, z_{\mathbf{x}_i})\}_{i \in [j]}$ is independent of β. Indeed if $\mathcal{A}$'s view after $j-1$ requests is independent of β then the j^{th} request performed by $\mathcal{A}$ must be so.

W.l.o.g. one may assume that the key requests $\vec{x}_i$ performed by the adversary are linearly independent. This implies that the $\overline{\mathbf{x}}_i$'s are linearly independent modulo p and generate a subspace of

$$\vec{y}^{\perp p} = \{\vec{x} \in (\mathbf{Z}/p\mathbf{Z})^\ell : \langle \vec{x}, \vec{y} \rangle = 0 \mod p\}.$$

The set $\{\overline{\mathbf{x}}_i\}_{i \in [j]}$ can be extended to a basis $\{\overline{\mathbf{x}}_i\}_{1 \le i \le \ell - 1}$ of $\vec{y}^{\perp p}$. We define $\mathbf{X}_{\mathsf{top}} \in \mathbf{Z}^{(\ell-1) \times \ell}$ to be the matrix whose rows are the vectors $\overline{\mathbf{x}}_i$ for $i \in [\ell - 1]$. Let $\vec{x}' \in (\mathbf{Z}/p\mathbf{Z})^\ell$ be a vector chosen deterministically, $\vec{x}' \notin \vec{y}^{\perp p}$, such that the adversary $\mathcal{A}$ can also easily compute $\vec{x}'$. We define $\mathbf{x}_{\mathsf{bot}}$ to be the canonical lift of $\vec{x}'$ over $\mathbf{Z}$, and $\mathbf{X}$ as:

$$\mathbf{X} = \left[\begin{array}{c} \mathbf{X}_{\mathsf{top}} \\ \hline \mathbf{x}_{\mathsf{bot}}^{\mathsf{T}} \end{array}\right] \in \mathbf{Z}^{\ell \times \ell}.$$

The matrix $\mathbf{X}$ is invertible modulo p, statistically independent of β by induction and by construction, and computable by $\mathcal{A}$, thus we need only prove that $\mathbf{X} \cdot \vec{z}_\beta$ is statistically independent of β. And since $\mathbf{X}_{\mathsf{top}} \cdot (\vec{y}_1 - \vec{y}_0) = 0 \mod p$, we need only consider

$$\langle \mathbf{x}_{\mathsf{bot}}, \vec{z}_\beta \rangle = \langle \mathbf{x}_{\mathsf{bot}}, \vec{y}_\beta \rangle + u \cdot \langle \mathbf{x}_{\mathsf{bot}}, \vec{t} \rangle \mod p.$$

We hereafter prove that, from $\mathcal{A}$'s perspective, $\langle \mathbf{x}_{\mathsf{bot}}, \vec{t} \rangle$ follows a distribution statistically close to the uniform distribution modulo p, thus proving that β is statistically hidden: since u is sampled uniformly at random from $\mathbf{Z}/p\mathbf{Z}$, $u \neq 0 \mod p$ with all but negligible probability as p is a μ bit prime, with $\mu \geq \lambda$. To this end, we analyse the information gained by $\mathcal{A}$ on $\vec{t} \bmod n$. From this, we prove that $\vec{t} \bmod p$ follows a distribution statistically close to the uniform distribution over $\vec{y} \cdot \mathbf{Z}/p\mathbf{Z}$, thus proving that $\langle \mathbf{x}_{\mathsf{bot}}, \vec{t} \rangle$ follows a distribution statistically close to uniform modulo p.

As in the proof of Theorem E−3, the adversary learns $\vec{z} := \vec{s} + \alpha \vec{t}$ modulo n from the public key as $\forall\, i \in [\ell], h_i = g^{s_i} h^{t_i}$. Knowing $\vec{z}$, the joint distribution of $(\vec{s}, \vec{t})$ modulo n is $(\vec{z} - \alpha \vec{t} \mod n, \vec{t} \mod n)$ where $\vec{t} \hookleftarrow \mathcal{D}_{\mathbf{Z}^\ell, \sigma}$.

As a result, knowing $\vec{z}$ does not give $\mathcal{A}$ more information on $\vec{t} \bmod n$. Then, as in the proof of Theorem E−5, private key queries give the adversary the knowledge of $\mathbf{X}_{\mathsf{top}} \cdot \vec{s}$ and $\mathbf{X}_{\mathsf{top}} \cdot \vec{t}$ in $\mathbf{Z}^{\ell - 1}$. The value of $\mathbf{X}_{\mathsf{top}} \cdot \vec{s}$ does not give the adversary more information on $\vec{t}$ modulo n than what he obtains from $\mathbf{X}_{\mathsf{top}} \cdot \vec{t}$. It is thus sufficient to analyse the distribution of $\vec{t}$ modulo n knowing $\mathbf{X}_{\mathsf{top}} \cdot \vec{t}$.

We define $\Lambda = \{\vec{x} \in \mathbf{Z}^\ell | \mathbf{X}_{\mathsf{top}} \cdot \vec{x} = \vec{0} \in \mathbf{Z}^\ell\}$. This one dimensional lattice can equivalently be defined as $\Lambda = \vec{y}' \cdot \mathbf{Z}$ where $\vec{y}' = \gamma \cdot \vec{y} \mod p$ for some $\gamma \in (\mathbf{Z}/p\mathbf{Z})^*$. One should note that all the coefficients of $\vec{y}'$ are co-prime (since $\vec{y}' / \gcd(y'_1, \dots, y'_\ell) \in \Lambda$).

Let $\vec{t}_0 \in \mathbf{Z}^\ell$ be an arbitrary vector such that $\mathbf{X}_{\mathsf{top}} \cdot \vec{t}_0 = \mathbf{X}_{\mathsf{top}} \cdot \vec{t}$. Knowing $\mathbf{X}_{\mathsf{top}} \cdot \vec{t}$, the distribution of $\vec{t}$ is $\vec{t}_0 + \mathscr{D}_{\Lambda,\sigma,-\vec{t}_0}$. Now consider the distribution obtained by reducing the distribution $\mathscr{D}_{\Lambda,\sigma,-\vec{t}_0}$ over Λ modulo the sublattice $\Lambda' := n \cdot \Lambda$. We first bound $\|\vec{y}'\|_2$ so as to bound $\lambda_1(\Lambda')$. We can then apply Lemma E−3 by imposing a lower bound for σ.

Since $\Lambda = \vec{y}' \cdot \mathbf{Z}$, it holds that $\|\vec{y}'\|_2 = \det(\Lambda)$. We define Λ_{top} as the lattice generated by the rows of $\mathbf{X}_{\mathsf{top}}$, then applying results from [Mar03] and [Ngu91], one obtains that

$$\|\vec{y}'\|_2 = \det(\Lambda) \leq \det(\Lambda_{\mathsf{top}}).$$

We now apply Hadamard's bound, which tells us that, since the coordinates of each $\overline{\mathbf{x}}_i$ are smaller than p and since we assumed all requested $\overline{\mathbf{x}}_i$'s are linearly independent,

$$\det(\Lambda_{\mathsf{top}}) \leq \prod_{i=1}^{\ell-1} \|\overline{\mathbf{x}}_i\|_2 \leq (\sqrt{\ell} p)^{\ell-1}.$$

Therefore $\|\vec{y}'\|_2 \leq (\sqrt{\ell} p)^{\ell-1}$, this implies

$$\lambda_1(\Lambda') \leq n \cdot (\sqrt{\ell} p)^{\ell-1} < \tilde{s} \cdot p^\ell \cdot (\sqrt{\ell})^{\ell-1}.$$

From [MR07] we know that the smoothing parameter verifies

$$\eta_\epsilon(\Lambda') \leq \sqrt{\frac{\ln(2(1+1/\epsilon))}{\pi}} \cdot \lambda_1(\Lambda').$$

Thus for $\epsilon = 2^{-\lambda-1}$, we have $\eta_\epsilon(\Lambda') \leq \tilde{s} \cdot p^\ell \cdot \sqrt{\lambda} \cdot (\sqrt{\ell})^{\ell-1}$. Therefore setting

$$\sigma > \tilde{s} \cdot p^\ell \cdot \sqrt{\lambda} \cdot (\sqrt{\ell})^{\ell-1}$$

and applying Lemma E−3 ensures that the distribution $\mathscr{D}_{\Lambda,\sigma,-\vec{t}_0} \mod \Lambda'$, and therefore that of $\vec{t}$ mod n is within distance $2^{-\lambda}$ from the uniform distribution over $\Lambda/\Lambda' \simeq \vec{y}' \cdot \mathbf{Z}/n\mathbf{Z}$. This entails that $\vec{t}$ mod p is within distance $2^{-\lambda}$ from the uniform distribution over $\vec{y}' \cdot \mathbf{Z}/p\mathbf{Z} \simeq \vec{y} \cdot \mathbf{Z}/p\mathbf{Z}$ since $\vec{y}' = \gamma \cdot \vec{y} \mod p$ for some $\gamma \in (\mathbf{Z}/p\mathbf{Z})^*$.

Since by construction $\langle \mathbf{x}_{\mathsf{bot}}, \vec{y} \rangle \neq 0 \mod p$, we get that $\langle \mathbf{x}_{\mathsf{bot}}, \vec{t} \rangle$ modulo p is statistically close to the uniform distribution over $\mathbf{Z}/p\mathbf{Z}$. Moreover, with overwhelming probability $u \neq 0 \mod p$, so $u \cdot \langle \mathbf{x}_{\mathsf{bot}}, \vec{t} \rangle$ statistically hides $\langle \mathbf{x}_{\mathsf{bot}}, \vec{y}_\beta \rangle$ which implies that $\langle \mathbf{x}_{\mathsf{bot}}, \vec{z}_\beta \rangle$ does not carry significant information about β, thus concluding the proof. □

E.5. Inner product FE relying on the HSM assumption

We here build IPFE schemes from the HSM assumption and the ind-cpa scheme described in Fig. E.2(a), using the formalism of a cyclic group with an easy DL subgroup. Our approach is inspired by, and somewhat generalises, the approach of [ALS16] with

Paillier's DCR assumption (an RSA integer N plays the role of p in this scheme so one should invoke the factoring assumption in our proof in order to encompass this construction). We first present an FE scheme for inner products over $\mathbf{Z}$ and then consider a scheme for inner products over $\mathbf{Z}/p\mathbf{Z}$.

E.5.1. HSM-based FE for inner product over Z

Setting the parameters. As in the ind-cpa scheme of Fig. E.2(a), we use the output $(p, \tilde{s}, g, f, g_p, \mathrm{G}, \mathrm{F}, \mathrm{G}^p)$ of the GenGroup generator of Def. E−6. We ignore the generator g (which is useless here). We require that p is a μ bit prime, with $\mu \geq \lambda$. The message space and decryption key space is $\mathbf{Z}^\ell$. As in Subsection E.4.1 norm bounds $\mathrm{X}, \mathrm{Y} < (p/2\ell)^{1/2}$ are chosen to ensure decryption correctness. Key vectors $\mathbf{x}$ and message vectors $\mathbf{y}$ are assumed to have an infinite norm bounded by X and Y respectively. The decryption algorithm uses a centered modulus to recover $\langle \mathbf{x}, \mathbf{y} \rangle$ over $\mathbf{Z}$. To guarantee the scheme's security we sample the coordinates of the secret key $\vec{s} = (s_1, \dots, s_\ell)^{\mathrm{T}} \hookleftarrow \mathscr{D}_{\mathbf{Z}^\ell, \sigma}$ with discrete Gaussian entries of standard deviation $\sigma > \sqrt{2\lambda} \cdot p^{3/2} \cdot \tilde{s}$. Setting $\sigma' > \tilde{s}\sqrt{\lambda}$ ensures that $\{g_p^r, r \hookleftarrow \mathscr{D}_{\mathbf{Z}^\ell, \sigma'}\}$ is at distance less than $2^{-\lambda}$ from the uniform distribution in G^p.

Construction. Fig. E.5 depicts our functional encryption for inner products over $\mathbf{Z}$ construction which relies on the HSM assumption. The proof of correctness is similar to that of the DDH-f construction.

Theorem E−7. *Under the* HSM *assumption, the functional encryption scheme for inner products over* $\mathbf{Z}$ *depicted in Fig. E.5 provides full security* (ind-fe-cpa).

Proof. The proof proceeds as a sequence of games, starting with the real ind-fe-cpa game (Game 0) and ending in a game where the ciphertext statistically hides the random bit β chosen by the challenger from the adversary's point of view. The beginning of the proof is similar to the proof of Theorem E−2 on ind-cpa security. Then we take into account the fact that the adversary $\mathscr{A}$ has access to a key derivation oracle. For each Game i, we denote S_i the event $\beta = \beta'$.
Game 0 ⇒ Game 1: In Game 1 the challenger uses the secret key $\vec{s} = (s_1, \dots, s_\ell)$ to compute ciphertext elements $\mathrm{C}_i = f^{y_{\beta,i}} \cdot (g_p^r)^{s_i} = f^{y_{\beta,i}} \cdot \mathrm{C}_0^{s_i}$. This change does not impact the distribution of the obtained ciphertext, therefore the adversary's success probability in both games is identical: $Pr[\mathrm{S}_0] = Pr[\mathrm{S}_1]$.
Game 1 ⇒ Game 2: In Game 1, the distribution of C_0 is at distance less than $2^{-\lambda}$ of the uniform distribution in the subgroup G^p. Thus under the HSM assumption, we can, in Game 2, substitute C_0 by $g_p^r \cdot f^a \in \mathrm{G}$, with $r \hookleftarrow \mathscr{D}_p, a \hookleftarrow \mathbf{Z}/p\mathbf{Z}$, which, as stated in Lemma E−4, Item 5, is indeed at distance less than $2^{-\lambda}$ of the uniform distribution in G. Therefore, $|Pr[\mathrm{S}_2] - Pr[\mathrm{S}_1]| \leq \mathsf{Adv}_{\mathscr{B}}^{\mathsf{HSM}}(\lambda, \mu)$.
Now in Game 2 we have, for $a \hookleftarrow \mathbf{Z}/p\mathbf{Z}$ and $r \hookleftarrow \mathscr{D}_{\mathbf{Z}, \sigma'}$:

$$\begin{cases} \mathrm{C}_0 = f^a \cdot g_p^r \\ \mathrm{C}_i = f^{y_{\beta,i} + a \cdot s_i} \cdot h_i^r \end{cases} . \tag{E.3}$$

Algorithm $\mathsf{Setup}(1^\lambda, 1^\mu, 1^\ell, X, Y)$

```
1. $(p, \tilde{s}, f, g_p, G, F, G^p) \leftarrow \mathsf{Gen}(1^\lambda, 1^\mu)$
2. $\vec{s} = (s_1, \dots, s_\ell)^T \hookleftarrow \mathscr{D}_{\mathbf{Z}^\ell, \sigma}$
3. For $1 \le i \le \ell$ :
4.     Compute $h_i = g_p^{s_i}$
5. Return $mpk = (\tilde{s}, g_p, f, p, \{h_i\}_{i \in [\ell]})$, $msk = \vec{s}$.
```

Algorithm $\mathsf{KeyDer}(msk, \mathbf{x})$

$\mathbf{x} = (x_1, \dots, x_\ell)^T \in \mathbf{Z}^\ell$,

```
1. Compute $sk_{\vec{x}} = \langle \vec{s}, \vec{x} \rangle$ over $\mathbf{Z}$.
2. Return $sk_{\vec{x}}$
```

Algorithm $\mathsf{Encrypt}(mpk, \mathbf{y})$

$\mathbf{y} = (y_1, \dots, y_\ell)^T \in \mathbf{Z}^\ell$,

```
1. Pick $r \hookleftarrow \mathscr{D}_{\mathbf{Z}, \sigma'}$
2. Compute $C_0 = g_p^r$
3. For $1 \le i \le \ell$ :
4.     Compute $C_i = f^{y_i} \cdot h_i^r$
5. Return $C_{\mathbf{y}} = (C_0, C_1, \dots, C_\ell)$
```

Algorithm $\mathsf{Decrypt}(mpk, C_{\mathbf{y}}, sk_{\vec{x}})$

```
1. Set $C_{\mathbf{x}} = \left(\prod_{i \in [\ell]} C_i^{x_i}\right) \cdot C_0^{-sk_{\vec{x}}}$
2. $\mathsf{sol} \leftarrow \mathsf{Solve}(C_{\mathbf{x}})$
3. If $\mathsf{sol} \ge p/2$ :
4.     Return $(\mathsf{sol} - p)$
5. Else return $\mathsf{sol}$
```

Figure E.5: FE scheme for inner product over $\mathbf{Z}$ from the HSM assumption.

Lemma E − 6. *In Game* 2 *the ciphertext* $C_{\boldsymbol{y}} = (C_0, C_1, \dots, C_\ell) \in G^{\ell+1}$ *statistically hides* β *such that* $|\Pr[S_2] - 1/2| \le 2^{-\lambda}$.

Proof. Let us begin with an overview of the proof. As in proof of Lemma E−5, we first delimit the information that is leaked in the challenge ciphertext by considering the dimension in which both potential challenge ciphertexts differ. Indeed, we denote $\mathbf{z}_\beta = \mathbf{y}_\beta + a\vec{s} \mod p$, then projecting $\mathbf{z}_\beta$ onto the subspace generated by $\mathbf{y}_0 - \mathbf{y}_1$ encapsulates all the information revealed by the challenge ciphertext.
Next, we consider the distribution of the projection of the secret key $\mathbf{s}$ on the subspace generated by $\mathbf{y}_0 - \mathbf{y}_1$, conditionally on the adversary's view (i.e. from the information leaked by private key queries and the public key). This amounts to a distribution over a one dimensional lattice Λ_0. We then reduce this distribution modulo a sub-lattice Λ_0' such that $\Lambda_0/\Lambda_0' \simeq \mathbf{Z}/p\mathbf{Z}$, and using Lemma E−3 one gets that choosing $\sigma > \sqrt{2\lambda} \cdot \tilde{s} \cdot p^{3/2}$ suffices to ensure that the distribution of the projection of $\vec{s}$ on the subspace generated by $\mathbf{y}_0 - \mathbf{y}_1$ is within distance $2^{-\lambda}$ from the uniform distribution over $\mathbf{Z}/p\mathbf{Z}$, and thus $\mathbf{y}_\beta$ (and therefore β) is statistically hidden in $\mathbf{z}_\beta$.

We now provide the full proof that in Game 2 the ciphertext $C_{\mathbf{y}} = (C_0, C_1, \dots, C_\ell) \in$

Game 1

```
1. $mpk, msk \leftarrow \mathsf{Setup}(1^\lambda, 1^\mu, 1^\ell, X, Y)$
2. Parse $(s_1, \dots, s_\ell)^T = msk$
3. Parse $(\tilde{s}, g_p, f, p, \{h_i\}_{i \in [\ell]}) = mpk$
4. $\mathbf{y}_0, \mathbf{y}_1 \leftarrow \mathscr{A}^{\mathsf{KeyDer}(msk,\cdot)}(mpk)$
5. Pick $\beta \hookleftarrow \{0,1\}$
6. Pick $r \hookleftarrow \mathscr{D}_{\mathbf{Z},\sigma'}$
7. Compute $C_0 = g_p^r \in G^p$
8. For $1 \leq i \leq \ell$ :
9.     Compute $C_i = f^{y_{\beta,i}} \cdot C_0^{s_i}$
10. $C_{\mathbf{y}} = (C_0, C_1, \dots, C_\ell)$
11. $\beta' \leftarrow \mathscr{A}^{\mathsf{KeyDer}(msk,\cdot)}(C_{\mathbf{y}})$
12. Return $(\beta = \beta')$
```

Game 2

```
1. $mpk, msk \leftarrow \mathsf{Setup}(1^\lambda, 1^\mu, 1^\ell, X, Y)$
2. Parse $(s_1, \dots, s_\ell)^T = msk$
3. Parse $(\tilde{s}, g_p, f, p, \{h_i\}_{i \in [\ell]}) = mpk$
4. $\mathbf{y}_0, \mathbf{y}_1 \leftarrow \mathscr{A}^{\mathsf{KeyDer}(msk,\cdot)}(mpk)$
5. Pick $\beta \hookleftarrow \{0,1\}$
6. Pick $r \hookleftarrow \mathscr{D}_{\mathbf{Z},\sigma'}$ and $a \hookleftarrow \mathbf{Z}/p\mathbf{Z}$
7. Compute $C_0 = f^a \cdot g_p^r \in G$
8. For $1 \leq i \leq \ell$ :
9.     Compute $C_i = f^{y_{\beta,i}} \cdot C_0^{s_i}$
10. $C_{\mathbf{y}} = (C_0, C_1, \dots, C_\ell)$
11. $\beta' \leftarrow \mathscr{A}^{\mathsf{KeyDer}(msk,\cdot)}(C_{\mathbf{y}})$
12. Return $(\beta = \beta')$
```

$G^{\ell+1}$ statistically hides β such that $|Pr[S_2] - 1/2| \leq 2^{-\lambda}$.

The proof follows the approach of [ALS16, Theorem 5]. Let us first consider the information leaked to $\mathscr{A}$ via private key queries. We denote $\mathbf{x}_i \in \mathbf{Z}^\ell$ the vectors corresponding to secret key queries made by $\mathscr{A}$. As $\mathscr{A}$ is a legitimate adversary, we have $\langle \mathbf{x}_i, \mathbf{y}_0 \rangle = \langle \mathbf{x}_i, \mathbf{y}_1 \rangle$ over $\mathbf{Z}$ for each secret key query $\mathbf{x}_i$.

Thus if we let $d \neq 0$ be the gcd of the coefficients of $\mathbf{y}_1 - \mathbf{y}_0$ and define $\mathbf{y} = (y_1, \dots, y_\ell) = 1/d \cdot (\mathbf{y}_1 - \mathbf{y}_0) \in \mathbf{Z}^\ell$, it holds that all queried vectors $\mathbf{x}_i$ must belong to

$$\vec{y}^\perp = \{\mathbf{x} \in \mathbf{Z}^\ell : \langle \mathbf{x}, \mathbf{y} \rangle = 0\}.$$

We construct matrices $\mathbf{X}_{\mathsf{top}} \in \mathbf{Z}^{(\ell-1)\times\ell}$ and $\vec{X} \in \mathbf{Z}^{\ell\times\ell}$ exactly as in the proof of Theorem E−5, such that the rows of $\mathbf{X}_{\mathsf{top}}$ form a basis of $\vec{y}^\perp$, $\vec{X}$ is invertible modulo p, and:

$$\mathbf{X} = \left[\begin{array}{c} \mathbf{X}_{\mathsf{top}} \\ \hline \mathbf{y}^T \end{array}\right].$$

One may assume that through its secret key queries, the information learned by the adversary is completely determined by $\mathbf{X}_{\mathsf{top}} \cdot \vec{s} \in \mathbf{Z}^{(\ell-1)}$, as all the queried vectors $\mathbf{x}_i$ can be obtained as linear combinations of the rows of $\mathbf{X}_{\mathsf{top}}$.

Now let us consider the information leaked from the challenge ciphertext in Game

2. We recall that it is of the form:

$$\begin{cases} C_0 = f^a \cdot g_p^r \\ C_i = f^{y_{\beta,i}+a\cdot s_i} \cdot h_i^r \end{cases} \qquad \text{for } a \hookleftarrow \mathbf{Z}/p\mathbf{Z} \text{ and } r \hookleftarrow \mathscr{D}_{\mathbf{Z},\sigma'}.$$

We denote:

$$\begin{aligned} \mathbf{z}_\beta &= (y_{\beta,1} + a \cdot s_1, \dots, y_{\beta,\ell} + a \cdot s_\ell) \in (\mathbf{Z}/p\mathbf{Z})^\ell \\ &= \mathbf{y}_\beta + a \cdot \vec{s} \mod p. \end{aligned}$$

As in proofs of Theorems E−5 and E−6, information theoretically, the adversary can glean:

$$\mathbf{y}^{\mathrm{T}} \cdot \mathbf{z}_\beta \mod p = \langle \mathbf{y}, \mathbf{y}_\beta \rangle + a \cdot \langle \mathbf{y}, \vec{s} \rangle \mod p. \tag{E.4}$$

From the public key and from private key queries, the information gained by the adversary amounts at most to:

- a subset of the coordinates of $\mathbf{X}_{\mathsf{top}} \cdot \vec{s} \in \mathbf{Z}^{\ell-1}$ (from private key queries).

- the knowledge that $h_i = g_p^{s_i}$ for $1 \leq i \leq \ell$ which information-theoretically reveals s_i modulo s (from the public key).

Let $\vec{s}_0$ denote an arbitrary vector satisfying the same equations as the secret key $\vec{s}$ from the view of the adversary, i.e.:

$$\mathbf{X}_{\mathsf{top}} \cdot \vec{s}_0 = \mathbf{X}_{\mathsf{top}} \cdot \vec{s} \in \mathbf{Z}^{\ell-1} \quad \wedge \quad \forall i \in [\ell], h_i = g_p^{s_i} = g_p^{s_{0,i}} \in G^p.$$

Denoting $\vec{t} = (t_1, \dots, t_\ell) = \vec{s} - \vec{s}_0$ we can rewrite the above as:

$$\mathbf{X}_{\mathsf{top}} \cdot \mathbf{t} = \mathbf{o} \in \mathbf{Z}^{\ell-1} \quad \wedge \quad \forall i \in [\ell], t_i = 0 \mod s.$$

We define $\Lambda = \{\mathbf{t} \in \mathbf{Z}^\ell | \mathbf{X}_{\mathsf{top}} \cdot \mathbf{t} = \mathbf{o}, \mathbf{t} = \vec{0} \mod s\} \subset \mathbf{Z}^\ell$. Since the protocol samples $\vec{s} \hookleftarrow \mathscr{D}_{\mathbf{Z}^\ell,\sigma}$, from the adversary's view $\vec{s}$ is of the form $\vec{s}_0 + \mathrm{T}$ where T is a random variable with values in Λ. The random variable T follows the same probability distribution as $\vec{s} - \vec{s}_0$ but taken over Λ, i.e.:

$$\begin{aligned} \forall \vec{t} \in \Lambda, \quad \Pr[\mathrm{T} = \vec{t}] &= \mathscr{D}_{\mathrm{Z}^\ell,\sigma,-\vec{s}_0}(\mathbf{t}) / \mathscr{D}_{\mathrm{Z}^\ell,\sigma,-\vec{s}_0}(\Lambda) \\ &= \frac{\rho_{\sigma,-\vec{s}_0}(\mathbf{t})}{\rho_{\sigma,-\vec{s}_0}(\mathbf{Z}^\ell)} \times \frac{\rho_{\sigma,-\vec{s}_0}(\mathbf{Z}^\square)}{\rho_{\sigma,-\vec{s}_0}(\Lambda)} \\ &= \mathscr{D}_{\Lambda,\sigma,-\vec{s}_0}(\mathbf{t}). \end{aligned}$$

Therefore, from the adversary's point of view, the distribution of $\vec{s} \in \mathbf{Z}^\ell$ is:

$$\vec{s}_0 + \mathscr{D}_{\Lambda,\sigma,-\vec{s}_0}.$$

Let us consider the lattice $\Lambda' = \{\mathbf{t} \in \mathbf{Z}^\ell : \mathbf{X}_{\mathsf{top}} \cdot \mathbf{t} = \mathbf{o}\}$. As in the proof of Theorem E−5, this lattice has dimension 1 and $\Lambda' = \vec{y} \cdot \mathbf{Z}$. Moreover

$$\Lambda = \Lambda' \cap (s \cdot \mathbf{Z}^\ell) = (\mathbf{y} \cdot \mathbf{Z}) \cap (s \cdot \mathbf{Z}^\ell) = s \cdot \mathbf{y} \cdot \mathbf{Z},$$

since $\gcd(y_1, \dots, y_\ell) = 1$ (for any $\alpha \in \mathbf{Z}$, for s to divide all $\alpha \cdot y_i$, s must divide $\alpha \cdot \gcd(y_1, \dots, y_\ell) = \alpha$).

We now consider the distribution of $\langle \vec{s}, \vec{y} \rangle$, and then reduce it modulo p, so as to prove that, from the adversary's view (i.e. conditionally on the public key and queried keys), in eq. (E.4) the bit β is statistically hidden. Let us denote $\Lambda_0 = s \cdot \|\mathbf{y}\|_2^2 \cdot \mathbf{Z}$. It follows from Lemma E−2 that the distribution of $\langle \vec{s}, \vec{y} \rangle$ is:

$$\langle \vec{s}_0, \mathbf{y} \rangle + \mathscr{D}_{\Lambda_0, \|\mathbf{y}\|_2 \cdot \sigma, -c}$$

where $c = \langle \vec{s}_0, \mathbf{y} \rangle$ in $\mathbf{Z}$.

In order to prove that the above distribution is statistically close to the uniform distribution over $\mathbf{Z}/p\mathbf{Z}$, we consider the distribution obtained by reducing the distribution $\mathscr{D}_{\Lambda_0, \|\mathbf{y}\|_2 \cdot \sigma, -c}$ over Λ_0 modulo the sublattice $\Lambda'_0 = p\Lambda_0$. Since $\Lambda_0/\Lambda'_0 \simeq \mathbf{Z}/p\mathbf{Z}$, demonstrating that $\langle \mathbf{y}, \vec{s} \rangle \mod p$ is within negligible statistical distance from the uniform distribution over Λ_0/Λ'_0 will conclude the proof.

From Lemma E−3 it follows that to achieve the required smoothing parameter $\eta_\epsilon(\Lambda'_0)$ one must impose a lower bound on the standard deviation, i.e. we need $\|\mathbf{y}\|_2 \cdot \sigma > \eta_\epsilon(\Lambda'_0)$. If we set ϵ to be $2^{-\lambda-1}$, from [MR07] we know that

$$\eta_\epsilon(\Lambda'_0) \le \sqrt{\frac{\ln(2(1+1/\epsilon))}{\pi}} \cdot \lambda_1(\Lambda'_0) < \sqrt{\lambda} \cdot \lambda_1(\Lambda'_0).$$

Since $\lambda_1(\Lambda'_0) = s \cdot \|\mathbf{y}\|_2^2 \cdot p < \tilde{s} \cdot \|\mathbf{y}\|_2^2 \cdot p$, we need

$$\sigma > \|\vec{y}\|_2 \sqrt{\lambda} \cdot p \cdot \tilde{s}$$

(i.e. $\|\mathbf{y}\|_2 \cdot \sigma > \sqrt{\lambda} \cdot \lambda_1(\Lambda'_0)$). Finally as $\|\mathbf{y}\|_2 < \sqrt{2p}$ (since $\|\mathbf{y}_i\|_\infty < \sqrt{p/(2\ell)}$ for $i \in \{0,1\}$) choosing

$$\sigma > \sqrt{2\lambda} \cdot \tilde{s} \cdot p^{3/2}$$

suffices to ensure that $\langle \mathbf{y}, \vec{s} \rangle \mod p$ is within distance $2^{-\lambda}$ from the uniform distribution over $\Lambda_0/\Lambda'_0 \simeq \mathbf{Z}/p\mathbf{Z}$.

Finally, since in eq. (E.4), $a \hookleftarrow \mathbf{Z}/p\mathbf{Z}$ is invertible modulo p with all but negligible probability, the term $\langle \mathbf{y}, \mathbf{y}_\beta \rangle \mod p$ is statistically hidden, and $|\Pr[S_2] - 1/2| \le 2^{-\lambda}$.

□

Over all game transitions, after adding up the different probabilities, we find that $\mathscr{A}$'s advantage in the real game can be bounded as $|\Pr[S_0] - 1/2| \le \mathsf{Adv}_{\mathscr{B}}^{\mathsf{HSM}}(\lambda, \mu) + 2^{-\lambda}$ which is negligible if the HSM assumption holds in G. □

E.5.2. HSM-based FE for inner product over $\mathbf{Z}/p\mathbf{Z}$

As in the DDH-f based scheme for inner products over $\mathbf{Z}/p\mathbf{Z}$ of Section E.4.2, the key generation algorithm is stateful to ensure the adversary cannot query keys for vectors that are linearly dependant over $(\mathbf{Z}/p\mathbf{Z})^\ell$ but independent over $\mathbf{Z}^\ell$.

Setting the parameters.

As in the previous construction, we use the output $(p, \tilde{s}, f, g_p, G, F, G^p)$ of the GenGroup generator of Def. E−6, with p a μ-bit prime, and with $\mu \geq \lambda$. The message space and vector space from which decryption keys are derived are now $(\mathbf{Z}/p\mathbf{Z})^\ell$. Given an encryption of $\vec{y} \in (\mathbf{Z}/p\mathbf{Z})^\ell$ and a decryption key for $\vec{x} \in (\mathbf{Z}/p\mathbf{Z})^\ell$, the decryption algorithm recovers $\langle \mathbf{x}, \mathbf{y} \rangle \in \mathbf{Z}/p\mathbf{Z}$. To guarantee the scheme's security we sample the coordinates of the secret key $\vec{s}$ from $\mathscr{D}_{\mathbf{Z}^\ell,\sigma}$ with discrete Gaussian entries of standard deviation $\sigma > \sqrt{\lambda} \cdot p \cdot \tilde{s} \cdot (\sqrt{\ell} p)^{\ell-1}$. We require $\sigma' > \tilde{s}\sqrt{\lambda}$ to ensure that $\{g_p^r, r \hookleftarrow \mathscr{D}_{\mathbf{Z}^\ell,\sigma'}\}$ is at distance less than $2^{-\lambda}$ from the uniform distribution in G^p.

Construction.

The Setup and Encrypt algorithms proceed exactly as in Fig. E.5, the only difference being that Encrypt operates on message vectors $\vec{y} \in (\mathbf{Z}/p\mathbf{Z})^\ell$ instead of $\vec{y} \in \mathbf{Z}^\ell$. In Fig. E.6 we only define algorithms KeyDer and Decrypt, since they differ from those of the previous construction.

Theorem E − 8. *Under the* HSM *assumption the above stateful functional encryption scheme for inner products over* $\mathbf{Z}/p\mathbf{Z}$ *provides full security* (ind-fe-cpa)*.*

The proof follows the same lines as the proof of the previous theorem and is adapted from the proofs of [ALS16].

The main issue is that we can no longer guarantee that $\vec{X}$ is invertible modulo p. We need to compute on-the-fly a basis for $\{\vec{x} \in (\mathbf{Z}/p\mathbf{Z})^\ell : \langle \vec{x}, \vec{y} \rangle = 0 \mod p\}$ to apply the same techniques as in Theorem E−7. The analysis gives significantly larger standard deviations as mentioned above due a bad approximation of the determinant of a related matrix.

Proof. We here provide the proof that under the HSM assumption the stateful functional encryption scheme for inner products over $\mathbf{Z}/p\mathbf{Z}$ presented above provides full security (ind-fe-cpa).

The proof proceeds similarly to that in $\mathbf{Z}$ (*cf.* Theorem E−7), starting with the real ind-fe-cpa game and ending in a game where the ciphertext statistically hides the random bit β chosen by the challenger from the adversary's point of view.

Games 0 to 2 basically proceed identically to those of the proof of Theorem E−7. The only difference is in the key derivation oracle that the adversary $\mathscr{A}$ has access to,

Algorithm $\mathsf{KeyDer}(msk, \vec{x}, st)$

Answering the j^{th} key request $sk_{\vec{x}}$ where $\vec{x} \in (\mathbf{Z}/p\mathbf{Z})^\ell$. At any time the internal state st contains at most ℓ tuples $(\vec{x}_i, \overline{\mathbf{x}}_i, z_{\mathbf{x}_i})$ where $(\overline{\mathbf{x}}_i, z_{\mathbf{x}_i})$ are previously queried secret keys and the $\vec{x}_i$'s are corresponding vectors.

```
1. If $\vec{x}$ is linearly independent of the $\vec{x}_i$'s modulo $p$ :
2.     Set $\overline{\mathbf{x}} \in \{0, \ldots, p-1\}^\ell$ with $\overline{\mathbf{x}} = \vec{x} \mod p$
3.     $z_{\mathbf{x}} = \langle \vec{s}, \overline{\mathbf{x}} \rangle \in \mathbf{Z}$ ; $st = (st, (\vec{x}, \overline{\mathbf{x}}, z_{\mathbf{x}}))$
4. If $\exists \{k_i\}_{1 \le i \le j-1} \in \mathbf{Z}^{j-1}$ such that $\vec{x} = \sum_{i=1}^{j-1} k_i \vec{x}_i \in (\mathbf{Z}/p\mathbf{Z})^\ell$ then:
5.     $\overline{\mathbf{x}} = \sum_{i=1}^{j-1} k_i \overline{\mathbf{x}}_i \in \mathbf{Z}^\ell$ ; $z_{\mathbf{x}} = \sum_{i=1}^{j-1} k_i z_{\mathbf{x}_i} \in \mathbf{Z}$
6. Return $sk_{\vec{x}} = (\overline{\mathbf{x}}, z_{\mathbf{x}})$
```

Algorithm $\mathsf{Decrypt}(mpk, \mathrm{C}_{\mathbf{y}}, sk_{\vec{x}})$

```
1. Parse $(\overline{\mathbf{x}} = (\overline{x}_1, \ldots, \overline{x}_\ell), z_{\mathbf{x}}) = sk_{\vec{x}}$
2. Compute $\mathrm{C}_{\vec{x}} = \left(\prod_{i \in [\ell]} \mathrm{C}_i^{\overline{x}_i}\right) \cdot (\mathrm{C}_0^{-z_{\mathbf{x}}})$
3. Return $\mathsf{Solve}(\mathrm{C}_{\vec{x}})$
```

Figure E.6: Functional encryption scheme for inner products over $\mathbf{Z}/p\mathbf{Z}$ from HSM.

which now executes the *stateful* key derivation algorithm. Thus we have a Game $2'$ for which:

$$|\Pr[S_2'] - \Pr[S_0]| \le \mathsf{Adv}_{\mathscr{B}}^{\mathsf{HSM}}(\lambda, \mu).$$

Recall that $\mathscr{A}$ can query the key derivation oracle for any vector $\vec{x} \in (\mathbf{Z}/p\mathbf{Z})^\ell$ satisfying $\langle \vec{x}, \vec{y}_0 \rangle = \langle \vec{x}, \vec{y}_1 \rangle \in \mathbf{Z}/p\mathbf{Z}$. For each query, $\mathscr{A}$ is given a secret key $(\overline{\mathbf{x}}, z_{\mathbf{x}})$ as in the real scheme. And in Game $2'$ we have:

$$\begin{cases} \mathrm{C}_0 = f^a \cdot g_p^r \\ \mathrm{C}_i = f^{y_{\beta,i} + a \cdot s_i} \cdot h_i^r, \quad \forall\, i \in [\ell] \end{cases}$$

where $a \hookleftarrow \mathbf{Z}/p\mathbf{Z}$ and $r \hookleftarrow \mathscr{D}_{\mathbf{Z},\sigma'}$.

Therefore, the challenge ciphertext information-theoretically reveals:

$$\vec{z}_\beta = \vec{y}_\beta + a\vec{s} \mod p.$$

We define $\vec{y} = (y_1, \ldots, y_\ell) = \vec{y}_1 - \vec{y}_0 \in (\mathbf{Z}/p\mathbf{Z})^\ell$, and, assuming $\mathscr{A}$ has performed j private key queries, for $1 \le i \le j$, we denote $\vec{x}_i \in (\mathbf{Z}/p\mathbf{Z})^\ell$ the vectors for which keys have been derived.

From here on, demonstrating that in Game 2′ the challenge ciphertext statistically hides the bit β is done as in proof of Theorem E−6, we prove via induction that after the j first private key queries, $\mathscr{A}$'s view remains statistically independent of β, thus proving that $|Pr[S_2'] - 1/2| \leq 2^{-\lambda}$. The induction proceeds on the value of j.

For $j = 0$ the adversary can make no private key queries. With this restriction games 2 and 2′ are identical. It thus follows from the proof of Theorem E−7 that for $j = 0$ the induction hypothesis holds, i.e. $\mathscr{A}$'s view is indeed statistically independent of β.

Consider $j \in \{0, \dots, \ell - 1\}$. From the induction hypothesis one may assume that at this point the state $st = \{(\vec{x}_i, \overline{\mathbf{x}}_i, z_{\mathbf{x}_i}) \in (\mathbf{Z}/p\mathbf{Z})^\ell \times \mathbf{Z}^\ell \times \mathbf{Z}\}_{i\in[j]}$ is independent of β. Indeed if $\mathscr{A}$'s view after $j-1$ requests is independent of β then the j^{th} request performed by $\mathscr{A}$ must be so. W.l.o.g. one may assume that the key requests $\vec{x}_i$ performed by the adversary are linearly independent (otherwise $\mathscr{A}$ does not gain any additional information from its request). This implies that the $\overline{\mathbf{x}}_i$'s are linearly independent modulo p and generate a subspace of:

$$\vec{y}^{\perp p} = \{\vec{x} \in (\mathbf{Z}/p\mathbf{Z})^\ell : \langle \vec{x}, \vec{y} \rangle = 0 \mod p\}$$

Moreover the set $\{\overline{\mathbf{x}}_i\}_{i\in[j]}$ (generated during private key queries) can be extended to a basis $\{\overline{\mathbf{x}}_i\}_{i\in[\ell-1]}$ of $\vec{y}^{\perp p}$. We define $\mathbf{X}_{\mathsf{top}} \in \mathbf{Z}^{(\ell-1)\times\ell}$ to be the matrix whose rows are the vectors $\overline{\mathbf{x}}_i$ for $i \in [\ell - 1]$.

$$\mathbf{X}_{\mathsf{top}} = \begin{bmatrix} \overline{\mathbf{x}}_1^{\mathsf{T}} \\ \overline{\mathbf{x}}_2^{\mathsf{T}} \\ \vdots \\ \overline{\mathbf{x}}_{\ell-1}^{\mathsf{T}} \end{bmatrix}$$

Let $\vec{x}' \in (\mathbf{Z}/p\mathbf{Z})^\ell$ be a vector such that $\vec{x}' \notin \vec{y}^{\perp p}$. This vector $\vec{x}'$ is constructed deterministically from the set $\{\overline{\mathbf{x}}_i\}_{i\in[j]}$ and $\vec{y}$. We define $\mathbf{x}_{\mathsf{bot}}$ to be the canonical lift of $\vec{x}'$ over $\mathbf{Z}$, and $\mathbf{X}$ as:

$$\mathbf{X} = \left[\begin{array}{c} \mathbf{X}_{\mathsf{top}} \\ \hline \mathbf{x}_{\mathsf{bot}}^{\mathsf{T}} \end{array}\right].$$

The matrix $\mathbf{X}$ is built deterministically, invertible modulo p by construction, and independent of β by induction hypothesis and by construction. So $\mathbf{X}$ is known to $\mathscr{A}$ in the information theoretical sense. As in the proof of Theorem E−7 we need only prove that $\mathbf{X} \cdot \vec{z}_\beta$ is statistically independent of β. And since $\mathbf{X}_{\mathsf{top}} \cdot (\vec{y}_1 - \vec{y}_0) = 0 \mod p$, we need only consider:

$$\langle \mathbf{x}_{\mathsf{bot}}, \vec{z}_\beta \rangle = \langle \mathbf{x}_{\mathsf{bot}}, \vec{y}_\beta \rangle + a \langle \mathbf{x}_{\mathsf{bot}}, \vec{s} \rangle \mod p. \tag{E.5}$$

As in the proof of Theorem E−7, let $\vec{s}_0$ denote an arbitrary vector such that from the adversary's point of view, the distribution of $\vec{s} \in \mathbf{Z}^\ell$ is $\vec{s}_0 + \mathscr{D}_{\Lambda,\sigma,-\vec{s}_0}$ where

$$\Lambda = \{\vec{t} \in \mathbf{Z}^\ell : \mathbf{X}_{\mathsf{top}} \cdot \vec{t} = \mathbf{o}, \vec{t} = \vec{0} \mod s\}.$$

We also define $\Lambda' = \{\vec{t} \in \mathbf{Z}^\ell : \mathbf{X}_{\mathsf{top}} \cdot \vec{t} = \mathbf{o}\}$, clearly Λ' is a one-dimensional lattice which can also be defined as $\Lambda' = \vec{y}' \cdot \mathbf{Z}$ for some $\vec{y}' \in \mathbf{Z}^\ell$. One should note that all the coefficients of $\vec{y}'$ are co-prime (since $\vec{y}'/\gcd(y'_1,\ldots,y'_\ell) \in \Lambda'$). Moreover, since $\mathbf{X}_{\mathsf{top}}$ is a basis of $\vec{y}^{\perp p}$, we have $\Lambda' \mod p = \vec{y} \cdot \mathbf{Z}/p\mathbf{Z}$. As a result, there exists $\alpha \in (\mathbf{Z}/p\mathbf{Z})^*$ s.t. $\vec{y}' = \alpha \cdot \vec{y} \mod p$. Finally, $\Lambda = \Lambda' \cap (s\mathbf{Z})^\ell$ and, since the coefficients of $\vec{y}'$ are co-prime $\Lambda = s \cdot \vec{y}' \cdot \mathbf{Z} = s \cdot \Lambda'$.

Intuition. So as to prove that in eq. (E.5), the term $a \cdot \langle \mathbf{x}_{\mathsf{bot}}, \vec{s} \rangle$ statistically hides β, we justify that the distribution $\mathscr{D}_{\Lambda,\sigma,-\vec{s}_0}$ reduced modulo the sub-lattice $p\Lambda$, (and therefore that of $\vec{s}$ reduced modulo p) is statistically close to the uniform distribution over $\vec{y} \cdot \mathbf{Z}/p\mathbf{Z}$. This is done by applying Lemma E−3 and imposing a lower bound for σ. In order to do this we first need to bound $\lambda_1(p\Lambda)$ and therefore $\|\vec{y}'\|_2$. We thereby demonstrate that $\langle \mathbf{x}_{\mathsf{bot}}, \vec{s} \rangle \mod p$ is statistically close to the uniform distribution over $\mathbf{Z}/p\mathbf{Z}$, and therefore, with overwhelming probability $\langle \mathbf{x}_{\mathsf{bot}}, \vec{z}_\beta \rangle$ statistically hides β, thus concluding the proof.

Details. Since $\Lambda' = \vec{y}' \cdot \mathbf{Z}$, it holds that $\|\vec{y}'\|_2 = \det(\Lambda')$. If we define Λ_{top} as the lattice generated by the rows of $\mathbf{X}_{\mathsf{top}}$, then applying results from [Mar03] and [Ngu91, Theorem 2.8], one obtains that

$$\|\vec{y}'\|_2 = \det(\Lambda') \leq \det(\Lambda_{\mathsf{top}}).$$

We now apply Hadamard's bound, which tells us that, since the coordinates of each $\overline{\mathbf{x}}_i$ are smaller than p (since we assumed all requested $\overline{\mathbf{x}}_i$'s are independent), $\det(\Lambda_{\mathsf{top}}) \leq \prod_{i=1}^{\ell-1} \|\overline{\mathbf{x}}_i\|_2 \leq (\sqrt{\ell}p)^{\ell-1}$. Therefore $\|\vec{y}'\|_2 \leq (\sqrt{\ell}p)^{\ell-1}$ and $s \cdot \|\vec{y}'\|_2 < \tilde{s} \cdot (\sqrt{\ell}p)^{\ell-1}$, this implies

$$\lambda_1(p \cdot \Lambda) \leq p\tilde{s} \cdot (\sqrt{\ell}p)^{\ell-1}.$$

From [MR07] we know that the smoothing parameter verifies

$$\eta_\epsilon(p \cdot \Lambda) \leq \sqrt{\frac{\ln(2(1+1/\epsilon))}{\pi}} \cdot \lambda_1(p \cdot \Lambda).$$

Thus for $\epsilon = 2^{-\lambda-1}$, we have $\eta_\epsilon(p \cdot \Lambda) \leq \sqrt{\lambda} \cdot p \cdot \tilde{s} \cdot (\sqrt{\ell}p)^{\ell-1}$. Therefore setting

$$\sigma \geq \sqrt{\lambda} \cdot p \cdot \tilde{s} \cdot (\sqrt{\ell}p)^{\ell-1}$$

and applying Lemma E−3 ensures that the distribution $\mathscr{D}_{\Lambda,\sigma,-\vec{s}_0} \mod (p \cdot \Lambda)$ is within distance $2^{-\lambda}$ from the uniform distribution over $\Lambda/(p\Lambda)$ which is isomorphic to $\vec{y} \cdot \mathbf{Z}/p\mathbf{Z}$ because $\gcd(p,s) = 1$ and $\vec{y}' = \alpha \cdot \vec{y} \mod p$ for some $\alpha \in (\mathbf{Z}/p\mathbf{Z})^*$.

We have thus proven that $\vec{s}$ modulo p is statistically close to the uniform distribution over $\vec{y} \cdot \mathbf{Z}/p\mathbf{Z}$ from the adversary's point of view. Since by construction $\langle \mathbf{x}_{\mathsf{bot}}, \vec{y} \rangle \neq 0 \mod p$, we get that $\langle \mathbf{x}_{\mathsf{bot}}, \vec{s} \rangle$ modulo p is statistically close to the uniform distribution over $\mathbf{Z}/p\mathbf{Z}$. Moreover, in eq. E.5, $\gcd(a,p) = 1$ with overwhelming probability, so $a \cdot \langle \mathbf{x}_{\mathsf{bot}}, \vec{s} \rangle$ statistically hides $\langle \mathbf{x}_{\mathsf{bot}}, \vec{y}_\beta \rangle$, which implies that $\langle \mathbf{x}_{\mathsf{bot}}, \vec{z}_\beta \rangle$ does not carry significant information about β, thus concluding the proof.

□

E.6. Instantiation and efficiency considerations

We put forth two generic constructions of FE for the evaluation of inner products. Both schemes are based on variants of Elgamal in the same group and both sample their master secret keys from Gaussian distributions with the same standard deviation. As a result their asymptotic complexities are the same. The second scheme's security relies on a hard subgroup membership assumption (HSM) and this scheme appears to be the most efficient FE which evaluates inner product modulo a prime p. At the (small) expense of a single additional element in the keys and in the ciphertext, the first scheme's security relies on a weaker DDH-like assumption, which is also weaker than the DDH assumption in the group. We compare, in Table E.1, an implementation of our HSM-based IPFE mod p of Subsection E.5.2 within the class group of an imaginary quadratic field and Paillier's variant of [ALS16]. This is the most relevant comparison since their DDH variant does not allow a full recovery of large inner products over $\mathbf{Z}/p\mathbf{Z}$, and, as detailed in the following paragraph, the LWE variant is far from being efficient, as ciphertexts are computed using arithmetic modulo $q = 2^{\ell}$ where ℓ is the dimension of the plaintext vectors.

Comparison with the LWE based scheme of [ALS16].

Parameter choices for lattice-based cryptography are complex, indeed [ALS16] do not provide a concrete set of parameters. This being said, using [ALS16, Theorem 3], and setting $\log p = \lambda$ as in Table E.1, we give rough bit sizes for their LWE based FE scheme for computing inner products over $\mathbf{Z}/p\mathbf{Z}$. Basic elements are integers modulo q of size ℓ since $q \approx 2^{\ell}$ for security to hold. The largest component in the master public key mpk consists of $\lambda^2\ell^3$ elements, so mpk is of size greater than $\lambda^2\ell^4$. The component $z_{\vec{x}}$ in secret keys is the product of a vector from $(\mathbf{Z}/p\mathbf{Z})^{\ell}$ with a matrix, which yields a secret key vector made up of $\lambda\ell^2$ inner products, where each inner product is of size $\ell\lambda$. Thus these keys are of size $\lambda^2\ell^3$. Finally ciphertexts consist of $\lambda\ell^2$ elements, and are thus of size greater than $\lambda\ell^3$. As a result, although it may be hard to compare the complexities in λ, for a fixed security level, the complexity in ℓ for all the parameters of the LWE based scheme is in ℓ^3 or ℓ^4 whereas we are linear in ℓ as one can see in Table E.1. For example, for $\lambda = 128, \ell = 100$, their $sk_{\vec{x}}$ is of approximately 2^{34} bits vs. 13852 bits in our instantiation.

Instantiation.

To instantiate the protocol of Section E.5.2, we first need to define the algorithm GenGroup of Def. E−6. To this end, we follow the lines of the construction from [CL15]. We start from a fundamental discriminant $\Delta_K = -p \cdot q$ with its class group $Cl(\Delta_K)$, where q is a prime such that $p \cdot q \equiv -1 \pmod 4$ and $(p/q) = -1$. Then, we consider a non-maximal order of discriminant $\Delta_p = p^2 \cdot \Delta_K$ and its class group $Cl(\Delta_p)$. The order of $Cl(\Delta_p)$ is

$$h(\Delta_p) = p \cdot h(\Delta_K).$$

Table E.1: Comparing our IPFE from HSM and the DCR scheme of [ALS16]

	$\lambda = 112$		$\lambda = 128$	
size	this work	DCR	this work	DCR
$(p, \tilde{s})$	$(112, 684)$	$(1024, 2046)$	$(128, 924)$	$(1536, 3070)$
group element	1572	4096	2084	6144
secret key* ($z_{\vec{x}}$)	$112(\ell+1)+684$	$2048(\ell+2)$	$128(\ell+1)+924$	$3072(\ell+2)$
ciphertext	$1572(\ell+1)$	$4096(\ell+1)$	$2084(\ell+1)$	$6144(\ell+1)$
enc. expo.	687	2046	928	3070
dec. expo.	$112(\ell+1)+684$	$2048(\ell+2)$	$128(\ell+1)+924$	$3072(\ell+2)$

* ignoring an additive term $(\ell \pm 1)\log(\sqrt{\ell})$

It is known (*cf.* [Coh00, p. 295]), that

$$h(\Delta_K) < \frac{1}{\pi} \log|\Delta_K| \sqrt{|\Delta_K|}$$

which is the bound we take for $\tilde{s}$ (note that a slightly better bound can be computed from the analytic class number formula, *cf.* [McC89]). In Fig. C.2 of [CL15] the authors show how to build a generator of a cyclic group of order ps of the class group of discriminant Δ_p and a generator for the subgroup of order p (in which the discrete logarithm problem is easy). We need to modify their generator of a DDH group with an easy DL subgroup, to make it output a generator g_p of the subgroup of p-th powers. The computation of such an element is actually implicit in their generator: this is done by computing an ideal $\mathfrak{r}$ in the maximal order with norm a small prime r such that $\left(\frac{\Delta_K}{r}\right) = 1$. Then the ideal $\mathfrak{r}^2$ is lifted into a class of $Cl(\Delta_p)$ which is then raised to the power p to define g_p. A second modification is to output $\tilde{s}$ instead of their larger bound B (since they sampled elements using a folded uniform distribution). We refer to [CL15] for a full description of the implementation. The manipulated objects are reduced ideals represented by two integers smaller than $\sqrt{p^3 q}$, and the arithmetic operations in class groups are very efficient, since the reduction and composition of quadratic forms have a quasi linear time complexity using fast arithmetic (see for instance [Coh00]).

The sole restriction on the size of the prime p is that it must have at least λ bits, where λ is the security parameter. The size of Δ_K, and thus of q, is chosen to thwart the best practical attack, which consists in computing discrete logarithms in $Cl(\Delta_K)$ (or equivalently the class number $h(\Delta_K)$). An index-calculus method to solve the discrete logarithm problem in a class group of imaginary quadratic field of discriminant Δ_K was proposed in [Jac00]. It is conjectured in [BJS10] that a state of the art implementation of this algorithm has complexity $\mathcal{O}(L_{|\Delta_K|}[1/2, o(1)])$. They estimate that the discrete logarithm problem with a discriminant Δ_K of1348 (resp. 1828 bits) is as hard as factoring a 2048 (resp. 3072 bits) RSA integer. This is our reference to estimate the bit size of the different elements in Table E.1.

Note that in this case, the size of our group elements (reduced ideals in the class group of discriminant p^3q), are significantly smaller than those of the Paillier variant of [ALS16] (elements of $\mathbf{Z}/N^2\mathbf{Z}$). This is also the case for ciphertexts (which consist in both protocols of $\ell+1$ group elements). We have the same situation with secret keys: to simplify the comparison we consider linearly independent queries (thus ignoring the vectors in $\mathbf{Z}^\ell$). As a result, we have, for our scheme, the inner product of a vector from $(\mathbf{Z}/p\mathbf{Z})^\ell$ with a vector sampled from a discrete Gaussian with standard deviation greater than $\sqrt{\lambda}p\tilde{s}(\sqrt{\ell}p)^{\ell-1}$ over $\mathbf{Z}^\ell$ vs. the inner product of a vector of $(\mathbf{Z}/N\mathbf{Z})^\ell$ with a vector sampled from a discrete Gaussian with standard deviation greater than $\sqrt{\lambda}(\sqrt{\ell}N)^{\ell+1}$ over $\mathbf{Z}^\ell$.

We note that our underlying message space $\mathbf{Z}/p\mathbf{Z}$ is much smaller than their message space $\mathbf{Z}/N\mathbf{Z}$. Using larger message spaces would be more favorable to their Paillier based scheme. But in practice, a 128 bits message space is large enough, if for instance, one needs to perform computations with double or quadruple precision. Our protocols are the most suited for such intermediate computations, since Paillier's construction from [ALS16] would add a large overhead cost, while their DDH construction could not decrypt the result.

In terms of timings, a fair comparison is difficult since to our knowledge, no library for the arithmetic of quadratic forms is as optimized as a standard library for the arithmetic of modular integers. Nevertheless, we note that the exponents involved in the (multi-)exponentiations (for encryption and decryption) are significantly smaller than those in [ALS16], and the group size is also smaller. Indeed, the encryption of Paillier's variant involves $(\ell+1)$ exponentiations to the power a $(|N|-2)$-bit integer modulo N^2, whereas our protocol involves one exponentiation to the power a $|\sigma'|$-bit integer in $Cl(p^3q)$, where $\sigma' > \tilde{s}\sqrt{\lambda}$ and ℓ (multi-)exponentiations whose maximum exponent size is also $|\sigma'|$. Decryptions involve respectively a multi-exponentiation whose maximum exponent size is lower than $\ell\sigma N = \ell\sqrt{\lambda}(\sqrt{\ell}N)^{\ell+1}N$ for [ALS16] and $\ell p\sigma = \ell p\sqrt{\lambda}p\tilde{s}(\sqrt{\ell}p)^{\ell-1}$ for our protocol.

Table E.2: Timings: our IPFE from HSM and *vs.* [ALS16]'s IPFE from DCR

	$\lambda=112, \ell=10$		$\lambda=128, \ell=10$	
	this work	[ALS16]	this work	[ALS16]
secret key bitsize	1920	24592	2340	36876
encryption time	40ms	27**ms**	78**ms**	85ms
decryption time	110**ms**	301ms	193**ms**	964ms

For all parameters our dependency in ℓ is linear which allows to extrapolate timings for $\ell > 10$.

We performed timings with Sage 8.1 on a standard laptop with a straight-forward implementation. Using the settings of [CL15], the exponentiation in class groups uses a PARI/GP function (qfbnupow), which is far less optimised than the exponentiation in $\mathbf{Z}/N\mathbf{Z}$, implying a huge bias in favour of Paillier. Despite this bias, the efficiency improvement we expected from our protocols is reflected in practice, as showed in Table E.2. We gain firstly from the fact that we can use smaller parameters for the same security level and secondly, because our security reductions allow to replace N^ℓ with p^ℓ in the derived keys. Thus the gain is not only in the constants and our scheme becomes more and more interesting as the security level and the dimension ℓ increase.

Acknowledgements:

The authors would like to thank both Benoît Libert and Damien Stehlé for fruitful discussions. This work was supported by the French ANR ALAMBIC project (ANR-16-CE39-0006), and by ERC Starting Grant ERC-2013-StG-335086-LATTAC.

E.A. Proofs for background lemmas on Gaussian distributions

We here provide proofs for two of the lemmas on Gaussian distributions stated in Section E.2. For ease of reading we here recall the lemmas before providing each of their respective proofs.

Lemma E − 1. Let $\vec{x} \in \mathbf{R}^\ell \setminus \{\vec{0}\}$, $\vec{c} \in \mathbf{R}^\ell$, $\sigma \in \mathbf{R}$ with $\sigma > 0$ and $\sigma' = \sigma/\|\vec{x}\|_2$, $c' = \frac{\langle\vec{c},\vec{x}\rangle}{\langle\vec{x},\vec{x}\rangle}$. A random variable K is distributed according to $\mathscr{D}_{\mathbf{Z},\sigma',c'}$ if and only if $V := K\vec{x}$ is distributed according to $\mathscr{D}_{\vec{x}\mathbf{Z},\sigma,\vec{c}}$.

Proof. Let $k \in \mathbf{Z}$, and $\vec{v} := k\vec{x} \in \vec{x}\mathbf{Z}$, then

$$\Pr[V = \vec{v}] = \Pr[V = k\vec{x}] = \Pr[K = k] = \frac{\rho_{\sigma',c'}(k)}{\rho_{\sigma',c'}(\mathbf{Z})}.$$

As in the proof of [GPV08, Lemma 4.5], one can compute

$$\rho_{\sigma',c'}(k) = \rho_\sigma((k - c')\|\vec{x}\|_2) = \rho_\sigma((k - c')\vec{x}) = \rho_\sigma(\vec{v} - c'\vec{x}).$$

It holds that $\vec{u} := c'\vec{x}$ is the orthogonal projection of $\vec{c}$ on $\vec{x}\mathbf{R}$. By Pythagoras' Theorem,

$$\|\vec{v} - \vec{c}\|_2^2 = \|\vec{v} - \vec{u}\|_2^2 + \|\vec{c} - \vec{u}\|_2^2.$$

Thus $\rho_\sigma(\vec{v}-c'\vec{x}) = \rho_\sigma(\|\vec{v}-\vec{u}\|_2) = \rho_\sigma(\|\vec{v}-\vec{c}\|_2)\times C$ where $C = \exp(\frac{\pi\|\vec{c}-\vec{u}\|_2^2}{\sigma^2})$ is a constant. Therefore we have demonstrated that for $k \in \mathbf{Z}$, $\vec{v} = k\vec{x}$, $\rho_{\sigma',c'}(k) = \rho_{\sigma,\vec{c}}(\vec{v}) \times C$. And so:

$$\Pr[V = \vec{v}] = \frac{\rho_{\sigma,\vec{c}}(\vec{v}) \times C}{\sum_{z\in\mathbf{Z}} \rho_{\sigma,\vec{c}}(z\vec{x}) \times C} = \mathscr{D}_{\vec{x}\mathbf{Z},\sigma,\vec{c}}.$$

□

```
Algorithm KeyGen($1^\lambda$)
  1. $(p, \tilde{s}, g, f, G, F) \leftarrow \mathsf{Gen}(1^\lambda, 1^\mu)$
  2. Pick $x \hookleftarrow \mathscr{D}_{\mathbf{Z},\sigma}$ and set $h = g^x$
  3. Set $pk = (p, \tilde{s}, g, h, f)$ and $sk = x$.
  4. Return $(pk, sk)$
```

```
Algorithm Decrypt($1^\lambda, pk, sk, (c_1, c_2)$)
  1. Compute $M = c_2 / c_1^x$
  2. $m \leftarrow \mathsf{Solve}(p, g, f, G, F, M)$
  3. Return $m$
```

```
Algorithm Encrypt($1^\lambda, pk, m$)
  1. Pick $r \hookleftarrow \mathscr{D}_{\mathbf{Z},\sigma}$
  2. Compute $c_1 = g^r$
  3. Compute $c_2 = f^m h^r$
  4. Return $(c_1, c_2)$
```

Figure E.7: The CL linearly homomorphic encryption scheme

Lemma E − 2. Let $\vec{x} \in \mathbf{R}^\ell$ with $\vec{x} \neq \vec{0}$, $\vec{c} \in \mathbf{R}^\ell$, $\sigma \in \mathbf{R}$ with $\sigma > 0$. Let V be a random variable distributed according to $\mathscr{D}_{\vec{x}\cdot\mathbf{Z},\sigma,\vec{c}}$. Then the random variable S defined as $S = \langle \vec{x}, V \rangle$ is distributed according to $\mathscr{D}_{\|\vec{x}\|_2^2\cdot\mathbf{Z},\sigma\cdot\|\vec{x}\|_2,\langle\vec{c},\vec{x}\rangle}$.

Proof. As V is distributed according to $\mathscr{D}_{\vec{x}\cdot\mathbf{Z},\sigma,\vec{c}}$, we have $V = K\vec{x}$ where K is sampled from $\mathscr{D}_{\mathbf{Z},\sigma/\|x\|_2,c'}$ where $c' = \frac{\langle\vec{c},\vec{x}\rangle}{\langle\vec{x},\vec{x}\rangle}$ from the previous lemma. As a result, one can write $S = K\langle\vec{x},\vec{x}\rangle$, and applying the previous lemma another time in dimension 1, we get that S is distributed according to $\mathscr{D}_{\|x\|_2^2\cdot\mathbf{Z},\sigma\cdot\|\vec{x}\|_2,\langle\vec{c},\vec{x}\rangle}$. □

E.B. Description of the original CL protocol

From a DDH group with an easy DL subgroup, Castagnos and Laguillaumie proposed a generic framework to design a linearly homomorphic encryption scheme. An Elgamal type scheme is used in G, with plaintext message $m \in \mathbf{Z}/p\mathbf{Z}$ mapped to $f^m \in F$. The resulting scheme is linearly homomorphic. Thanks to the Solve algorithm, the decryption does not need a complex DL computation. We depict this scheme in Fig. E.7. The Gen and Solve algorithms are those of Def. E − 6 except that we ignore the group G^p and its generator (which are useless here). From Lemma E − 4, Item 2, choosing $\sigma > \tilde{s}p\sqrt{\lambda}$ suffices to ensure that the distribution $\{g^x, x \hookleftarrow \mathscr{D}_{\mathbf{Z},\sigma}\}$ is at distance less than $2^{-\lambda}$ from the uniform distribution in G. Note that the use of Gaussian sampling instead of uniform was suggested in [CIL17].

We now recall the main assumption of [CL15].

Definition E − 9 (DDH assumption [CL15]). We say that GenGroup is the generator of a DDH group with easy DL subgroup F if it holds that the DDH problem is hard in

G even with access to the Solve algorithm. More precisely, let $\mathcal{D}$ be a distribution over the integers such that the distribution over G induced by $\{g^x; x \hookleftarrow \mathcal{D}\}$ is at distance less than $2^{-\lambda}$ from the uniform distribution in G. Let $\mathcal{A}$ be an adversary for the DDH problem, its advantage is defined as:

$$\begin{aligned}\mathsf{Adv}_{\mathcal{A}}^{\mathsf{DDH}}(\lambda,\mu) = \Big| 2\cdot \Pr\big[b = b^\star : (p,\tilde{s},g,f,g_p,G,F,G^p) \leftarrow \mathsf{Gen}(1^\lambda,1^\mu),\\ x,y,z \hookleftarrow \mathcal{D}, X = g^x, Y = g^y, b \hookleftarrow \{0,1\}, Z_0 = g^z, Z_1 = g^{xy},\\ b^\star \leftarrow \mathcal{A}(p,\tilde{s},g,f,g_p,G,F,G^p,X,Y,Z_b,\mathsf{Solve}(.))\big] - 1\Big|\end{aligned}$$

The DDH problem is said to be hard in G if for all probabilistic polynomial time attacker $\mathcal{A}$, $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{DDH}}(\lambda,\mu)$ is negligible.

E.C. Proof of Lemma E − 4: chosing distributions $\mathcal{D}$ and $\mathcal{D}_p$

The first item is a consequence of Lemma C − 4 of [CL15]: it holds that the induced distribution on G is at distance less than $(p\cdot s)/(2^\lambda \tilde{s} p) \le 2^{-\lambda}$.

Item 2 follows from Lemma D − 1 of [CIL17], Castagnos et al. demonstrate that the choice of $\mathcal{D} = \mathcal{D}_{\mathbf{Z},\sigma}$ with $\sigma > \tilde{s}\cdot p\cdot\sqrt{\lambda} > \tilde{s}\cdot p\cdot\sqrt{\ln(2(1+2^{\lambda+1}))/\pi}$ induces a distribution over G at distance less than $2^{-\lambda}$ from the uniform distribution, and therefore trades a factor $2^{\lambda-1}$ for a factor $\sqrt{\lambda}$ compared to the previous choice. This also proves Item 3.

Since G^p is a subgroup of G, $\mathcal{D}_p$ can be defined from $\mathcal{D}$ as in Item 4: the distribution $\{g_p^x, x \hookleftarrow \mathcal{D}\}$ is statistically close to the uniform distribution in G^p.

Item 5 follows from the fact that $G = F \times G^p$ and the following lemma.

Lemma E − 7. *Let $G = \langle g\rangle$ be a cyclic group of order $n = p\cdot s$ with $\gcd(p,s) = 1$, $G^p = \langle g_p\rangle$ the subgroup of G of order s, and $F = \langle f\rangle$ the subgroup of G of order p. Let $\mathcal{D}_p$ be a distribution over the integers such that $\{g_p^x, x \hookleftarrow \mathcal{D}_p\}$ is at statistical distance δ_p of the uniform distribution over G^p. Then the distribution induced by $\{g_p^x\cdot f^a, x \hookleftarrow \mathcal{D}_p, a \xleftarrow{\$} \mathbf{Z}/p\mathbf{Z}\}$ is also at statistical distance δ_p from the uniform distribution over G.*

Proof. As $\gcd(p,s) = 1$, one has $G = G^p \times F$. Let us denote $\psi = (\psi_1,\psi_2)$ the induced isomorphism from G to $G^p \times F$. The probability that $\{g_p^x\cdot f^a, x \hookleftarrow \mathcal{D}_p, a \xleftarrow{\$} \mathbf{Z}/p\mathbf{Z}\}$ gives an element h of G, is $\Pr[g_p^x = \psi_1(h)]\cdot\Pr[f^a = \psi_2(h)] = 1/p \Pr[g_p^x = \psi_1(h)]$. As a result, the statistical distance to the uniform distribution in G is

$$\begin{aligned}\frac{1}{2}\sum_{h\in G}\left|\frac{1}{n} - \frac{1}{p}\cdot\Pr[g_p^x = \psi_1(h)]\right| &= \frac{1}{2}\cdot\frac{1}{p}\sum_{h\in G}\left|\frac{1}{s} - \Pr[g_p^x = \psi_1(h)]\right|\\ &= \frac{1}{2}\cdot\frac{1}{p}\cdot p\sum_{h_p\in G^p}\left|\frac{1}{s} - \Pr[g_p^x = h_p]\right| = \delta_p.\end{aligned}$$

□

E.D. Proof of Theorem E − 2: the scheme of Fig. E.2(a) is ind-cpa

The proof proceeds as a sequence of games, starting with the real ind-cpa game and ending in a game where the ciphertext statistically hides the random bit β chosen by the challenger. In Game i, we denote S_i the event $\beta = \beta'$.

Game 1

```
1. $(p, \tilde{s}, f, g_p, G, F, G^p) \leftarrow \mathsf{Gen}(1^\lambda, 1^\mu)$
2. Pick $x \hookleftarrow \mathscr{D}_{\mathbf{Z},p\sigma'}$ and $h = g_p^x$
3. Set $pk = (\tilde{s}, g_p, f, p, h)$ and $sk = x$
4. $m_0, m_1 \leftarrow \mathscr{A}(pk)$
5. Pick $\beta \hookleftarrow \{0,1\}$
6. Pick $r \hookleftarrow \mathscr{D}_{\mathbf{Z},\sigma'}$
7. Compute $c_1 = g_p^r \in G^p$
8. Compute $c_2 = c_1^x \cdot f^{m_\beta}$
9. $\beta' \leftarrow \mathscr{A}(pk, c_1, c_2)$
10. Return $(\beta = \beta')$
```

Game 2

```
1. $(p, \tilde{s}, g, f, g_p, G, F, G^p) \leftarrow \mathsf{Gen}(1^\lambda, 1^\mu)$
2. Pick $x \hookleftarrow \mathscr{D}_{\mathbf{Z},p\sigma'}$ and $h = g_p^x$
3. Set $pk = (\tilde{s}, g_p, f, p, h)$ and $sk = x$
4. $m_0, m_1 \leftarrow \mathscr{A}(pk)$
5. Pick $\beta \hookleftarrow \{0,1\}$
6. Pick $r \hookleftarrow \mathscr{D}_{\mathbf{Z},\sigma'}$ and $u \hookleftarrow \mathbf{Z}/p\mathbf{Z}$
7. Compute $c_1 = f^u \cdot g_p^r \in G$
8. Compute $c_2 = c_1^x \cdot f^{m_\beta}$
9. $\beta' \leftarrow \mathscr{A}(pk, c_1, c_2)$
10. Return $(\beta = \beta')$
```

Game 0 ⇒ Game 1: In Game 1 the challenger creates the secret key x from $\mathscr{D}_{\mathbf{Z},p\sigma'}$ instead of $\mathscr{D}_{\mathbf{Z},\sigma'}$. From Lemma E−4, Item 4, h is still at negligible distance of the uniform in G^p. Moreover, the challenger uses the secret key x to compute the ciphertext element $c_2 = f^{m_\beta} g_p^{xr} = f^{m_\beta} c_1^x$. These two changes do not impact the distribution of the public key and of the ciphertext, therefore the adversary's success probability in both games is identical, $Pr[S_0] = Pr[S_1]$.

Game 1 ⇒ Game 2: In Game 1, the distribution of c_1 is at negligible distance of the uniform distribution in G^p. Now, in Game 2, the challenger samples a random $u \hookleftarrow \mathbf{Z}/p\mathbf{Z}$ and computes the ciphertext element $c_1 = f^u \cdot g_p^r \in G$ where $r \hookleftarrow \mathscr{D}_{\mathbf{Z},\sigma'}$. This gives an element at negligible distance of the uniform distribution in G (*cf.* Lemma E−4, Item 5). Both games are indistinguishable under the HSM assumption. Therefore, $|Pr[S_2] - Pr[S_1]| \leq \mathsf{Adv}_{\mathscr{B}}^{\mathsf{HSM}}(\lambda, \mu)$.

Now in Game 2 we have $c_1 = f^u \cdot g_p^r \in G$ which information theoretically reveals u modulo p and r modulo s by using the fact that $G = F \times G^p$. We also have $c_2 = c_1^x f^{m_\beta} = g_p^{rx} f^{m_\beta + ux} = h^r f^{m_\beta + ux}$. For the adversary the value of h^r is fixed, so he can infer $m_\beta + ux \in \mathbf{Z}/p\mathbf{Z}$. Since u is sampled uniformly at random from $\mathbf{Z}/p\mathbf{Z}$, $u \neq 0 \mod p$ with all but negligible probability as p is a μ bit prime, with $\mu \geq \lambda$. Furthermore, as x is sampled from $\mathscr{D}_{\mathbf{Z},p\sigma'}$ with $p\sigma' > p\tilde{s}\sqrt{\lambda}$, the distribution of x modulo n is at negligible distance of the uniform modulo n (*cf.* Lemma E−4, Item 2). In particular, as $n = ps$ with

Game 1

```
1. $(pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda, 1^\mu)$
2. $m_0, m_1 \leftarrow \mathscr{A}(pk)$
3. Pick $\beta \hookleftarrow \{0,1\}$
4. Pick $r \hookleftarrow \mathscr{D}_{\mathbf{Z},\sigma}$
5. Compute $c_1 = g^r$
6. Compute $c_2 = h^r$
7. Compute $c_3 = c_1^x c_2^y f^{m_\beta}$
8. $\beta' \leftarrow \mathscr{A}(pk, c_1, c_2, c_3)$
9. Return $(\beta = \beta')$
```

Game 2

```
1. $(pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda, 1^\mu)$
2. $m_0, m_1 \leftarrow \mathscr{A}(pk)$
3. Pick $\beta \hookleftarrow \{0,1\}$
4. Pick $r \hookleftarrow \mathscr{D}_{\mathbf{Z},\sigma}$ and $u \hookleftarrow \mathbf{Z}/p\mathbf{Z}$
5. Compute $c_1 = g^r$
6. Compute $c_2 = h^r f^u$
7. Compute $c_3 = c_1^x c_2^y f^{m_\beta}$
8. $\beta' \leftarrow \mathscr{A}(pk, c_1, c_2, c_3)$
9. Return $(\beta = \beta')$
```

$\gcd(p,s) = 1$, x modulo p is at negligible distance of the uniform and is independent of x modulo s. So even if an unbounded adversary can learn x modulo s from h, x modulo p remains at negligible distance of the uniform from his point of view and $m_\beta + ux$ perfectly hides $m_\beta \in \mathbf{Z}/p\mathbf{Z}$. Therefore: $|\Pr[\mathsf{S}_2] - 1/2| \leq 2^{-\lambda}$. Combining the probability equations, we conclude the proof with the following inequality:

$$\mathsf{Adv}_{\mathscr{A}}^{\Pi_{E.2(a)}}(\lambda, \mu) \leq \mathsf{Adv}_{\mathscr{B}}^{\mathsf{HSM}}(\lambda, \mu) + 2^{-\lambda}$$

E.E. Proof of Theorem E − 3: the scheme of Fig. E.2(b) is ind-cpa

The proof proceeds as a sequence of games, starting with the real ind-cpa game and ending in a game where the ciphertext statistically hides the random bit β chosen by the challenger. In Game i, we denote S_i the event $\beta = \beta'$.

Game 0 ⇒ Game 1: In Game 1 the challenger uses the secret key x, y to compute the ciphertext element $c_3 = c_1^x c_2^y f^{m_\beta} = g^{rx} h^{ry} f^{m_\beta} = \eta^r f^{m_\beta}$. This change does not impact the distribution of the ciphertext, therefore the adversary's success probability in both games is identical, $\Pr[\mathsf{S}_0] = \Pr[\mathsf{S}_1]$.

Game 1 ⇒ Game 2: In Game 1, $(h = g^\alpha, c_1 = g^r, c_2 = h^r = g^{\alpha r})$ with $\alpha, r \hookleftarrow \mathscr{D}_{\mathbf{Z},\sigma}$ is a DH triplet. Now, in Game 2, the challenger samples a random $u \hookleftarrow \mathbf{Z}/p\mathbf{Z}$ and computes $c_2 = h^r f^u$. Both games are indistinguishable under the DDH-f assumption (*cf.* Def. E − 8). Therefore, $|\Pr[\mathsf{S}_2] - \Pr[\mathsf{S}_1]| \leq \mathsf{Adv}_{\mathscr{B}}^{\mathsf{DDH\text{-}f}}(\lambda, \mu)$.

Now in Game 2 we have $c_1 = g^r$ which information theoretically reveals r modulo n. Furthermore, $c_3 = c_1^x c_2^y f^{m_\beta} = \eta^r f^{m_\beta + uy}$. This information theoretically reveals $m_\beta + uy \in \mathbf{Z}/p\mathbf{Z}$ as the value of η^r is fixed from c_1. Since u is sampled uniformly at random from $\mathbf{Z}/p\mathbf{Z}$, $u \neq 0 \mod p$ with all but negligible probability as p is a μ bit prime, with $\mu \geq \lambda$.

As a result, we are interested in the distribution of y modulo p from the adversary's point of view.

The only information that $\mathscr{A}$ learns about y comes from c_3 and $\eta = g^x h^y$. This means that from η an unbounded adversary learns $z := x + \alpha y$ modulo n. Knowing z, the joint distribution of (x, y) modulo n is

$$(z - \alpha y \mod n, y \mod n) \text{ where } y \hookleftarrow \mathscr{D}_{\mathbf{Z},\sigma}.$$

As a result, knowing z, y modulo n is statistically close to the uniform distribution modulo n due to the choice of $\mathscr{D}_{\mathbf{Z},\sigma}$. Consequently for the adversary y modulo p is also statistically close to the uniform distribution modulo p and $m_\beta + uy \in \mathbf{Z}/p\mathbf{Z}$ perfectly hides $m_\beta \in \mathbf{Z}/p\mathbf{Z}$.

Therefore: $|\mathrm{P}r[\mathrm{S}_2] - 1/2| \leq 2^{-\lambda}$. Combining the probability equations, we conclude the proof with the following inequality:

$$\mathsf{Adv}_{\mathscr{A}}^{\Pi_{E.2(b)}}(\lambda, \mu) \leq \mathsf{Adv}_{\mathscr{B}}^{\mathsf{DDH\text{-}f}}(\lambda, \mu) + 2^{-\lambda}$$

Chapter F

Two-Party ECDSA from Hash Proof Systems and Efficient Instantiations

Joint work with Dario Catalano, Fabien Laguillaumie, Federico Savasta and Ida Tucker
[CCL+19]

Abstract. ECDSA is a widely adopted digital signature standard. Unfortunately, efficient distributed variants of this primitive are notoriously hard to achieve and known solutions often require expensive zero knowledge proofs to deal with malicious adversaries. For the two party case, Lindell [Lin17] recently managed to get an efficient solution which, to achieve simulation-based security, relies on an interactive, non standard, assumption on Paillier's cryptosystem. In this paper we generalize Lindell's solution using hash proof systems. The main advantage of our generic method is that it results in a simulation-based security proof without resorting to non-standard interactive assumptions.

Moving to concrete constructions, we show how to instantiate our framework using class groups of imaginary quadratic fields. Our implementations show that the practical impact of dropping such interactive assumptions is minimal. Indeed, while for 128-bit security our scheme is marginally slower than Lindell's, for 256-bit security it turns out to be better both in key generation and signing time. Moreover, in terms of communication cost, our implementation significantly reduces both the number of rounds and the transmitted bits without exception.

F.1. Introduction

Threshold cryptography [Des88, DF90, GJKR96, SG98, Sho00, Boy86, CH89, MR04a] allows n users to share a common key in such a way that any subset of t parties can use this key to decrypt or sign, while any coalition of less than t can do nothing. The key feature of this paradigm is that it allows to use the shared key without explicitly reconstructing it in the clear. This means a subset of t parties have to actively participate in the protocol whenever the secret key is used.

Applications of threshold cryptography range from contexts where many signers need to agree to sign one common document to distributed scenarios where sensitive documents should become accessible only by a quorum. This versatility sparked intense research efforts that, mainly in the decade from the early 1990s to the early 2000s, produced efficient threshold versions of most commonly used cryptographic schemes. Recent years have seen renewed interest in the field (e.g. [GGN16, Lin17, GG18, DKLs18, LN18, DKLs19]) for several reasons. First a number of start-up companies are using this technology to protect keys in real life applications [Ser, Unb, Sep]. Moreover, Bitcoin and other cryptocurrencies – for which security breaches can result in concrete financial losses – use ECDSA as underlying digital signature scheme. While multisignature-based countermeasures are built-in to Bitcoin, they offer less flexibility and introduce anonymity and scalability issues (see [GGN16]). Finally, some of the schemes developed twenty years ago are not as efficient as current applications want them to be. This is the case, for instance, for ECDSA/DSA signatures. Indeed, while for many other schemes fast threshold variants are known (e.g. RSA decryption/signing and ECIES decryption) constructing efficient threshold variant of these signatures proved to be much harder. The main reason for this unfair distribution seems to result from the inversion step that requires one to compute $k^{-1} \bmod q$ from an unknown k. To explain why this is the case, let us first briefly recall how ECDSA actually works[1]. The public key is an elliptic curve point Q and the signing key is x, such that $\mathrm{Q} \leftarrow x\mathrm{P}$, where P is a generator of the elliptic curve group of points of order q. To sign a message m one first hashes it using some suitable hash function H and then proceeds according to the following algorithm

1. Choose k random in $\mathbf{Z}/q\mathbf{Z}$
2. Compute $\mathrm{R} \leftarrow k\mathrm{P}$
3. Let $r \leftarrow r_x \bmod q$ where $\mathrm{R} = (r_x, r_y)$
4. Set $s \leftarrow k^{-1}(\mathrm{H}(m) + rx) \bmod q$
5. Output (r, s)

Now, the natural approach to make the above algorithm distributed would be to share x additively among the participants and then start a multiparty computation protocol to produce the signature. In the two party case, this means that players start with

[1] From now on we will focus on the elliptic curve variant of the scheme, as this is the most commonly used scheme in applications. We stress that our reasoning apply to the basic DSA case as well.

shares x_1 and x_2 such that $\mathrm{Q} = (x_1 + x_2)\mathrm{P}$. The players can then proceed by generating random shares k_1, k_2 such that $\mathrm{R} = (k_1 + k_2)\mathrm{P}$. At this point, however, it is not clear how to compute, efficiently, shares k_1', k_2' such that $k_1' + k_2' = k'^{-1} \bmod q$.

Starting from [MR04a] two party ECDSA signature protocols started adopting a less common *multiplicative* sharing both for x and k. The basic idea of these constructions is very simple. Players start holding shares x_1, x_2 such that $\mathrm{Q} = x_1 x_2 \mathrm{P} = x\mathrm{P}$. Whenever a new signature has to be generated they generate random k_1, k_2 such that $\mathrm{R} = k_1 k_2 \mathrm{P} = k\mathrm{P}$. This immediately allows to get shares of the inverse k' as clearly $(k_1)^{-1}(k_2)^{-1} = (k_1 k_2)^{-1} \bmod q$. As a final ingredient, the parties use Paillier's homomorphic encryption to secretly add their shares and complete the signature. For instance, player P_1 computes $c_1 \leftarrow \mathsf{Enc}((k_1)^{-1}\mathrm{H}(m))$ and $c_2 \leftarrow \mathsf{Enc}((k_1)^{-1}x_1 r)$. P_2 can then complete the signature, using the homomorphic properties of the scheme as follows

$$c \leftarrow c_1^{k_2^{-1}} c_2^{k_2^{-1}x_2} = \mathsf{Enc}(k^{-1}\mathrm{H}(m))\mathsf{Enc}(k^{-1}xr) = \mathsf{Enc}(k^{-1}(\mathrm{H}(m) + xr))$$

P_2 concludes the protocol by sending back c to P_1. Now, if P_1 also knows the decryption key, he can extract the signature $s \leftarrow k^{-1}(\mathrm{H}(m) + xr))$ from c.

However, proving that each party followed the protocol correctly turns out to be hard. Initial attempts [MR04a] addressed this via expensive zero knowledge proofs. More recently Lindell in [Lin17] managed to provide a much simpler and efficient protocol. The crucial idea of Lindell's protocol is the observation that, in the above two party ECDSA signing protocol, dishonest parties can create very little trouble. Indeed, if in a preliminary phase P_2 receives both Paillier's encryption key *and* an encryption $\mathsf{Enc}(x_1)$ of P_1's share of the secret signing key, essentially, all a corrupted P_1 can do is participate in the generation of $\mathrm{R} \leftarrow k_1 k_2 \mathrm{P}$. Notice however that the latter is just the well established Diffie-Hellman protocol for which very efficient and robust protocols exist.

On the other hand, if P_2 is corrupted all she can do (except again participate in the generation of R) is to create a bad c as a final response for P_1. However, while P_2 can certainly try that, this would be easy to detect by simply checking the validity of the resulting signature.

Turning this nice intuition into a formal proof induces some caveats though. A first problem comes from the fact that Paillier's plaintexts space is $\mathbf{Z}/\mathrm{N}\mathbf{Z}$ (N is a large composite) whereas ECDSA signatures live in $\mathbf{Z}/q\mathbf{Z}$ (q is prime). Thus to avoid inconsistencies one needs to make sure that N is taken large enough so that no wraps around occur during the whole signature generation process. This also means that, when sending $\mathsf{Enc}(x_1)$ to P_2, P_1 needs to prove that the plaintext x_1 is in the right range (*i.e.*, sufficiently small).

A more subtle issue arises from the use of Paillier's encryption in the proof. Indeed, if one wants to use the scheme to argue indistinguishability of an adversary's view in real and simulated executions, it seems necessary to set up a reduction to the indistinguishability of Paillier's cryptosystem. This means one must design a proof technique that manages to successfully use Paillier's scheme *without* knowing the corresponding secret key. In Lindell's protocol the issue arises when designing the simulator's strategy against a corrupted player P_2. In such a case, P_2 might indeed send a wrong ciphertext c (*i.e.*, one that does not encrypt a signature) that the simulator simply cannot recognize as bad.

Lindell [Lin17] proposes two alternative proofs to overcome this. The first one relies on a game-based definition and avoids the problem by simply allowing the simulator to abort with a probability that depends on the number of issued signatures q_s. This results in a proof of security that is not tight (as the reduction actually looses a factor q_s). The second proof is simulation based, avoids the aborts, but requires the introduction of a new (interactive) non standard assumption regarding Paillier's encryption.

Thus, it is fair to say that, in spite of recent progress in the area, the following question remains open

Is it possible to devise a two party ECDSA signing protocol which is practical (both in terms of computational efficiency and in terms of bandwidth consumption), does not require interactive assumptions and allows for a tight security reduction?[2]

Our contribution

In this paper we provide a positive answer to the question above. In this sense, our contribution is twofold.

First, we provide a generic construction for two-party ECDSA signing from hash proof systems (HPS). Our solution can be seen as a generalization of Lindell's scheme [Lin17] to the general setting of HPSs that are homomorphic in the sense of [HO09]. This generic solution is not efficient enough for practical applications as, for instance, it employs general purpose zero knowledge as underlying building block. Still, beyond providing a clean, general framework which is of interest in its own right, it allows us to abstract away the properties we want to realize. In particular, our new protocol allows for a proof of security that is both tight and does not require artificial interactive assumptions when proving simulation security. Indeed, in encryption schemes based on HPSs, indistinguishability of ciphertexts is not compromised by the challenger knowing the scheme's secret keys as it relies on a computational assumption and a statistical argument.

The correctness of our protocol follows from homomorphic properties that we require of the underlying HPS. We define the notion of *homomorphically-extended projective hash families* which ensure the homomorphic properties of the HPS hold for any public key sampled from an efficiently recognisable set, thus no zero-knowledge proofs are required for the public key.

Towards efficient solutions, we then show how to instantiate our (homomorphic) HPS construction using class groups of imaginary quadratic fields. Although the devastating attack from [CL09] shows that large families of protocols built over such groups are insecure, Castagnos and Laguillaumie [CL15] showed that, if carefully designed, discrete logarithm based cryptosystems within such groups are still possible and allow for very efficient solutions. Algorithms to compute discrete logarithms in such groups

[2] We note here that the very recent two party protocol of [DKLs18] is very fast in signing time and only relies on the ECDSA assumption. However its bandwidth consumption is much higher than [Lin17].

have been extensively studied since the 80's and the best ones known to date have a subexponential complexity[3] of $\mathcal{O}(\mathrm{L}[1/2, o(1)])$ (compared to an $\mathcal{O}(\mathrm{L}[1/3, o(1)])$ complexity for factorisation or discrete logarithm computation in finite fields). In [BH03], Bauer and Hamdy also showed that, for the specific case of imaginary quadratic fields, better complexities seem unlikely. Thus, the resulting schemes benefit from (asymptotically) shorter keys. Moreover, interest in the area has recently been renewed as it allows versatile and efficient solutions such as encryption switching protocols [CIL17], inner product functional encryption [CLT18] or verifiable delay functions [BBBF18, Wes19].

Concretely, the main feature of the Castagnos and Laguillaumie cryptosystem and its variants (CL from now on) is that they rely on the existence of groups with associated easy discrete log subgroups, for which hard decision problems can be defined. More precisely, in [CL15] there exist a cyclic group $\mathrm{G} := \langle g \rangle$ of order qs where s is unknown, q is prime and $\gcd(q, s) = 1$, and an associated cyclic subgroup of order q, $\mathrm{F} := \langle f \rangle$. Denoting with $\mathrm{G}^q := \langle g_q \rangle$ the subgroup of q-th powers in G (of unknown order s), one has $\mathrm{G} = \mathrm{F} \times \mathrm{G}^q$, and one can define an hard subgroup membership problem. Informally, and deferring for later the necessary mathematical details, this allows to build a linearly homomorphic encryption scheme where the plaintext space is $\mathbf{Z}/q\mathbf{Z}$ for arbitrarily large q. This also means that if one uses the very same q underlying the ECDSA signature, one gets a concrete instantiation of our general protocol which naturally avoids all the inefficiencies resulting from N and q being different!

We remark that, similarly to Lindell's solution, our schemes require P_2 to hold an encryption $\mathsf{Enc}(x_1)$ of P_1's share of the secret key. As for Lindell's case, this imposes a somewhat heavy key registration phase in which P_1 has to prove, among other things, that the public key is correctly generated. While, in our setting we can achieve this without resorting to expensive range proofs, difficulties arise from the fact that (1) we work with groups of unknown order and (2) we cannot assume that all ciphertexts are valid (*i.e.*, actually encrypt a message)[4]. We address this by developing a new proof that solves both issues at the same time. Our proof is inspired by the Girault *et al.* [GPS06] identification protocol but introduces new ideas to adapt it to our setting and to make it a proof of knowledge. As for Lindell's case, it uses a binary challenge, which implies that the proof has to be repeated t times to get soundness error 2^t. We believe that it should be possible to enlarge the challenge space using techniques similar to those [CKY09] adapted to work in the context of class groups. Exploring the actual feasibility of this idea is left as a future work. Clearly, advances in this direction would lead to substantial efficiency improvements.

As final contribution, we propose a C implementation of our protocol[5]. Our results show that our improved security guarantees come almost at no additional cost. Indeed, while our scheme is slightly slower (by a factor 1.5 for key generation and 3.5 for signing) for 128-bit security level, we are actually better for larger parameters: for 256-bit security, we are more efficient both in terms of key generation and signing time (by respective factors of 4.2 and 1.3). Intuitively, this behavior is due to the fact that our interactive

[3] $\mathrm{L}[\alpha, c]$ denotes $\mathrm{L}_{\alpha,c}(x) := \exp(c \log(x)^{\alpha} \log(\log(x))^{1-\alpha})$

[4] For Paillier's scheme, used in [Lin17], this is not an issue: every ciphertext is valid

[5] We also re-implemented Lindell's protocol to ensure a fair comparison

key generation requires fewer exponentiations than that of Lindell's protocol (160 vs. 360), but an exponentiation in a class group is more expensive than an exponentiation in $\mathbf{Z}/n\mathbf{Z}$. The effect of the $L_{1/2}$ complexity and the fewer number of exponentiations starts at 192 bit of security. In terms of bandwidth, our protocol dramatically improves the communication cost by *factors varying from 5 (112 bit security) to 10 (256 bit security)* for key generation, and from 1.2 to 2.1 for signatures. It reduces the number of rounds from 175 (in Lindell's protocol) to 126 for the key generation process (the two signatures have the same number of rounds). We refer to Section F.5 for precise implementation considerations and timings.

As a final remark, our HPS methods also allow a concrete implementation based on Paillier's decisional composite residuosity assumption, competitive with Lindell's for 112 and 128 bits of security as detailed in Appendix F.6.

F.2. Preliminaries

Notations.

For a distribution $\mathscr{D}$, we write $d \hookleftarrow \mathscr{D}$ to refer to d being sampled from $\mathscr{D}$ and $b \overset{\$}{\leftarrow} B$ if b is sampled uniformly in the set B. In an interactive protocol IP, between parties P_1 and P_2, we denote by $\mathsf{IP}\langle x_1; x_2\rangle \rightarrow \langle y_1; y_2\rangle$ the joint execution of parties $\{P_i\}_{i\in\{1,2\}}$ in the protocol, with respective inputs x_i, and where P_i's private output at the end of the execution is y_i.

The elliptic curve digital signature algorithm.

ECDSA is the elliptic curve analogue of the Digital Signature Algoritm (DSA). It was put forth by Vanstone [Van92] and accepted as ISO, ANSI, IEEE and FIPS standards. It works in a group $(\mathbb{G}, +)$ of prime order q (of say μ bits) of points of an elliptic curve over a finite field, generated by P and consists of the following algorithms.

$\mathsf{KeyGen}(\mathbb{G}, q, P) \rightarrow (x, Q)$ where $x \overset{\$}{\leftarrow} \mathbf{Z}/q\mathbf{Z}$ is the secret signing key and $Q \leftarrow xP$ is the public verification key.

$\mathsf{Sign}(x, m) \rightarrow (r, s)$ where r and s are computed as follows:

1. Compute m': the μ leftmost bits of $\mathsf{SHA256}(m)$ where m is to be signed.
2. Sample $k \overset{\$}{\leftarrow} (\mathbf{Z}/q\mathbf{Z})^*$ and compute $R \leftarrow kP$; denote $R = (r_x, r_y)$ and let $r \leftarrow r_x \mod q$. If $r = 0$ chose another k.
3. Compute $s \leftarrow k^{-1} \cdot (m' + r \cdot x) \mod q$

$\mathsf{Verif}(Q, m, (r, s)) \rightarrow \{0, 1\}$ indicating whether or not the signature is accepted.

Two-party ECDSA.

This consists of the following interactive protocols:

$\mathsf{IKeyGen}\langle(\mathbb{G},q,P);(\mathbb{G},q,P)\rangle \rightarrow \langle(x_1,Q);(x_2,Q)\rangle$ such that $\mathsf{KeyGen}(\mathbb{G},q,P) \rightarrow (x,Q)$, where x_1 and x_2 two shares of x.

$\mathsf{ISign}\langle(x_1,m);(x_2,m)\rangle \rightarrow \langle\varnothing;(r,s)\rangle$ or $\langle(r,s);\varnothing\rangle$ or $\langle(r,s);(r,s)\rangle$ denoting by $\varnothing$ the empty output, signifying that one of the parties may have no output and $\mathsf{Sign}(x,m) \rightarrow (r,s)$.

The verification algorithm is non interactive and identical to that of ECDSA.

Interactive zero-knowledge proof systems.

A zero-knowledge proof system (P,V) for a language $\mathscr{L}$ is an interactive protocol between two probabilistic algorithms: a prover P and a polynomial-time verifier V. Informally P, detaining a witness for a given statement, must convince V that it is true without revealing anything other to V. A more formal definition is provided in Appendix F.A.

Simulation-based security and ideal functionalities.

In order to prove a protocol is secure, one must first define what *secure* means. Basically, the Ideal/Real paradigm is to imagine what properties one would have in an ideal world; then if a real world (constructed) protocol has similar properties it is considered secure. We consider static adversaries, that choose which parties are corrupted before the protocol begins. For a detailed explanation of the simulation paradigm we refer the reader to [Lin16].

- Upon receiving (prove,sid,x,w) from a party P_i (for $i \in \{1,2\}$): if $(x,w) \notin \mathsf{R}$ or sid has been previously used then ignore the message. Otherwise, send (proof,sid,x) to party P_{3-i}

Figure F.1: The $\mathscr{F}_{\mathsf{zk}}^{\mathsf{R}}$ functionality

We will use ideal functionalities for commitments, zero-knowledge proofs of knowledge (ZKPoK) and commitments to non interactive zero-knowledge (NIZK) proofs of knowledge between two parties P_1 and P_2. We give the intuition behind these ideal functionalities with the example of ZKPoK. We consider the case of a prover P_i with $i \in \{1,2\}$ who wants to prove the knowledge of a witness w for an element x which ensures that (x,w) satisfy the relation R, *i.e.* $(x,w) \in \mathsf{R}$.

In an ideal world we can imagine an honest and trustful third party, which can communicate with both P_i and P_{3-i}. In this ideal scenario, P_i could give (x,w) to this trusted party, the latter would then check if $(x, w) \in \mathsf{R}$ and tell P_{3-i} if this is true or false. In the real world we do not have such trusted parties and must substitute them with a cryptographic protocol between P_1 and P_2. Roughly speaking, the Ideal/Real paradigm requires that whatever information an adversary $\mathscr{A}$ (corrupting either P_1 or P_2) could recover in the real world, it can also recover in the ideal world.

The trusted third party can be viewed as the ideal functionality and we denote it by $\mathscr{F}$. If some protocol satisfies the above property regarding this functionality, we call it secure.

Formally, we denote $\mathscr{F}\langle x_1; x_2\rangle \rightarrow \langle y_1; y_2\rangle$ the joint execution of the parties via the functionality $\mathscr{F}$, with respective inputs x_i, and respective private outputs at the end of the execution y_i. Each transmitted message is labelled with a session identifier sid, which identifies an iteration of the functionality. The *ideal ZKPoK functionality* [HL10, Section 6.5.3], denoted $\mathscr{F}_{\mathsf{zk}}$, is defined for a relation R by $\mathscr{F}_{\mathsf{zk}}\langle(x, w); \varnothing\rangle \rightarrow \langle\varnothing; (x, \mathsf{R}(x, w))\rangle$, where $\varnothing$ is the empty output, signifying that the first party receives no output (cf. Fig. F.1).

- Upon receiving $(\mathsf{commit}, sid, x)$ from party P_i (for $i \in \{1, 2\}$), record (sid, i, x) and send $(\mathsf{receipt}, sid)$ to party P_{3-i}. If some $(\mathsf{commit}, sid, *)$ is already stored, then ignore the message.
- Upon receiving $(\mathsf{decommit}, sid)$ from party P_i , if (sid, i, x) is recorded then send $(\mathsf{decommit}, sid, x)$ to party P_{3-i}.

Figure F.2: The $\mathscr{F}_{\mathsf{com}}$ functionality

The *ideal commitment functionality*, denoted $\mathscr{F}_{\mathsf{com}}$, is depicted in Fig. F.2. We also use an ideal functionality $\mathscr{F}^{\mathsf{R}}_{\mathsf{com-zk}}$ for *commitments to NIZK proofs* for a relation R (cf. Fig. F.3). Essentially, this is a commitment functionality, where the committed value is a NIZK proof.

- Upon receiving $(\mathsf{com}-\mathsf{prove}, sid, x, w)$ from a party P_i (for $i \in \{1, 2\}$): if $(x, w) \notin \mathsf{R}$ or sid has been previously used then ignore the message. Otherwise, store (sid, i, x) and send $(\mathsf{proof}-\mathsf{receipt}, sid)$ to P_{3-i}.
- Upon receiving $(\mathsf{decom}-\mathsf{proof}, sid)$ from a party P_i (for $i \in \{1, 2\}$): if (sid, i, x) has been stored then send $(\mathsf{decom}-\mathsf{proof}, sid, x)$ to P_{3-i}

Figure F.3: The $\mathscr{F}^{\mathsf{R}}_{\mathsf{com-zk}}$ functionality

The ideal functionality for two-party ECDSA.

The ideal functionality $\mathscr{F}_{\mathrm{ECDSA}}$ (cf. Fig. F.4) consists of two functions: a key generation function, called once, and a signing function, called an arbitrary number of times with the generated keys.

Consider an Elliptic-curve group $\mathbb{G}$ of order q with generator a point P, then:

- Upon receiving $\mathsf{KeyGen}(\mathbb{G}, \mathrm{P}, q)$ from both P_1 and P_2:

    1. Generate an ECDSA key pair (Q, x), where $x \xleftarrow{\$} (\mathbf{Z}/q\mathbf{Z})^*$ is chosen randomly and Q is computed as $\mathrm{Q} \leftarrow x \cdot \mathrm{P}$.
    2. Choose a hash function $\mathrm{H}_q : \{0,1\}^* \to \{0,1\}^{\lfloor \log |q| \rfloor}$, and store $(\mathbb{G}, \mathrm{P}, q, \mathrm{H}_q, x)$.
    3. Send Q (and H_q) to both P_1 and P_2.
    4. Ignore future calls to KeyGen.

- Upon receiving $\mathsf{Sign}(sid, m)$ from both P_1 and P_2, where keys have already been generated from a call to Keygen and sid has not been previously used, compute an ECDSA signature (r, s) on m, and send it to both P_1 and P_2.

    To do this, choose a random $k \xleftarrow{\$} (\mathbf{Z}/q\mathbf{Z})^*$, compute $(r_x, r_y) \leftarrow k \cdot \mathrm{P}$ and set $r \leftarrow r_x \mod q$. Finally, compute $s \leftarrow k^{-1}(\mathrm{H}_q(m) + rx)$ and output (r, s).

Figure F.4: The $\mathscr{F}_{\mathrm{ECDSA}}$ functionality

F.3. Two-Party ECDSA from Hash Proof Systems

In this section we provide a generic construction for two-party ECDSA signing from hash proof systems (HPS) which we prove secure in the simulation based model. In the following, we consider the group of points of an elliptic curve $\mathbb{G}$ of order q, generated by P. In Subsection F.3.1, we first recall some basic definitions on the HPS framework from [CS02], before defining the specific properties we require for our construction in Subsection F.3.2, in particular, to guarantee correctness of the protocol (in order for party P_2 to be able to perform homomorphic operations on ciphertexts provided by P_1, which are encryptions of elements in $\mathbf{Z}/q\mathbf{Z}$) the HPS must be homomorphic; and for security to hold against malicious adversaries we also require that the subset membership problem underlying the HPS be hard, and that the HPS be smooth. We note that diverse group systems (often used as a foundation for constructions of HPSs) imply all the aforementioned properties. Such HPSs define linearly homomorphic encryption schemes as described in Subsection F.3.3. Finally, before presenting the overall two party signing protocol and proving its security, we describe the zero-knowledge proofs

(ZKP) related to the aforementioned HPSs, and justify that they fulfil the $\mathscr{F}_{\mathsf{com}}/\mathscr{F}_{\mathsf{com-zk}}$ hybrid model.

F.3.1. Background on Hash Proof Systems

Hash proof systems were introduced in [CS02] as a generalisation of the techniques used in [CS98] to design chosen ciphertext secure public-key encryption schemes. Consider a set of words $\mathscr{X}$, an NP language $\mathscr{L} \subset \mathscr{X}$ such that $\mathscr{L} := \{x \in \mathscr{X} | w \in \mathscr{W} : (x, w) \in \mathsf{R}\}$ where R is the relation defining the language, $\mathscr{L}$ is the language of true statements in $\mathscr{X}$, and for $(x, w) \in \mathsf{R}$, $w \in \mathscr{W}$ is a witness for $x \in \mathscr{L}$. The set $(\mathscr{X}, \mathscr{L}, \mathscr{W}, \mathsf{R})$ defines an instance of a subset membership problem, *i.e.*, the problem of deciding if an element $x \in \mathscr{X}$ is in $\mathscr{L}$ or in $\mathscr{X} \backslash \mathscr{L}$.

A HPS associates a projective hash family (PHF) to such a subset membership problem. The PHF defines a key generation algorithm $\mathsf{PHF.KeyGen}$ which outputs a secret hashing key hk sampled from distribution of hashing keys $\mathscr{D}_{\mathsf{hk}}$ over a hash key space K_{hk}, and a public projection key $\mathsf{hp} \leftarrow \alpha(\mathsf{hk})$ in projection key space K_{hp} (where $\alpha : \mathrm{K}_{\mathsf{hk}} \mapsto \mathrm{K}_{\mathsf{hp}}$ is an efficient auxiliary function). The secret hashing key hk defines a hash function $\mathscr{H}_{\mathsf{hk}} : \mathscr{X} \mapsto \Pi$, and hp allows for the (public) evaluation of the hash function on words $x \in \mathscr{L}$, *i.e.*, $\mathscr{H}_{\mathsf{hp}}(x, w) = \mathscr{H}_{\mathsf{hk}}(x)$ for $(x, w) \in \mathsf{R}$. A projective hash family PHF is thus defined by $\mathsf{PHF} := (\{\mathscr{H}_{\mathsf{hk}}\}_{\mathsf{hk} \in \mathrm{K}_{\mathsf{hk}}}, \mathrm{K}_{\mathsf{hk}}, \mathscr{X}, \mathscr{L}, \Pi, \mathrm{K}_{\mathsf{hp}}, \alpha)$.

F.3.2. Required Properties

δ_s-smoothness.

The standard *smoothness* property of a PHF requires that for any $x \notin \mathscr{L}$, the value $\mathscr{H}_{\mathsf{hk}}(x)$ be uniformly distributed knowing hp. In this work messages will be encoded in a subgroup $\mathscr{F}$ of Π of order q, indeed, for integration with ECDSA it must hold that the group in which the message is encoded has order q, since the message space is dictated by the order of the elliptic curve group $\mathbb{G}$. In some instantiations $\mathscr{F} = \Pi$, but $\mathscr{F}$ may also be a strict subgroup of Π. For $m \in \mathbf{Z}/q\mathbf{Z}$ we denote $\mathsf{Encode}(m)$ the encoding of m in $\mathscr{F}$, where $\mathsf{Encode} : (\mathbf{Z}/q\mathbf{Z}, +) \mapsto (\mathscr{F}, \cdot)$ is an efficient isomorphism. We denote Decode the inverse isomorphism, which must also be efficiently computable.

If $\mathscr{F} \subsetneq \Pi$, we require smoothness over $\mathscr{X}$ on $\mathscr{F}$ [CS02, Subsection 8.2.4]. A projective hash family is δ_s-smooth over $\mathscr{X}$ on $\mathscr{F}$ if for any $x \in \mathscr{X} \backslash \mathscr{L}$, a random $\pi \in \mathscr{F}$ and a randomly sampled hashing key $\mathsf{hk} \hookleftarrow \mathscr{D}_{\mathsf{hk}}$, the distributions $\mathscr{U} := \{x, \alpha(\mathsf{hk}), \pi \cdot \mathscr{H}_{\mathsf{hk}}(x)\}$ and $\mathscr{V} := \{x, \alpha(\mathsf{hk}), \mathscr{H}_{\mathsf{hk}}(x)\}$ are δ_s-close.

$\delta_{\mathscr{L}}$-hard subset membership problem.

For security to hold $(\mathscr{X}, \mathscr{L}, \mathscr{W}, \mathsf{R})$ must be an instance of a hard subset membership problem, *i.e.*, no polynomial time algorithm can distinguish random elements of $\mathscr{X} \backslash \mathscr{L}$ from those of $\mathscr{L}$ with significant advantage. We say $(\mathscr{X}, \mathscr{L}, \mathscr{W}, \mathsf{R})$ is a $\delta_{\mathscr{L}}$-hard subset membership problem if $\delta_{\mathscr{L}}$ is the maximal advantage of any polynomial time adversary in distinguishing random elements of $\mathscr{X} \backslash \mathscr{L}$ from those of $\mathscr{L}$.

Linearly homomorphic PHF.

In order for the homomorphic operations performed by P_2 to hold in the two party ECDSA protocol, we require that the projective hash family also be homomorphic as defined in [HO09].

Definition F – 1 ([HO09]). The family $\mathsf{PHF} := (\{\mathscr{H}_{\mathsf{hk}}\}_{\mathsf{hk}\in\mathrm{K}_{\mathsf{hk}}}, \mathrm{K}_{\mathsf{hk}}, \mathscr{X}, \mathscr{L}, \Pi, \mathrm{K}_{\mathsf{hp}}, \alpha)$ is homomorphic if $(\mathscr{X}, \star)$ and $(\Pi, \cdot)$ are groups, and for all $\mathsf{hk} \in \mathrm{K}_{\mathsf{hk}}$, and $u_1, u_2 \in \mathscr{X}$, we have $\mathscr{H}_{\mathsf{hk}}(u_1) \cdot \mathscr{H}_{\mathsf{hk}}(u_2) = \mathscr{H}_{\mathsf{hk}}(u_1 \star u_2)$, that is to say $\mathscr{H}_{\mathsf{hk}}$ is a homomorphism for each hk.

This clearly implies that for $\mathsf{hp} \leftarrow \alpha(\mathsf{hk})$ the public projective hash function is linearly homomorphic with respect to elements $u_1, u_2 \in \mathscr{L}$.

Homomorphically extended PHF.

Note that the co-domain of α, which specifies the set of valid projection keys, may not be efficiently recognisable. Though we do not require – as did the protocol of [Lin17] – a costly ZKPoK of the secret key associated to the public key, it is essential in our protocol that even if a public key is chosen maliciously (*i.e.*, there does not exist $\mathsf{hk} \in \mathrm{K}_{\mathsf{hk}}$ such that $\mathsf{hp} \leftarrow \alpha(\mathsf{hk})$, which may go unnoticed to honest parties in the protocol), the homomorphic properties of the public projective hash function still hold. We thus require that the co-domain of α, which defines valid projection keys, be contained in an *efficiently recognisable* space $\mathrm{K}'_{\mathsf{hp}}$, such that for all $\mathsf{hp}' \in \mathrm{K}'_{\mathsf{hp}}$, $\mathscr{H}_{\mathsf{hp}'}$ is a homomorphism (respectively to its inputs in $\mathscr{L}$).

Definition F – 2 (Homomorphically extended PHF). We say that the family $\mathsf{PHF} := (\{\mathscr{H}_{\mathsf{hk}}\}_{\mathsf{hk}\in\mathrm{K}_{\mathsf{hk}}}, \mathrm{K}_{\mathsf{hk}}, \mathscr{X}, \mathscr{L}, \Pi, \mathrm{K}_{\mathsf{hp}}, \mathrm{K}'_{\mathsf{hp}}, \alpha)$ is homomorphically extended if $(\{\mathscr{H}_{\mathsf{hk}}\}_{\mathsf{hk}\in\mathrm{K}_{\mathsf{hk}}}, \mathrm{K}_{\mathsf{hk}}, \mathscr{X}, \mathscr{L}, \Pi, \mathrm{K}_{\mathsf{hp}}, \alpha)$ is a homomorphic PHF and that there exists an efficiently recognizable space $\mathrm{K}'_{\mathsf{hp}} \supseteq \mathrm{K}_{\mathsf{hp}}$ such that for any $\mathsf{hp}' \in \mathrm{K}'_{\mathsf{hp}}$, the projective hash function associated to hp' is a homomorphism (respectively to its inputs in $\mathscr{L}$).

ECDSA-friendly HPS.

We here define the notion of an ECDSA-friendly HPS, essentially it is a HPS which meets all of the aforementioned properties, and which suffices to ensure simulation based security in the protocol of Subsection F.3.5.

Definition F – 3 ($\delta_s/\delta_{\mathscr{L}}$-ECDSA-friendly HPS). Let $\mathscr{X}$, Π and $\mathscr{F}$ be groups such that $\mathscr{F}$ is a subgroup of Π of prime order q, and such that there exists an efficient isomorphism $\mathsf{Encode} : (\mathbf{Z}/q\mathbf{Z}, +) \mapsto (\mathscr{F}, \cdot)$, whose inverse Decode is also efficiently computable. A $\delta_s/\delta_{\mathscr{L}}$-ECDSA-friendly hash proof system is a hash proof system which associates to a $\delta_{\mathscr{L}}$-hard subset membership problem a homomorphically extended projective hash family $\mathsf{PHF} := (\{\mathscr{H}_{\mathsf{hk}}\}_{\mathsf{hk}\in\mathrm{K}_{\mathsf{hk}}}, \mathrm{K}_{\mathsf{hk}}, \mathscr{X}, \mathscr{L}, \Pi, \mathrm{K}_{\mathsf{hp}}, \mathrm{K}'_{\mathsf{hp}}, \alpha)$ which is δ_s–smooth over $\mathscr{X}$ on $\mathscr{F}$.

F.3.3. Resulting Encryption Scheme

We use the standard chosen plaintext attack secure encryption scheme which results from a HPS [CS02]. The key generation algorithm simply runs $\mathsf{PHF.KeyGen}$ and sets $\mathsf{hk} \in \mathrm{K}_{\mathsf{hk}}$ as the secret key, and the associated public key is $\mathsf{hp} \leftarrow \alpha(\mathsf{hk})$. Encryption of a plaintext message m in $\mathbf{Z}/q\mathbf{Z}$ is done by sampling a random pair $(u, w) \in \mathrm{R}$ and computing $\mathsf{Enc}(\mathsf{hp}, m) \leftarrow (u, \mathscr{H}_{\mathsf{hp}}(u, w) \cdot \mathsf{Encode}(m))$. To specify the randomness used in the encryption algorithm, we sometimes use the notation $\mathsf{Enc}((u, w); (\mathsf{hp}, m))$. To decrypt a ciphertext $(u, e) \in \mathscr{X} \times \Pi$ with secret key hk do: $\mathsf{Dec}(\mathsf{hk}, (u, e)) \leftarrow \mathsf{Decode}(\frac{e}{\mathscr{H}_{\mathsf{hk}}(u)})$. The scheme is indistinguishable under chosen plaintext attacks assuming both the smoothness of the HPS and the hardness of the underlying subset membership problem.

Homomorphic properties.

Given encryptions (u_1, e_1) and (u_2, e_2) of respectively m_1 and m_2, and an integer a, we require that there exist two procedures $\mathsf{EvalSum}$ and $\mathsf{EvalScal}$ such that

$$\mathsf{Dec}(\mathsf{hk}, \mathsf{EvalSum}(\mathsf{hp}, (u_1, e_1), (u_2, e_2))) = m_1 + m_2$$

and

$$\mathsf{Dec}(\mathsf{hk}, \mathsf{EvalScal}(\mathsf{hp}, (u_1, e_1), a)) = a \cdot m_1;$$

which is the case if the projective hash family is homomorphic.

F.3.4. Zero-Knowledge Proofs

Proofs of knowledge.

We use the $\mathscr{F}_{\mathsf{zk}}$, $\mathscr{F}_{\mathsf{com-zk}}$ hybrid model. Ideal ZK functionalities are used for the following relations, were the parameters of the elliptic curve $(\mathbb{G}, \mathrm{P}, q)$ are implicit public inputs:

1. $\mathrm{R}_{\mathsf{DL}} := \{(\mathrm{Q}, w) | \mathrm{Q} = w\mathrm{P}\}$, proves the knowledge of the discrete log of an elliptic curve point.
2. $\mathrm{R}_{\mathsf{HPS-DL}} := \{(\mathsf{hp}, (c_1, c_2), \mathrm{Q}_1); (x_1, w) | (c_1, c_2) = \mathsf{Enc}((u, w); (\mathsf{hp}, x_1)) \wedge (c_1, w) \in \mathrm{R} \wedge \mathrm{Q}_1 = x_1\mathrm{P}\}$, proves the knowledge of the randomness used for encryption, and of the value x_1 which is both encrypted in the ciphertext (c_1, c_2) and the discrete log of the elliptic curve point Q_1.

The functionalities $\mathscr{F}_{\mathsf{zk}}^{\mathrm{R}_{\mathsf{DL}}}$, $\mathscr{F}_{\mathsf{com-zk}}^{\mathrm{R}_{\mathsf{DL}}}$ can be instantiated using Schnorr proofs [Sch91a]. For the $\mathrm{R}_{\mathsf{HPS-DL}}$ proof, Lindell in [Lin17] uses a proof of language membership as opposed to a proof of knowledge. Though his technique is quite generic, it cannot be used in our setting. Indeed, his approach requires that the ciphertext be *valid*, which means that the element c must be decryptable. As Lindell uses Paillier's encryption scheme, any element of $(\mathbf{Z}/\mathrm{N}^2\mathbf{Z})^\times$ is a valid ciphertext. This is not the case for a HPS-based encryption scheme, since it incorporates redundancy so that any pair in $\mathscr{X} \times \Pi$ is not a valid ciphertext.

For our instantiations, we will introduce specific and efficient proofs. Note that in any case, we needn't prove that $x_1 \in \mathbf{Z}/q\mathbf{Z}$ since both the message space of our encryption scheme and the elliptic curve group $\mathbb{G}$ are of order q.

F.3.5. Two-Party ECDSA Signing Protocol with Simulation-Based Security

We here provide our generic construction for two-party ECDSA signing from hash proof systems (Figure F.5), along with a proof that the protocol is secure in the Ideal/Real paradigm (Theorem F − 1). To this end, we must argue the indistinguishability of an adversary's view – corrupting either party P_1 or P_2 – in real and simulated executions. In Cramer-Shoup like encryption schemes (resulting from HPSs as described in Subsection F.3.3), the chosen plaintext attack indistinguishability of ciphertexts allows for the challenger in the security game to sample the secret hashing key hk, and compute the resulting projection key hp. Thus hk is *known* to the challenger. Indeed here, in order to prove indistinguishability, the challenger first replaces the random masking element $u \in \mathscr{L}$ in the original encryption scheme with an element sampled outside the language $u' \in \mathscr{X} \backslash \mathscr{L}$. Note that in order to perform this change the challenger *must* know the secret hashing key. The hardness of the subset membership problem ensures this goes unnoticed to any polynomial time adversary. Then the smoothness of the projective hash family allows one to replace the plaintext value by some random element from the plaintext space, thus guaranteeing the indistinguishability of the resulting encryption scheme. We insist on this point since in Lindell's protocol [Lin17], many issues arise from the use of Paillier's cryptosystem, for which the indistinguishability of ciphertexts no longer holds if the challenger knows the secret key. In particular this implies that in Lindell's game based proof, instead of letting the simulator use the Paillier secret key to decrypt the incoming ciphertext (and check the corrupted party P_2 did not send a different ciphertext c than that prescribed by the protocol), the simulator *guesses* when the adversary may have cheated by simulating an abort with a probability depending on the number of issued signatures. This results in a proof of security which is not tight.

Moreover, though this technique suffices for a game-based definition, it does not for simulation-based security definitions. Thus, in order to be able to prove their protocol using simulation, they use a rather non-standard assumption, called Paillier-EC assumption and recalled in Appendix F.D. Thanks to the framework we have chosen to adopt, we are able to avoid such an artificial interactive assumption. Moreover, should one write a game based proof for our construction, the security loss present in [Lin17] would not appear.

Finally we note that the correctness of our protocol follows from the fact Encode is an efficient isomorphism, and from the fact the hash function is linearly homomorphic for any public key in the efficiently recognisable space $\mathrm{K}'_{\mathsf{hp}}$.

Theorem F − 1. *Assume* HPS *is a* $\delta_s/\delta_{\mathscr{L}}$*-ECDSA-friendly HPS then the protocol of Figure F.5 securely computes* $\mathscr{F}_{\mathsf{ECDSA}}$ *in the* $(\mathscr{F}_{\mathsf{zk}}, \mathscr{F}_{\mathsf{com-zk}})$*-hybrid model in the presence of a malicious static adversary (under the ideal/real definition). Indeed there exists a simulator for the scheme such that*

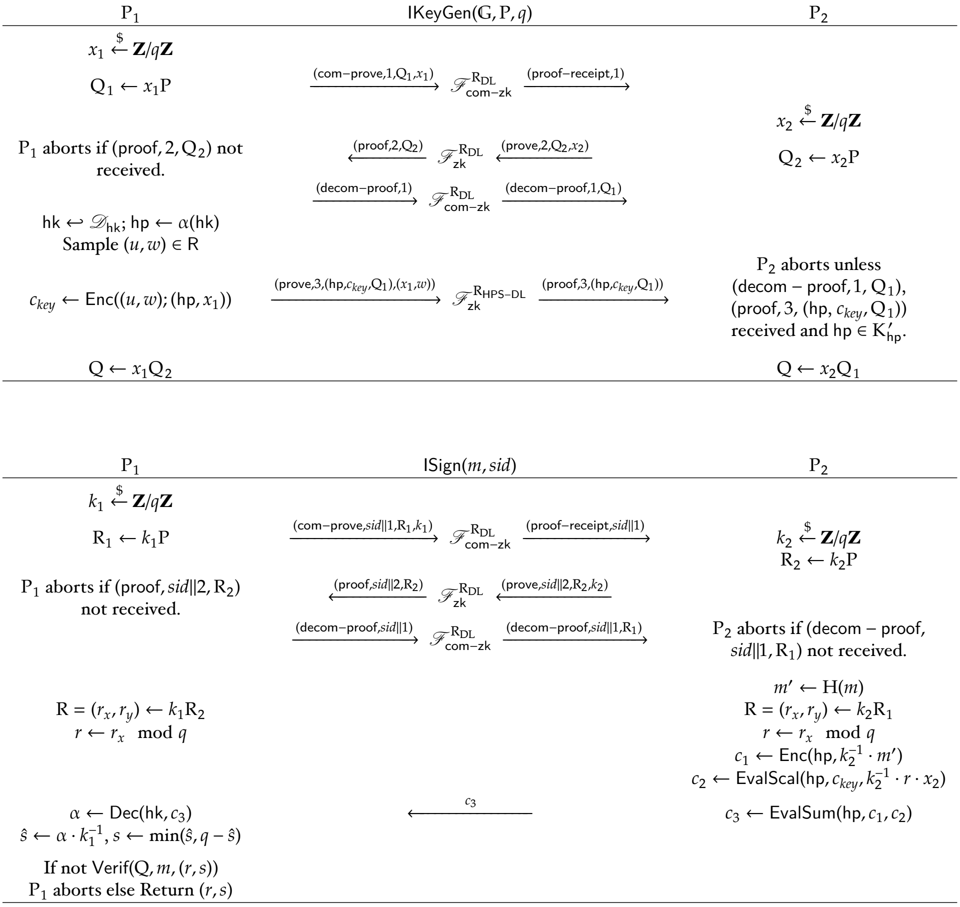

P_1
$\mathsf{IKeyGen}(\mathbb{G}, P, q)$
P_2
$x_1 \xleftarrow{\$} \mathbf{Z}/q\mathbf{Z}$
$Q_1 \leftarrow x_1 P$
(com−prove,1,Q_1,x_1)
$\mathscr{F}_{\text{com-zk}}^{R_{DL}}$
(proof−receipt,1)
$x_2 \xleftarrow{\$} \mathbf{Z}/q\mathbf{Z}$
$Q_2 \leftarrow x_2 P$
P_1 aborts if (proof, 2, Q_2) not received.
(proof,2,Q_2)
$\mathscr{F}_{\text{zk}}^{R_{DL}}$
(prove,2,Q_2,x_2)
(decom−proof,1)
$\mathscr{F}_{\text{com-zk}}^{R_{DL}}$
(decom−proof,1,Q_1)
$\mathsf{hk} \hookleftarrow \mathscr{D}_{\mathsf{hk}}$; $\mathsf{hp} \leftarrow \alpha(\mathsf{hk})$
Sample $(u, w) \in \mathsf{R}$
$c_{key} \leftarrow \mathsf{Enc}((u, w); (\mathsf{hp}, x_1))$
(prove,3,(hp,c_{key},Q_1),(x_1,w))
$\mathscr{F}_{\text{zk}}^{R_{\text{HPS-DL}}}$
(proof,3,(hp,c_{key},Q_1))
P_2 aborts unless
(decom − proof, 1, Q_1),
(proof, 3, (hp, c_{key}, Q_1))
received and hp $\in K'_{\mathsf{hp}}$.
$Q \leftarrow x_1 Q_2$
$Q \leftarrow x_2 Q_1$
P_1
$\mathsf{ISign}(m, sid)$
P_2
$k_1 \xleftarrow{\$} \mathbf{Z}/q\mathbf{Z}$
$R_1 \leftarrow k_1 P$
(com−prove,$sid\|1$,R_1,k_1)
$\mathscr{F}_{\text{com-zk}}^{R_{DL}}$
(proof−receipt,$sid\|1$)
$k_2 \xleftarrow{\$} \mathbf{Z}/q\mathbf{Z}$
$R_2 \leftarrow k_2 P$
P_1 aborts if (proof, $sid\|2$, R_2) not received.
(proof,$sid\|2$,R_2)
$\mathscr{F}_{\text{zk}}^{R_{DL}}$
(prove,$sid\|2$,R_2,k_2)
(decom−proof,$sid\|1$)
$\mathscr{F}_{\text{com-zk}}^{R_{DL}}$
(decom−proof,$sid\|1$,R_1)
P_2 aborts if (decom − proof, $sid\|1$, R_1) not received.
$m' \leftarrow H(m)$
$R = (r_x, r_y) \leftarrow k_1 R_2$
$r \leftarrow r_x \mod q$
$R = (r_x, r_y) \leftarrow k_2 R_1$
$r \leftarrow r_x \mod q$
$c_1 \leftarrow \mathsf{Enc}(\mathsf{hp}, k_2^{-1} \cdot m')$
$c_2 \leftarrow \mathsf{EvalScal}(\mathsf{hp}, c_{key}, k_2^{-1} \cdot r \cdot x_2)$
$\alpha \leftarrow \mathsf{Dec}(\mathsf{hk}, c_3)$
c_3
$c_3 \leftarrow \mathsf{EvalSum}(\mathsf{hp}, c_1, c_2)$
$\hat{s} \leftarrow \alpha \cdot k_1^{-1}$, $s \leftarrow \min(\hat{s}, q - \hat{s})$
If not $\mathsf{Verif}(Q, m, (r, s))$
P_1 aborts else Return (r, s)

Figure F.5: Two-Party ECDSA Key Generation and Signing Protocols from HPSs

no polynomial time adversary – having corrupted either P_1 *or* P_2 *– can distinguish a real execution of the protocol from a simulated one with probability greater than* $2\delta_{\mathscr{L}} + \delta_s$.

Proof. In this proof, the simulator $\mathscr{S}$ only has access to an ideal functionality $\mathscr{F}_{\text{ECDSA}}$ for computing ECDSA signatures, so all it learns in the ideal world is the public key Q which it gets as output of the KeyGen phase from $\mathscr{F}_{\text{ECDSA}}$ and signatures (r, s) for messages m of its choice as output of the Sign phase. However in the real world, the adversary, having either corrupted P_1 or P_2 will also see all the interactions with the non corrupted party which lead to the computation of a signature. Thus $\mathscr{S}$ must be ale to simulate $\mathscr{A}$'s view of these interactions, while only knowing the expected output. To this end $\mathscr{S}$ must set up with $\mathscr{A}$ the same public key Q that it received from $\mathscr{F}_{\text{ECDSA}}$, in order to be able to subsequently simulate interactively signing messages with $\mathscr{A}$, using the output of $\mathscr{F}_{\mathsf{ECDSA}}$ from the Sign phase.

$\mathscr{S}$ simulates P_2 – Corrupted P_1

We first show that if an adversary $\mathscr{A}_1$ corrupts P_1, one can construct a simulator $\mathscr{S}$ s.t. the output distribution of $\mathscr{S}$ is indistinguishable from $\mathscr{A}_1$'s view in an interaction with an honest party P_2. The main difference here with the proof of [Lin17] arises from the fact we no longer use a ZKP from which $\mathscr{S}$ can extract the encryption scheme's secret key. Instead, $\mathscr{S}$ extracts the randomness used for encryption and the plaintext x_1 from the ZKPoK for $R_{\mathsf{HPS-DL}}$, which allows it to recompute the ciphertext and verify it obtains the expected value c_{key}. Moreover since the message space of our encryption scheme is $\mathbf{Z}/q\mathbf{Z}$, if $\mathscr{A}_1$ does not cheat in the proofs (which is guaranteed by the $(\mathscr{F}_{\mathsf{zk}}, \mathscr{F}_{\mathsf{com-zk}})$-hybrid model), the obtained distributions are identical in the ideal and real executions (as opposed to statistically close as in [Lin17]).

Key Generation Phase

1. Given input $\mathsf{KeyGen}(\mathbb{G}, P, q)$, the simulator $\mathscr{S}$ sends $\mathsf{KeyGen}(\mathbb{G}, P, q)$ to the ideal functionality $\mathscr{F}_{\mathsf{ECDSA}}$ and receives back a public key Q.
2. $\mathscr{S}$ invokes $\mathscr{A}_1$ on input $\mathsf{IKeyGen}(\mathbb{G}, P, q)$ and receives the commitment to a proof of knowledge of x_1 such that $Q_1 = x_1 P$ denoted $(\mathsf{com-prove}, 1, Q_1, x_1)$ as $\mathscr{A}_1$ intends to send to $\mathscr{F}^{R_{\mathsf{DL}}}_{\mathsf{com-zk}}$, such that $\mathscr{S}$ can extract x_1 and Q_1.
3. Using the extracted value x_1, $\mathscr{S}$ verifies that $Q_1 = x_1 P$. If so, it computes $Q_2 \leftarrow x_1^{-1} Q$ (using the value Q received from $\mathscr{F}_{\mathsf{ECDSA}}$); otherwise $\mathscr{S}$ samples a random Q_2 from $\mathbb{G}$.
4. $\mathscr{S}$ sends $(\mathsf{proof}, 2, Q_2)$ to $\mathscr{A}_1$ as if sent by $\mathscr{F}^{R_{\mathsf{DL}}}_{\mathsf{zk}}$ thereby $\mathscr{S}$ simulating a proof of knowledge of x_2 s.t. $Q_2 = x_2 P$.
5. $\mathscr{S}$ receives $(\mathsf{decom-proof}, 1)$ as $\mathscr{A}_1$ intends to send to $\mathscr{F}^{R_{\mathsf{DL}}}_{\mathsf{com-zk}}$ and simulates P_2 aborting if $Q_1 \neq x_1 P$. $\mathscr{S}$ also receives $(\mathsf{prove}, 3, (\mathsf{hp}, c_{key}, Q_1), (x_1, w))$ as $\mathscr{A}_1$ intends to send to $\mathscr{F}^{R_{\mathsf{HPS-DL}}}_{\mathsf{zk}}$.
6. $\mathscr{S}$ computes u from w such that $(u, w) \in \mathsf{R}$, and using the extracted value x_1 verifies that $c_{key} = \mathsf{Enc}((u, w); (\mathsf{hp}, x_1))$, and simulates P_2 aborting if not.
7. $\mathscr{S}$ sends $\mathsf{continue}$ to $\mathscr{F}_{\mathsf{ECDSA}}$ for P_2 to receive output, and stores (x_1, Q, c_{key}).

When taking $\mathscr{F}_{\mathsf{zk}}$ and $\mathscr{F}_{\mathsf{com-zk}}$ as ideal functionalities, the only difference between the real execution as ran by an honest P_2, and the ideal execution simulated by $\mathscr{S}$ is that in the former $Q_2 \leftarrow x_2 P$ where $x_2 \overset{\$}{\leftarrow} \mathbf{Z}/q\mathbf{Z}$, whereas in the latter $Q_2 \leftarrow x_1^{-1} Q$, where Q is the public key returned by the ideal functionality $\mathscr{F}_{\mathsf{ECDSA}}$. However since $\mathscr{F}_{\mathsf{ECDSA}}$ samples Q uniformly at random from $\mathbb{G}$, the distribution of Q_2 in both cases is identical.

Signing Phase

1. Upon input $\mathsf{Sign}(sid, m)$, simulator $\mathscr{S}$ sends $\mathsf{Sign}(sid, m)$ to $\mathscr{F}_{\mathrm{ECDSA}}$ and receives back a signature (r, s).

2. $\mathscr{S}$ computes the elliptic curve point $\mathrm{R} = (r, r_y)$ using the ECDSA verification algorithm.

3. $\mathscr{S}$ invokes $\mathscr{A}_1$ with input $\mathsf{ISign}(sid, m)$ and simulates the first three interactions such that $\mathscr{A}_1$ computes R. The strategy is similar to that used to compute Q, in brief, it proceeds as follows:

   (a) $\mathscr{S}$ receives $(\mathsf{com}-\mathsf{prove}, sid\|1, \mathrm{R}_1, k_1)$ from $\mathscr{A}_1$.

   (b) If $\mathrm{R}_1 = k_1\mathrm{P}$ then $\mathscr{S}$ sets $\mathrm{R}_2 \leftarrow k_1^{-1}\mathrm{R}$; otherwise it chooses R_2 at random. $\mathscr{S}$ sends $(\mathsf{proof}, sid\|2, \mathrm{R}_2)$ to $\mathscr{A}_1$.

   (c) $\mathscr{S}$ receives $(\mathsf{decom}-\mathsf{proof}, sid\|1)$ from $\mathscr{A}_1$. If $\mathrm{R}_1 \neq k_1\mathrm{P}$ then $\mathscr{S}$ simulates P_2 aborting and instructs the trusted party computing $\mathscr{F}_{\mathrm{ECDSA}}$ to abort.

4. $\mathscr{S}$ computes $c_3 \leftarrow \mathsf{Enc}_{pk}(k_1 \cdot s \mod q)$, where s was received from $\mathscr{F}_{\mathrm{ECDSA}}$, and sends c_3 to $\mathscr{A}_1$.

As with the computation of Q_2 in the key generation phase, R_2 is distributed identically in the real and ideal executions since R is randomly generated by $\mathscr{F}_{\mathrm{ECDSA}}$. The zero-knowledge proofs and verifications are also identically distributed in the $\mathscr{F}_{\mathsf{zk}}$ and $\mathscr{F}_{\mathsf{com-zk}}$-hybrid model. Thus, the only difference between a real execution and the simulation is the way that c_3 is computed. In the simulation it is an encryption of $k_1 \cdot s = k_1 \cdot k^{-1}(m' + r \cdot x) = k_2^{-1} \cdot (m' + r \cdot x) \mod q$, whereas in a real execution c_3 is computed from c_{key}, using the homomorphic properties of the encryption scheme. However, notice that as long as there exist (u, w) such that $c_{key} = \mathsf{Enc}((u, w); (\mathsf{hp}, x_1))$ where $\mathrm{Q} = x_1\mathrm{P}$ – which is guaranteed by the ideal functionality $\mathscr{F}_{\mathsf{zk}}^{\mathrm{R}_{\mathsf{HPS-DL}}}$ – and as long as the homomorphic operations hold – which is guaranteed for any hp in the efficiently verifiable ensemble $\mathrm{K}'_{\mathsf{hp}}$ (cf. Subsection F.3.2) – the c_3 obtained in the real scenario is also an encryption of $s' = k_2^{-1} \cdot (m' + r \cdot x) \mod q$. Thus c_3 is distributed identically in both cases.

This implies that the view of a corrupted P_1 is identical in the real and ideal executions of the protocol (in the $\mathscr{F}_{\mathsf{zk}}$, $\mathscr{F}_{\mathsf{com-zk}}$-hybrid model), *i.e.*, the simulator perfectly simulates the real environment, which completes the proof of this simulation case.

$\mathscr{S}$ simulates P_1 – Corrupted P_2

We now suppose an adversary $\mathscr{A}_2$ corrupts P_2 and describe the ideal execution of the protocol. We demonstrate via a sequence of games – where the first game is a real execution and the last game is a simulated execution – that both executions are indistinguishable. This proof methodology differs considerably to that of [Lin17] since the main differences between a real and simulated execution are due to the ciphertext c_{key}, so the indistinguishability of both executions reduces to the ind-cpa security of the underlying encryption scheme. The necessity for the properties required of HPS will thus here become apparent. We first describe an ideal execution of the protocol:

Key Generation Phase

1. Given input $\mathsf{KeyGen}(\mathbb{G}, P, q)$, the simulator $\mathcal{S}$ sends $\mathsf{KeyGen}(\mathbb{G}, P, q)$ to the functionality $\mathcal{F}_{\mathsf{ECDSA}}$ and receives back Q.
2. $\mathcal{S}$ invokes $\mathcal{A}_2$ upon input $\mathsf{IKeyGen}(\mathbb{G}, P, q)$ and sends $(\mathsf{proof-receipt}, 1)$ as $\mathcal{A}_2$ expects to receive from $\mathcal{F}_{\mathsf{com-zk}}^{R_{\mathrm{DL}}}$.
3. $\mathcal{S}$ receives $(\mathsf{prove}, 2, Q_2, x_2)$ as $\mathcal{A}_2$ intends to send to $\mathcal{F}_{\mathsf{zk}}^{R_{\mathrm{DL}}}$.
4. Using the extracted value x_2, $\mathcal{S}$ verifies that Q_2 is a non zero point on the curve and that $Q_2 = x_2 P$. If so it computes $Q_1 \leftarrow (x_2)^{-1} Q$ and sends $(\mathsf{decom-proof}, 1, Q_1)$ to $\mathcal{A}_2$ as it expects to receive from $\mathcal{F}_{\mathsf{com-zk}}^{R_{\mathrm{DL}}}$. If not it simulates P_1 aborting and halts.
5. $\mathcal{S}$ samples $\mathsf{hk} \hookleftarrow \mathcal{D}_{\mathsf{hk}}$ and computes $\mathsf{hp} \leftarrow \alpha(\mathsf{hk})$. It also samples $\tilde{x}_1 \xleftarrow{\$} \mathbf{Z}/q\mathbf{Z}$ and $(u, w) \in \mathsf{R}$ and computes $c_{key} \leftarrow \mathsf{Enc}((u, w); (\mathsf{hp}, \tilde{x}_1))$.
6. $\mathcal{S}$ sends $(\mathsf{proof}, 3, (\mathsf{hp}, c_{key}, Q_1))$ to $\mathcal{A}_2$, as $\mathcal{A}_2$ expects to receive from $\mathcal{F}_{\mathsf{zk}}^{R_{\mathsf{HPS-DL}}}$.
7. $\mathcal{S}$ sends $\mathsf{continue}$ to $\mathcal{F}_{\mathsf{ECDSA}}$ for P_1 to receive output, and stores Q.

Signing Phase

1. Upon input $\mathsf{Sign}(sid, m)$, simulator $\mathcal{S}$ sends $\mathsf{Sign}(sid, m)$ to $\mathcal{F}_{\mathsf{ECDSA}}$ and receives back a signature (r, s).
2. $\mathcal{S}$ computes the point $R = (r, r_y)$ using the ECDSA verification algorithm.
3. $\mathcal{S}$ invokes $\mathcal{A}_2$ with input $\mathsf{ISign}(sid, m)$ and sends $(\mathsf{proof-receipt}, sid\|1)$ as $\mathcal{A}_2$ expects to receive from $\mathcal{F}_{\mathsf{com-zk}}^{R_{\mathrm{DL}}}$.
4. $\mathcal{S}$ receives $(\mathsf{prove}, sid\|2, R_2, k_2)$ as $\mathcal{A}_2$ intends to send to $\mathcal{F}_{\mathsf{zk}}^{R_{\mathrm{DL}}}$.
5. Using the extracted value k_2, $\mathcal{S}$ verifies that R_2 is a non zero point and that $R_2 = k_2 P$. If so it computes $R_1 \leftarrow k_2^{-1} R$ and sends $(\mathsf{decom-proof}, sid\|1, R_1)$ to $\mathcal{A}_2$ as it expects to receive from $\mathcal{F}_{\mathsf{com-zk}}^{R_{\mathrm{DL}}}$. If not it simulates P_1 aborting and instructs the trusted party computing $\mathcal{F}_{\mathsf{ECDSA}}$ to abort.
6. $\mathcal{S}$ receives $c_3 = (\bar{u}, \bar{e})$ from $\mathcal{A}_2$, which it can decrypt using hk, *i.e.*,

$$\alpha \leftarrow \mathsf{Decode}\left(\frac{\bar{e}}{\mathcal{H}_{\mathsf{hk}}(\bar{u})}\right).$$

If $\alpha = k_2^{-1} \cdot (m' + r \cdot x_2 \cdot \tilde{x}_1) \mod q$ then $\mathscr{S}$ sends continue to the trusted party $\mathscr{F}_{\mathrm{ECDSA}}$, s.t. the honest party P_1 receives output. Otherwise it instructs $\mathscr{F}_{\mathrm{ECDSA}}$ to abort.

We now describe the sequence of games. Game_0 is the real execution of the protocol, and we finish in Game_6 which is the ideal simulation described above. In the following intermediary games, only the differences in the steps performed by $\mathscr{S}$ are depicted.

Game_0	Game_1	Game_2
$\mathrm{Q} \leftarrow x_1 x_2 \mathrm{P}$	$\mathrm{Q} \leftarrow x_1 x_2 \mathrm{P}$	$\mathrm{Q} \leftarrow x_1 x_2 \mathrm{P}$
$\vdots$	$\vdots$	$\vdots$
$\mathsf{hk} \hookleftarrow \mathscr{D}_{\mathsf{hk}}$	$\mathsf{hk} \hookleftarrow \mathscr{D}_{\mathsf{hk}}$	$\mathsf{hk} \hookleftarrow \mathscr{D}_{\mathsf{hk}}$
$\mathsf{hp} \leftarrow \alpha(\mathsf{hk})$	$\mathsf{hp} \leftarrow \alpha(\mathsf{hk})$	$\mathsf{hp} \leftarrow \alpha(\mathsf{hk})$
	Sample $(u, w) \in \mathrm{R}$	$\tilde{u} \xleftarrow{\$} \mathscr{X} \backslash \mathscr{L}$
$c_{key} \leftarrow \mathsf{Enc}(\mathsf{hp}, x_1)$	$c_{key} \leftarrow (u, \mathscr{H}_{\mathsf{hk}}(u) \cdot \mathsf{Encode}(x_1))$	$c_{key} \leftarrow (\tilde{u}, \mathscr{H}_{\mathsf{hk}}(\tilde{u}) \cdot \mathsf{Encode}(x_1))$
$\vdots$	$\vdots$	$\vdots$
$\mathrm{R} \leftarrow k_1 k_2 \mathrm{P}$	$\mathrm{R} \leftarrow k_1 k_2 \mathrm{P}$	$\mathrm{R} \leftarrow k_1 k_2 \mathrm{P}$

Game_3	Game_4
$\mathrm{Q} \leftarrow x_1 x_2 \mathrm{P}$	$\mathrm{Q} \leftarrow x_1 x_2 \mathrm{P}$
$\vdots$	$\vdots$
$\mathsf{hk} \hookleftarrow \mathscr{D}_{\mathsf{hk}}$	$\mathsf{hk} \hookleftarrow \mathscr{D}_{\mathsf{hk}}$
$\mathsf{hp} \leftarrow \alpha(\mathsf{hk})$	$\mathsf{hp} \leftarrow \alpha(\mathsf{hk})$
$\tilde{u} \xleftarrow{\$} \mathscr{X} \backslash \mathscr{L}, \gamma \xleftarrow{\$} \mathbf{Z}/q\mathbf{Z}$	Sample $(u, w) \in \mathsf{R}, \gamma \xleftarrow{\$} \mathbf{Z}/q\mathbf{Z}$
$c_{key} \leftarrow (\tilde{u}, \mathscr{H}_{\mathsf{hk}}(\tilde{u}) \cdot \mathsf{Encode}(x_1 + \gamma))$	$c_{key} \leftarrow (u, \mathscr{H}_{\mathsf{hk}}(u) \cdot \mathsf{Encode}(x_1 + \gamma))$
$\vdots$	$\vdots$
$\mathrm{R} \leftarrow k_1 k_2 \mathrm{P}$	$\mathrm{R} \leftarrow k_1 k_2 \mathrm{P}$

Game_5	Game_6
$\mathrm{Q} \leftarrow x_1 x_2 \mathrm{P}$	$\mathrm{Q} \leftarrow \mathscr{F}_{\mathrm{ECDSA}}$
$\vdots$	$\vdots$
$\mathsf{hk} \hookleftarrow \mathscr{D}_{\mathsf{hk}}$	$\mathsf{hk} \hookleftarrow \mathscr{D}_{\mathsf{hk}}$
$\mathsf{hp} \leftarrow \alpha(\mathsf{hk})$	$\mathsf{hp} \leftarrow \alpha(\mathsf{hk})$
Sample $\gamma \xleftarrow{\$} \mathbf{Z}/q\mathbf{Z}$	Sample $\gamma \xleftarrow{\$} \mathbf{Z}/q\mathbf{Z}$
$c_{key} \leftarrow \mathsf{Enc}(\mathsf{hp}, x_1 + \gamma))$	$c_{key} \leftarrow \mathsf{Enc}(\mathsf{hp}, x_1 + \gamma))$
$\vdots$	$\vdots$
$\mathrm{R} \leftarrow k_1 k_2 \mathrm{P}$	$\mathrm{R} \leftarrow \mathscr{F}_{\mathrm{ECDSA}}$

We now demonstrate that the previous games are indistinguishable from the view of $\mathscr{A}_2$. Intuitively, in Game_1 the simulator uses the secret hashing key hk instead of the public projection key hp to compute c_{key}. Though the values are computed differently, they are distributed identically, and are perfectly indistinguishable. Next in Game_2 we replace the first element of the ciphertext (in Game_1 this is $u \in \mathscr{L}$) with an element $\tilde{u} \in \mathscr{X} \backslash \mathscr{L}$. By the hardness of the subset membership problem Game_1 and Game_2 are indistinguishable. Next in Game_3 we multiply the second element of the ciphertext by a random element of the subgroup group $\mathscr{F}$ in which messages are encoded (or equivalently, add a random element of $\mathbf{Z}/q\mathbf{Z}$ to x_1 such that this sum will be encoded in $\mathscr{F}$), under the assumption that the hash proof system is δ_s−smooth over $\mathscr{X}$ in $\mathscr{F}$, the obtained distributions of the public key and ciphertext (as seen by an adversary) are δ_s−close. So both games are indistinguishable. We then again use the hardness of the subset membership problem underlying the hash proof to hop from Game_3 to Game_4, such that in the latter the first element of the ciphertext is once again in $\mathscr{L}$; and again Game_4 to Game_5 are identical from an adversary's point of view since we simply use the public evaluation function of the hash function $\mathscr{H}$ instead of the private one. And finally in Game_6 we change the way R and Q are generated.

We denote by E_i the probability that an algorithm interacting with the simulator in Game_i outputs 1. Thus by demonstrating that $|\Pr[\mathsf{E}_0] - \Pr[\mathsf{E}_6]|$ is negligible, we demonstrate that − from $\mathscr{A}_2$'s view, the real and ideal executions are indistinguishable.

Invalid ciphertexts

We define the notion of invalid ciphertexts as these will be useful in our game steps. A ciphertext is said to be ***invalid*** if it is of the form $(u', e') := (u', \mathscr{H}_{\mathsf{hk}}(u') \cdot \mathsf{Encode}(m'))$ where $u' \in \mathscr{X} \backslash \mathscr{L}$. Note that one can compute such a ciphertext using the secret hashing key hk, but not the public projection key hp; that the decryption algorithm applied to (u', e') with secret key hk recovers m'; and that an invalid ciphertext is indistinguishable of a valid one under the hardness of the subset membership problem.

Homomorphic properties over invalid ciphertexts

It is easy to verify that homomorphic operations hold even if a ciphertext is invalid, whether this be between two invalid ciphertexts of between a valid and invalid ciphertext. This is true since the homomorphic properties we required of the hash family hold over the whole group $\mathscr{X}$ (and not only in $\mathscr{L}$).

Game_0 **to** Game_1**.**

In Game_0 and in Game_1, Q and R are computed in the same way. The only difference between Game_0 and Game_1 is the way c_{key} is computed, namely we use the secret hashing key hk instead of the public projection key hp and the witness w to compute c_{key}. Though the values are computed differently, they are distributed identically, and are perfectly indistinguishable from an adversary's point of view:

$$\left|\Pr[\mathsf{E}_1] - \Pr[\mathsf{E}_0]\right| = 0.$$

Game_1 **to** Game_2**.**

Suppose that $\mathscr{D}$ is able to distinguish, with non negligible advantage, between the distribution generated in Game_1 from that generated in Game_2. Then we can devise $\hat{\mathscr{S}}$ that can use $\mathscr{D}$ to break the hard subset membership assumption, *i.e.*, distinguish random elements of $\mathscr{L}$ from those of $\mathscr{X}\backslash\mathscr{L}$. The input of $\hat{\mathscr{S}}$ is a hard subset membership challenge x^* which is either an element in $\mathscr{L}$ or an element of $\mathscr{X}\backslash\mathscr{L}$. Precisely $\hat{\mathscr{S}}$ works as $\mathscr{S}$ would in Game_1, interacting with the distinguisher $\mathscr{D}$ instead of $\mathscr{A}_2$, the only difference being that instead of sampling $(u, w) \in \mathsf{R}$ it sets $u := x^*$ and computes $c_{key} \leftarrow (u, \mathscr{H}_{\mathsf{hk}}(u) \cdot \mathsf{Encode}(x_1))$. When $\mathscr{D}$ returns a bit b (relative to Game_{b+1}), $\hat{\mathscr{S}}$ returns the same bit, where 0 represents the case $x^* \in \mathscr{L}$ and 1 represents the case $x^* \in \mathscr{X}\backslash\mathscr{L}$.

Analysis – Case $x^* \in \mathscr{L}$

There exists $w \in \mathscr{W}$ such that $(x^*, w) \in \mathsf{R}$ and $\mathscr{H}_{\mathsf{hp}}(x^*, w) = \mathscr{H}_{\mathsf{hk}}(x^*)$. So $c_{key} = (u, e)$ is an encryption of x_1 as computed in Game_1. *Case* $x^* \in \mathscr{X}\backslash\mathscr{L}$*:* The ciphertext is $(x^*, \mathscr{H}_{\mathsf{hk}}(x^*) \cdot \mathsf{Encode}(x_1))$, which is exactly the distribution obtained in Game_2. So the advantage of $\hat{\mathscr{S}}$ in breaking the hard subset membership assumption is at least that of $\mathscr{D}$ in distinguishing both games. Thus:

$$|\Pr[\mathsf{E}_2] - \Pr[\mathsf{E}_1]| \le \delta_{\mathscr{L}}.$$

Game_2 **to** Game_3**.**

Let us denote $\tilde{x}_1 := x_1 + \gamma \mod q$. Under the assumption that the hash proof system is δ_s-smooth over $\mathscr{X}$ in $\mathscr{F}$ (*i.e.*, the co-domain of Encode), it holds that the distribution of $(x^*, \mathscr{H}_{\mathsf{hk}}(x^*) \cdot \mathsf{Encode}(x_1))$ and of $(x^*, \mathscr{H}_{\mathsf{hk}}(x^*) \cdot \mathsf{Encode}(x_1) \cdot \mathsf{Encode}(\gamma) = \mathscr{H}_{\mathsf{hk}}(x^*) \cdot \mathsf{Encode}(\tilde{x}_1))$ for some random $\tilde{x}_1 \in \mathbf{Z}/q\mathbf{Z}$ are δ_s–close. Thus replacing $(x^*, \mathscr{H}_{\mathsf{hk}}(x^*) \cdot \mathsf{Encode}(x_1))$ by $(x^*, \mathscr{H}_{\mathsf{hk}}(x^*) \cdot \mathsf{Encode}(\tilde{x}_1))$ – as is done from Game_2 to Game_3 – cannot be noticed by any PT adversary with advantage greater than δ_s and:

$$\left|\Pr[\mathsf{E}_3] - \Pr[\mathsf{E}_2]\right| \le \delta_s.$$

Game_3 **to** Game_4**.**

The change here is exactly that between Game_1 and Game_2, thus both games are indistinguishable under the hardness of the subset membership problem on which relies the hash proof system and:

$$\left|\Pr[\mathsf{E}_4] - \Pr[\mathsf{E}_3]\right| \le \delta_{\mathscr{L}}.$$

Game_4 **to** Game_5**.**

The change here is exactly that between Game_0 and Game_1, thus both games are perfectly indistinguishable, even for an unbounded adversary, thus:

$$\left|\Pr[\mathsf{E}_5] - \Pr[\mathsf{E}_4]\right| = 0.$$

Game_5 **to** Game_6**.**

The only differences between Game_5 and Game_6 are the ways Q and R are generated. In Game_5, Q and R derive from a Diffie-Hellman Key Exchange, which can be simulated. Moreover, since the ideal functionality $\mathcal{F}_{\mathrm{ECDSA}}$ samples Q and R uniformly at random from the group $\mathbb{G}$, it holds that $x_2^{-1}\mathrm{Q}$ and $x_1\mathrm{P}$ have the same distribution, as do $k_2^{-1}\mathrm{R}$ and $k_1\mathrm{P}$. All other steps of Game_5 and Game_6 are identical. We conclude that, in the $\mathcal{F}_{\mathsf{zk}}$, $\mathcal{F}_{\mathsf{com-zk}}$ hybrid model, Game_5 and Game_6 are identical from $\mathcal{A}_2$'s view, and so:

$$\left|\Pr[\mathrm{E}_6] - \Pr[\mathrm{E}_5]\right| = 0.$$

Real/Ideal executions.

Putting together the above probabilities, we get that:

$$\left|\Pr[\mathrm{E}_6] - \Pr[\mathrm{E}_0]\right| \leq 2\delta_{\mathscr{L}} + \delta_s,$$

and so, assuming the hardness of the subset membership problem underlying HPS, and assuming the smoothness of HPS, it holds that the real and ideal executions are computationally indistinguishable from $\mathcal{A}_2$'s view, which concludes the proof of the theorem. □

F.4. Instantiation in Class Groups of an Imaginary Quadratic Field

In this section, we give an instantiation of a hash proof system with the required properties in order to apply the generic construction of the previous section. For that we will use a linearly homomorphic encryption scheme modulo a prime number, denoted CL in the following, introduced in [CL15] using a group with an easy Dlog subgroup, with a concrete instantiation using class groups of quadratic fields. In order to define a HPS, we use the recent results of [CLT18] that enhance the CL framework by introducing a hard subgroup membership assumption (HSM). We first give the definition of this assumption in the context of a group with an easy Dlog subgroup, then the instantiation with class groups, and then define a HPS from HSM and prove that it has the required properties to instantiate the generic construction in Section F.3.

F.4.1. A Hard Subgroup Membership Assumption

To start with, we explicitly define the generator $\mathsf{GenGroup}$ used in the framework of a group with an easy Dlog subgroup introduced in [CL15] and enhanced in [CLT18], with small modifications as discussed below.

Definition F − 4. Let $\mathsf{GenGroup}$ be a pair of algorithms $(\mathsf{Gen}, \mathsf{Solve})$. The Gen algorithm is a group generator which takes as inputs a parameter λ and a prime q and outputs a tuple $(\tilde{s}, g, f, g_q, \widehat{G}, G, F, G^q)$. The set $(\widehat{G}, \cdot)$ is a finite abelian group of order $q \cdot \hat{s}$ where the bitsize of $\hat{s}$ is a function of λ and $\gcd(q,\hat{s}) = 1$. The algorithm Gen only outputs an upper bound $\tilde{s}$ of $\hat{s}$. It is also required that one can efficiently recognise valid encodings of elements in $\widehat{\mathrm{G}}$. The set $(\mathrm{F},\cdot)$ is the unique cyclic subgroup of $\widehat{\mathrm{G}}$ of order q, generated by f. The set $(\mathrm{G},\cdot)$ is a cyclic subgroup of $\widehat{\mathrm{G}}$ of order $q \cdot s$ where s divides $\hat{s}$. By construction $\mathrm{F} \subset \mathrm{G}$, and, denoting $\mathrm{G}^q := \{x^q, x \in \mathrm{G}\}$ the subgroup of order s of G, it holds that $\mathrm{G} = \mathrm{G}^q \times \mathrm{F}$. The algorithm Gen outputs f, g_q and $g := f \cdot g_q$ which are respective generators of F, G^q and G. Moreover, the Dlog problem is easy in F, which means that the Solve algorithm is a deterministic polynomial time algorithm that solves the discrete logarithm problem in F:

$$\Pr\Big[x = x^\star : (\tilde{s}, g, f, g_q, \widehat{\mathrm{G}}, \mathrm{G}, \mathrm{F}, \mathrm{G}^q) \leftarrow \mathsf{Gen}(1^\lambda, q), x \xleftarrow{\$} \mathbf{Z}/q\mathbf{Z}, \mathrm{X} \leftarrow f^x,$$
$$x^\star \leftarrow \mathsf{Solve}(q, \tilde{s}, g, f, g_q, \widehat{\mathrm{G}}, \mathrm{G}, \mathrm{F}, \mathrm{G}^q, \mathrm{X})\Big] = 1.$$

Remark F – 1. In this definition, there are a few modifications compared to the definition of [CLT18]. Namely we take as input the prime q instead of having Gen generating it, and we output the group $\widehat{\mathrm{G}}$ from which the group G with an easy Dlog subgroup F is produced. In practice, with the concrete instantiation with class groups, this is a just a matter of rewriting: the prime q was generated independently of the rest of the output in [CL15, CLT18] so it can be an input of the algorithm, and the group $\widehat{\mathrm{G}}$ would be the class group which was implicitly defined by its discriminant. We note that it is easy to recognise valid encodings of $\widehat{\mathrm{G}}$ while it will be not so for elements of $\mathrm{G} \subset \widehat{\mathrm{G}}$. This is an important difference with Paillier's encryption, and one of the reason why Lindell's $\mathrm{L}_{\mathrm{PDL}}$ proof does not work in our setting.

We recall here the definition of a hard subgroup membership (HSM) problem within a group with an easy Dlog subgroup as defined in [CLT18]. HSM is closely related to Paillier's DCR assumption. Such hard subgroup membership problems are based on a long line of assumptions on the hardness of distinguishing powers in groups. In short, DCR and HSM are essentially the same assumption but in different groups, hence there is no direct reduction between them. We emphasise that this assumption is well understood both in general, and for the specific case of class groups of quadratic fields, which we will use to instantiate $\mathsf{GenGroup}$. It was first used by [CLT18] within class groups, this being said, cryptography based on class groups is now well established, and is seeing renewed interest as it allows versatile and efficient solutions such as encryption switching protocols [CIL17], inner product functional encryption [CLT18] or verifiable delay functions [BBBF18, Wes19].

In Def. F – 4, one has $\mathrm{G} = \mathrm{F} \times \mathrm{G}^q$. The assumption is that it is hard to distinguish the elements of G^q in G.

Definition F – 5 (HSM assumption)**.** We say that $\mathsf{GenGroup}$ is the generator of a HSM group with easy Dlog subgroup F if it holds that the HSM problem is hard even with access to the Solve algorithm. Let $\mathscr{D}$ (resp. $\mathscr{D}_q$) be a distribution over the integers such that the distribution $\{g^x, x \hookleftarrow \mathscr{D}\}$ (resp. $\{g_q^x, x \hookleftarrow \mathscr{D}_q\}$) is at distance less than $2^{-\lambda}$

from the uniform distribution in G (resp. in G^q). Let $\mathscr{A}$ be an adversary for the HSM problem, its advantage is defined as:

$$\begin{aligned}\mathsf{Adv}^{\mathsf{HSM}}_{\mathscr{A}}(\lambda) = \Big|2\cdot\Pr\big[b = b^\star : (\tilde{s}, g, f, g_q, \widehat{G}, G, F, G^q) \leftarrow \mathsf{Gen}(1^\lambda, q),\\ x \hookleftarrow \mathscr{D}, x' \hookleftarrow \mathscr{D}_q, b \overset{\$}{\leftarrow} \{0,1\}, Z_0 \leftarrow g^x, Z_1 \leftarrow g_q^{x'},\\ b^\star \leftarrow \mathscr{A}(q, \tilde{s}, g, f, g_q, \widehat{G}, G, F, G^q, Z_b, \mathsf{Solve}(.))\big] - 1\Big|\end{aligned}$$

The HSM problem is said to be hard in G if for all probabilistic polynomial time attacker $\mathscr{A}$, $\mathsf{Adv}^{\mathsf{HSM}}_{\mathscr{A}}(\lambda)$ is negligible.

Class groups

Our instantiation makes use of class groups of orders of imaginary quadratic fields. We refer the interested reader to [BH01] for background on this algebraic object and its early use in cryptography. We here briefly sketch an instantiation of algorithm $\mathsf{GenGroup}$ in Definition F−4, following [CL15, Fig. C.2]. The formal description is given in Fig. F.6 below and concrete details can be found in [CL15]. Let q be a prime. We construct a fundamental discriminant $\Delta_K := -q\cdot\tilde{q}$ where $\tilde{q}$ is a prime such that $q\cdot\tilde{q} \equiv -1 \pmod 4$ and $(q/\tilde{q}) = -1$. We then consider the non-maximal order of discriminant $\Delta_q := q^2\cdot\Delta_K$ and its class group $\widehat{G} := Cl(\Delta_q)$ whose order is $h(\Delta_q) = q\cdot h(\Delta_K)$ where $h(\Delta_K)$ is the class number, *i.e.*, the order of $Cl(\Delta_K)$, the class group of fundamental discriminant Δ_k. This number is known to satisfy the following inequality (see [Coh00, p. 295] for instance): $h(\Delta_K) < \frac{1}{\pi}\log|\Delta_K|\sqrt{|\Delta_K|}$ which is the bound we take for $\tilde{s}$ (a slightly better bound can be computed from the analytic class number formula).

Elements of $\widehat{G}$ are classes of ideals of the order of discriminant Δ_q. Such classes can be represented by a unique reduced ideal. Moreover, ideals can be represented using the so-called two elements representation which correspond to their basis as a lattice of dimension two. Informally, classes can be uniquely represented by two integers (a, b), $a, b < \sqrt{|\Delta_q|}$ and one can efficiently verify that this indeed defines an element of $\widehat{G}$ (by checking if $b^2 \equiv \Delta_q \pmod{4a}$). The arithmetic in class groups (which corresponds to reduction and composition of quadratic forms) is very efficient: the algorithms have a quasi linear time complexity using fast arithmetic (see [Coh00]).

Following Fig. C.2 of [CL15], we build a generator g_q of a cyclic subgroup of q−th powers of $\widehat{G}$, and denote $G^q := \langle g_q\rangle$. Then we build a generator f for the subgroup F of order q, and then set $g := f\cdot g_q$ as a generator of a cyclic subgroup G of $Cl(\Delta_q)$ of order $q\cdot s$, where s is unknown. Computing discrete logarithms is easy in F thanks to the following facts. We denote the surjection $\bar{\varphi}_q : Cl(\Delta_q) \longrightarrow Cl(\Delta_K)$. From Proposition C−1 of [CL15], its kernel is cyclic of order q and is generated by f represented by (q^2, q). Moreover, if $1 \leqslant m \leqslant q-1$ then, once reduced, f^m is of the form $(q^2, L(m)q)$ where $L(m)$

$\mathsf{Gen}(1^\lambda, q)$

1. Let μ be the bit size of q. Pick $\tilde{q}$ a random $\eta(\lambda) - \mu$ bits prime such that $q\tilde{q} \equiv -1 \pmod 4$ and $(q/\tilde{q}) = -1$.
2. $\Delta_K \leftarrow -q\tilde{q}$, $\Delta_q \leftarrow q^2\Delta_K$ and $\widehat{G} \leftarrow Cl(\Delta_q)$
3. $f \leftarrow [(q^2, q)]$ in $Cl(\Delta_q)$ and $F := \langle f \rangle$
4. $\tilde{s} \leftarrow \lceil \frac{1}{\pi} \log|\Delta_K|\sqrt{|\Delta_K|}\rceil$
5. Let r be a small prime, with $r \neq q$ and $\left(\frac{\Delta_K}{r}\right) = 1$, set $\mathfrak{r}$ an ideal lying above r.
6. Set $g_q \leftarrow [\varphi_q^{-1}(\mathfrak{r}^2)]^q$ in $C(\Delta_q)$ and $G^q \leftarrow \langle g_q \rangle$
7. Set $g \leftarrow g_p \cdot f$ and $G \leftarrow \langle g \rangle$
8. Return $(\tilde{s}, g, f, g_q, \widehat{G}, G, F, G^q)$

Figure F.6: Group generator Gen

is the odd integer in $[-q, q]$ such that $L(m) \equiv 1/m \pmod q$, which gives the efficient algorithm to compute discrete logarithms in $\langle f \rangle$.

Note that following [CL15] the bit size of q must have at least λ bits, where λ is the security parameter, which is the case for ECDSA: q will be the order of the elliptic curve. The size $\eta(\lambda)$ of Δ_K is chosen to resist the best practical attacks, which consists in computing discrete logarithms in $Cl(\Delta_K)$ (or equivalently the class number $h(\Delta_K)$). An index-calculus method to solve the Dlog problem in a class group of imaginary quadratic field of discriminant Δ_K was proposed in [Jac00]. It is conjectured in [BJS10] that a state of the art implementation of this algorithm has complexity $\mathscr{O}(L_{|\Delta_K|}[1/2, o(1)])$, which allows to use asymptotically shorter keys compared to protocols using classical problems that are solved in subexponential complexity $\mathscr{O}(L[1/3, o(1)])$ (see Section F.5 for concrete sizes for η).

F.4.2. A Smooth Homomorphic Hash Proof System from HSM

We set $\mathscr{X} := G$ and $\mathscr{L} := G^q$ then $\mathscr{X} \cap \mathscr{L} = G^q$ and the HSM assumption states that it is hard to distinguish random elements of G from those of G^q. This clearly implies the hardness of the subset membership problem, *i.e.*, it is hard to distinguish random elements of $G \backslash G^q$ from those of G^q.

Let $\mathscr{D}$ be a distribution over the integers such that the distribution $\{g^w, w \hookleftarrow \mathscr{D}\}$ is at distance $\delta_{\mathscr{D}} \leq 2^{-\lambda}$ of the uniform distribution in G.

Associated projective hash family.

Let PHF be the projective hash family associated to the above subset membership problem, the description of which specifies:

- A hash key space $\mathrm{K} := \mathbf{Z}$.
- A keyed hash function, with input and output domain G, s.t., for $\mathsf{hk} \hookleftarrow \mathscr{D}$, and for $x \in \mathrm{G}$, $\mathscr{H}_{\mathsf{hk}}(x) := x^{\mathsf{hk}}$.
- An auxiliary function $\alpha : \mathrm{K} \mapsto \mathrm{G}^q$ such that for $\mathsf{hk} \in \mathrm{K}$, $\alpha(\mathsf{hk}) := \mathscr{H}_{\mathsf{hk}}(g_q) = g_q^{\mathsf{hk}}$. Notice that for a hash key hk, and for $x \in \mathrm{G}^q$, the knowledge of $\alpha(\mathsf{hk})$ completely determines the value $\mathscr{H}_{\mathsf{hk}}(x)$.
- An efficient public evaluation function, such that, for $x \in \mathrm{G}^q$ with witness w such that $x = g_q^w$ one can efficiently compute $\mathscr{H}_{\mathsf{hk}}(x) = \alpha(\mathsf{hk})^w = x^{\mathsf{hk}}$ knowing only the value of the auxiliary function $\alpha(\mathsf{hk})$ (but not hk).

Lemma F − 1 (Smoothness). *The projective hash family* PHF *is* δ_s*-smooth over* G *in* F*, with* $\delta_s \leqslant 2\delta_{\mathscr{D}}$, i.e., *for any* $x \in \mathrm{G}\backslash\mathrm{G}^q$, $\pi \leftarrow f^{\gamma} \in \mathrm{F} \subset \mathrm{G}$ *where* $\gamma \xleftarrow{\$} \mathbf{Z}/q\mathbf{Z}$ *and* $k \hookleftarrow \mathscr{D}$*, the distributions* $\mathscr{D}_1 := \{x, g_q^k, \pi \cdot x^k\}$ *and* $\mathscr{D}_2 := \{x, g_q^k, x^k\}$ *are less than* $2\delta_{\mathscr{D}}$*-close.*

Proof. For $x \in \mathrm{G}\backslash\mathrm{G}^q$, there exist $a \in \mathbf{Z}/s\mathbf{Z}$ and $b \in (\mathbf{Z}/q\mathbf{Z})^*$ such that $x = g_q^a f^b$. Thus we can write $\mathscr{D}_1 = \{g_q^a f^b, g_q^k, g_q^{a\cdot k} f^{b\cdot k+\gamma}\}$ and $\mathscr{D}_2 = \{g_q^a f^b, g_q^k, g_q^{a\cdot k} f^{b\cdot k}\}$. It remains to study the statistical distance of the third coordinates of the two distributions, given the two first coordinates, *i.e*, if $(a \mod s)$, $(b \mod q)$, and $(k \mod s)$ are fixed. This is the statistical between $\mathrm{X} := b \cdot k + \gamma$ and $\mathrm{Y} := b \cdot k$ in $\mathbf{Z}/q\mathbf{Z}$. Since γ is uniform in $\mathbf{Z}/q\mathbf{Z}$, X is the uniform distribution. As $\mathscr{D}$ is by definition at statistical distance $\delta_{\mathscr{D}}$ from the uniform distribution modulo $q \cdot s$, and $\gcd(q, s) = 1$, one can prove (cf. Lemma F − 2 in Appendix F.B) that even knowing $(k \mod s)$, the distribution of $(k \mod q)$ is at distance less than $2\delta_{\mathscr{D}}$ from the uniform distribution over $\mathbf{Z}/q\mathbf{Z}$. As a result, the distance between X and Y is bounded by $2\delta_{\mathscr{D}}$, which concludes the proof. □

Linearly homomorphic.

For all $\mathsf{hk} \in \mathbf{Z}$, and $u_1, u_2 \in \mathrm{G}$, $\mathscr{H}_{\mathsf{hk}}(u_1) \cdot \mathscr{H}_{\mathsf{hk}}(u_2) = u_1^{\mathsf{hk}} \cdot u_2^{\mathsf{hk}} = (u_1 \cdot u_2)^{\mathsf{hk}} = \mathscr{H}_{\mathsf{hk}}(u_1 \cdot u_2)$. Thus $\mathscr{H}_{\mathsf{hk}}$ is a homomorphism for each hk.

Resulting encryption scheme.

A direct application of Subsection F.3.3 using the above HPS results in the encryption scheme called HSM-CL in [CLT18], which is linearly homomorphic modulo q and ind − cpa under the HSM assumption. We describe this scheme in Fig. F.7 for completeness. Note that here the secret key x (and the randomness r) is drawn with a distribution $\mathscr{D}_q$ such that $\{g_q^x, x \hookleftarrow \mathscr{D}_q\}$ is at distance less than $2^{-\lambda}$ from the uniform distribution in G^q, this does not change the view of the attacker. Let $\mathrm{S} := 2^{\lambda-2} \cdot \tilde{s}$. In practice, we will use for $\mathscr{D}_q$ the uniform distribution on $\{0, \ldots, \mathrm{S}\}$.

```
Algorithm KeyGen$(1^\lambda, q)$
  1. $(\tilde{s}, g, f, g_q, \widehat{\mathrm{G}}, \mathrm{G}, \mathrm{F}, \mathrm{G}^q) \leftarrow \mathsf{Gen}(1^\lambda, q)$
  2. Pick $x \hookleftarrow \mathscr{D}_q$ and $h \leftarrow g_q^x$
  3. Set $pk \leftarrow (\tilde{s}, g_q, f, p, h)$
  4. Set $sk \leftarrow x$
  5. Return $(pk, sk)$
```

```
Algorithm Enc$(pk, m)$
  1. Pick $r \hookleftarrow \mathscr{D}_q$
  2. Return $(g_q^r, f^m h^r)$
```

```
Algorithm Dec$(sk, (c_1, c_2))$
  1. Compute $\mathrm{M} \leftarrow c_2/c_1^x$
  2. Return $\mathsf{Solve}(\mathrm{M})$
```

Figure F.7: Description of the HSM-CL encryption scheme

F.4.3. A zero-knowledge proof for $\mathrm{R}_{\mathsf{CL-DL}}$

We describe here the ZKPoK for $\mathrm{R}_{\mathsf{HPS-DL}}$ used for our instantiation with the encryption scheme of Fig. F.7 and denote it $\mathrm{R}_{\mathsf{CL-DL}}$. It relies on the Schnorr-like GPS (statistically) zero-knowledge identification scheme [GPS06] that we turn into a zero-knowledge proof of knowledge of the randomness used for encryption and of the discrete logarithm of an element on an elliptic curve, using a binary challenge. This proof is partly performed in a group of unknown order.

We denote $c_{key} := (c_1, c_2)$. If c_{key} is a valid encryption of x_1 under public key pk it holds that $c_{key} = (g_q^r, f^{x_1}\mathsf{pk}^r)$ for some $r \in \{0, \dots, \mathrm{S}\}$. The protocol $\mathrm{R}_{\mathsf{CL-DL}}$ provides a ZKPoK for the following relation

$$\mathrm{R}_{\mathsf{CL-DL}} := \{(\mathsf{pk}, (c_1, c_2), \mathrm{Q}_1); (x_1, r) \mid c_1 = g_q^r \wedge c_2 = f^{x_1}\mathsf{pk}^r \wedge \mathrm{Q}_1 = x_1\mathrm{G}\}.$$

The following theorem, whose proof is given in Appendix F.C, states the security of the zero-knowledge proof of knowledge $\mathrm{R}_{\mathsf{CL-DL}}$.

Theorem F – 2. *The protocol described in Figure F.8 is a statistical zero-knowledge proof of knowledge with soundness $2^{-\ell}$, as long as ℓ is polynomial and $\ell\mathrm{S}/\mathrm{A}$ is negligible, where A is a positive integer.*

F.4.4. Two-Party Distributed ECDSA Protocol from HSM

The protocol results from a direct application of Subsection F.3.5 using the HPS defined in Subsection F.4.2, an the $\mathrm{R}_{\mathsf{CL-DL}}$ proof of the previous subsection. Therefore we defer the detailed protocol to Appendix F.E, Fig. F.10 and simply state the following theorem.

Theorem F – 3. *Assuming* GenGroup *is the generator of a* HSM *group with easy Dlog subgroup* F*, then the protocol of Appendix F.E securely computes $\mathscr{F}_{\mathsf{ECDSA}}$ in the $(\mathscr{F}_{\mathsf{zk}}, \mathscr{F}_{\mathsf{com-zk}})$-hybrid model in the presence of a malicious static adversary (under the ideal/real definition).*

Input : (r, x_1) and $(\mathsf{pk}, c_1, c_2, \mathrm{Q}, \mathrm{P})$		Input : $(\mathsf{pk}, c_1, c_2, \mathrm{Q}, \mathrm{P})$
`Repeat` ℓ `times`		
$r_1 \xleftarrow{\$} [0, \mathrm{A}[\,;\, r_2 \xleftarrow{\$} \mathbf{Z}/q\mathbf{Z}$		
$t_1 \leftarrow \mathsf{pk}^{r_1} f^{r_2}\,;\, t_2 \leftarrow r_2\mathrm{P}\,;\, t_3 \leftarrow g_q^{r_1}$	$\xrightarrow{\quad t_1, t_2, t_3 \quad}$	
	$\xleftarrow{\quad k \quad}$	$k \xleftarrow{\$} \{0, 1\}$
$u_1 \leftarrow r_1 + kr$ in $\mathbf{Z}$		
$u_2 \leftarrow r_2 + kx_1 \mod q$	$\xrightarrow{\quad u_1, u_2 \quad}$	`Check` $u_1 \in [0, \mathrm{A} + \mathrm{S}[$ $t_1 \cdot c_2^k = \mathsf{pk}^{u_1} \cdot f^{u_2}$ $t_2 + [k]\mathrm{Q} = [u_2]\mathrm{P}$ $t_3 \cdot c_1^k = g_q^{u_1}$

Figure F.8: The zero-knowledge proof of knowledge $\mathrm{R}_{\mathsf{CL-DL}}$

F.5. Implementation and Efficiency Comparisons

In this section we compare an implementation of our protocol with Lindell's protocol of [Lin17]. For fair comparison, we implement both protocols with the Pari C Library ([PAR18]), as this library handles arithmetic in class groups, $\mathbf{Z}/n\mathbf{Z}$ and elliptic curves. In particular, in this library, exponentiations in $\mathbf{Z}/n\mathbf{Z}$ and in class groups both use the same sliding window method. The running times are measured on a single core of an Intel(R) Core(TM) i7-7700 CPU @ 3.60GHz (even if key generation can easily be parallelized). We do not implement commitments (this does not bias the comparison as they appear with equal weight in both schemes), and we only measure computation time and do not include communication (again this is fair as communication is similar).

As in [Lin17], we ran our implementation on the standard NIST curves P-256, P-384 and P-521, corresponding to levels of security 128, 192 and 256. For the encryption scheme, we start with a 112 bit security, as in [Lin17], but also study the case where its level of security matches the security of the elliptic curves.

Again as in [Lin17], we fixed the number of rounds in zero knowledge proofs to reach a statistical soundness error of 2^{-40}. For the distributions we also set the parameters to get statistical error of 2^{-40}. The zero knowledge proofs for R_{DL} are implemented with the Schnorr protocol.

In the following, we review the theoretical complexity and experimental results of both schemes, before comparing them. In terms of theoretical complexity, exponentiations in the encryption schemes dominate the computation as elliptic curve operations are much cheaper. Thus, we only count these exponentiations; we will see this results in an accurate prediction of experimental timings.

Curve	Sec. Param.	Modulus	Expo. (ms)	Keygen (ms)	Signing (ms)	Keygen (b)	Signing (b)
P-256	112	2048	**7**	**2 133**	**20**	881 901	5 636
P-256	128	3072	**22**	**6 340**	**49**	1 317 101	7 684
P-384	192	7680	214	65 986	**437**	3 280 429	17 668
P-521	256	15360	1196	429 965	2 415	6 549 402	33 832

(a) Lindell's Protocol with Paillier

Curve	Sec. Param.	Discriminant	Expo. (ms)	Keygen (ms)	Signing (ms)	Keygen (b)	Signing (b)
P-256	112	1348	32	5 521	101	**178 668**	**4 748**
P-256	128	1827	55	9 350	170	**227 526**	**5 706**
P-384	192	3598	**212**	**35 491**	649	**427 112**	**10 272**
P-521	256	5971	**623**	**103 095**	**1 888**	**688 498**	**16 078**

(b) Our Protocol with HSM-CL

Figure F.9: Experimental results (timings in ms, sizes in bits)

F.5.1. Lindell's Protocol with Paillier's Encryption Scheme

The key generation uses on average 360 Paillier exponentiations (of the form $r^{\mathrm{N}} \bmod \mathrm{N}^2$) but not all of them are full exponentiations. The signing phase uses only 2 Paillier exponentiations.

The timings corresponds to the mean of several experiments (30 to 1000 depending on the security level). The timings are quite stable other than the generation of the RSA modulus in the key generation. We use standard RSA integers (*i.e.*, not strong prime factors) as this would be too slow for high security levels. For example, for 256 bits security (15360 bits modulus), the generation of the modulus takes 95 seconds (mean of 30 experiments) with a standard deviation of 56s. For the rest of the protocol the experimental timings are roughly equal to the number of exponentiations multiplied by the cost of one exponentiation.

The result are summarized in Fig. F.9(a). Timings are given in milliseconds and sizes in bits. The columns corresponds to the elliptic curve used for ECDSA, the security parameter in bits for the encryption scheme, the corresponding modulus bit size, the timings of one Paillier exponentiation, of the key generation and of the signing phase and the total communication in bits for two phases. Modulus sizes are set according to the NIST recommendations.

Note that for the first line, we use a 2048 bits modulus as in [Lin17] and we obtain a similar experimental result.

F.5.2. Our Protocol with HSM-CL Encryption Scheme

The key generation uses a total of 160 class group exponentiations (of the form g_q^r in the class group of discriminant $\Delta_q = -q^3 \cdot \tilde{q}$). This corresponds to the 40 rounds of the $\mathrm{R}_{\mathsf{CL-DL}}$ zero-knowledge proof of knowledge of Fig. F.8. Note that exponentiations in $\langle f \rangle$ are almost free as seen in Subsection F.4.1. Signing uses 3 class group exponentiations (one encryption and one decryption).

We use the same number of experiments as for Lindell's protocol. Here timings are very stable. Indeed during key generation, we only compute the public key $h \leftarrow g_q^x$ with one exponentiation, as the output of Gen (mainly the discriminant Δ_q of the class group and the generator g_q) is a common public parameter that only depends on the cardinality q of the elliptic curve. As a result this can be considered as an input of the protocol, as the same group can be used by all users. Moreover, doing this does not change the global result of the comparison with Lindell's protocol: the running time of Gen is dominated by the generation of $\tilde{q}$, a prime of size much smaller than the factor of the RSA modulus. So even if we add this running time in the Keygen column, this does not affect the results of our comparisons for any of the considered security levels.

The results are summarized in Fig. F.9(b). Timings are given in milliseconds and sizes in bits. The columns correspond to the elliptic curve used for ECDSA, the security parameter in bits for the encryption scheme, the corresponding fundamental discriminant $\Delta_K = -q \cdot \tilde{q}$ bit size, the timings of one class group exponentiation, of the key generation and of the signing phase and the total communication in bits for two phases. The discriminant sizes are chosen according to [BJS10].

F.5.3. Comparison

Figure F.9 shows that Lindell's protocol is faster for both key generation and signing for standard security levels for the encryption scheme (112 and 128 bits of security) while our solution remains of the same order of magnitude. However for high security levels, our solution becomes faster (in terms of key generation from a 192-bits security level and for both key generation and signing from a 256-bits security level).

In terms of communications, our solution outperforms the scheme of Lindell at all level of security by a factor 5 to 10 for Keygen. For Signing, we gain 15% for basic security to a factor 2 at 256 bits security level. In terms of rounds, our protocol uses 126 rounds for Keygen and Lindell's protocol uses 175 rounds, so we get a 28% gain. Both protocol use 7 rounds for Signing.

This situation can be explained by the following facts. Firstly we use less than half the number of exponentiations in the key generation as we do not need a range proof: our message space is $\mathbf{Z}/q\mathbf{Z}$ as the CL encryption scheme is homomorphic modulo a prime. Secondly, with class groups of quadratic fields we can use lower parameters than with $\mathbf{Z}/n\mathbf{Z}$ (as shown in the introduction, the best algorithm against the discrete logarithm problem in class groups has complexity $\mathcal{O}(\mathrm{L}[1/2, o(1)])$ compared to an $\mathcal{O}(\mathrm{L}[1/3, o(1)])$ for factoring). However, the group law is more complex in class groups. By comparing the Expo. time columns in the tables, we see that exponentiations in class groups are cheaper from the 192 bits level. So even if we use half as many exponentiations, the key generation for our solution only takes less time from that level (while being of the same order of magnitude below this level). For signing, we increase the cost by one exponentiation due to the Elgamal structure of the CL encryption scheme. However, one can note that we can pre process this encryption by computing (g_q^τ, h^τ) in an offline phase and computing $c_1 \leftarrow (g_q^\tau, h^\tau f^{k_2^{-1} m'})$ which results in only one multiplication in the online phase (cf. Appendix F.E). As a result we will have only one exponentiation in the online signing for the decryption operation. The same holds for

Lindell's protocol with Paillier. Using that both protocols take the same time for signing at the 192 bits level.

Increasing the number of rounds to obtain a 2^{-60} soundness error.

This impacts only KeyGen, where the [Lin17] scheme and ours both use 40 iterations of ZK proofs to achieve a 2^{-40} soundness error. Lindell's protocol performs 9 exponentiations per iteration while ours performs 4. All timings will thus be multiplied by 3/2 to achieve a 2^{-60} soundness error, and indeed this is what we observe in practice. Complexity is linear in the number of iterations and the ratio between our timings and those of [Lin17] remains constant.

F.6. Instantiation of our Generic Construction Using DCR

As stated at the end of the introduction, we can instantiate the generic construction of Section F.3 with the hash proof system built upon Paillier's decision composite residuosity assumption (DCR). This yields the Elgamal Paillier encryption scheme of [CS03] that closely resembles the HSM-CL Encryption scheme. However, the message space is $\mathbf{Z}/n\mathbf{Z}$ as in Lindell's protocol, so in addition to the $\mathsf{R}_{\mathsf{HPS-DL}}$ proof, P_1 has to prove that x_1 is an element of $\mathbf{Z}_q$ with a range proof. Moreover, this encryption scheme uses two exponentiations instead of one for Paillier. This being said a gain arrises from the fact that following the techniques of [CS03] one can make a sound proof for $\mathsf{R}_{\mathsf{HPS-DL}}$ in a single round by relying on the strong RSA assumption. This means that one should use safe primes that can be very costly to generate at high security level. However, for 112 and 128 bits of security this should give a competitive solution compared to Lindell's with a simulation based security relying on the hardness of classical problems, the DCR and the strong RSA assumptions.

F.7. Conclusion

Inspired by Lindell's scheme, we have provided the first generic construction for two-party ECDSA signing from hash proof systems which are homomorphic modulo a prime number. Theoretically, our construction allows for a simulation-based proof of security that is both tight and requires no artificial interactive assumptions, due to the structure of the underlying semantically secure homomorphic encryption schemes. Practically, we provide a detailed instantiation, and C implementation, from class groups of imaginary quadratic fields using the CL framework. This yields a better performance than Lindell's Paillier-based scheme for high levels of security, and same order of magnitude for standard levels. Our solution becomes faster than Lindell's from 192-bits of security upwards. Improvements could come from advances in ideal arithmetic in imaginary quadratic fields (see [IJS10] for instance). Recent proposals of verifiable delay functions based on class groups should also motivate research in this area (for example the Chia Network [Chi] has opened a competition for this).

Moreover, the bottleneck of our instantiation is the use of binary challenges in a zero knowledge proof of knowledge, used during key generation, in order to cope with the fact we are working in a cyclic subgroup of a group of unknown order and that we can not check that elements belong to the subgroup. There have been many proposals to deal with generalized Schnorr proofs in groups of unknown order (see for instance the framework of [CKY09] using safeguard groups, or [TW12]). For the case of subgroups of $(\mathbf{Z}/n\mathbf{Z})^\times$, efficient solutions for this type of proofs enlarge the challenge space, and rely on variants of the strong RSA assumption. For class groups, there have been informal proposals (see [DF02] for instance). However, computing square roots or finding elements of order 2 can be done efficiently in class groups knowing the factorization of the discriminant (which is public in our case). Moreover, as suggested in [BBF18], there may be other approaches to find low order elements in class groups. Advances in our understanding of class groups would lead to substantial efficiency improvements in several areas of cryptography.

Last but not least, our work focuses on the two party case. We believe that the ideas of our generic construction will lead to improvements in the general case of threshold ECDSA signatures. We leave this for future work.

Acknowledgements: The authors would like to thank Benoît Libert for fruitful discussions. This work was supported by the Universita' degli Studi di Catania,"Piano della Ricerca 2016/2018 Linea di intervento 2", and the French ANR ALAMBIC project (ANR-16-CE39-0006).

F.A. A brief definition of interactive zero-knowledge proofs

A zero-knowledge proof system (P, V) for a language $\mathscr{L}$ is an interactive protocol between two probabilistic algorithms: a prover P and a polynomial-time verifier V. Informally P — who detains a witness for a given statement — must convince V that the statement is true without revealing anything other than the truth of this statement to V.

Specifically, if $\mathscr{L}$ is a language, $x \in \mathscr{L}$ is a *true* statement while $x \notin \mathscr{L}$ is a *false* statement; and $1 \leftarrow (\mathrm{P},\mathrm{V})(x)$ (resp. $0 \leftarrow (\mathrm{P},\mathrm{V})(x)$) denotes the case when V interacting with P accepts (resp. rejects) the proof, the following properties must hold:

- *Completeness*: for any $x \in \mathscr{L}$:

$$\Pr[1 \leftarrow (\mathrm{P},\mathrm{V})(x)] > 1/2$$

- *Soundness*: for any prover P^* and for any $x \notin \mathscr{L}$:

$$\Pr[0 \leftarrow (\mathrm{P}^*,\mathrm{V})(x)] > 1/2$$

- *Zero-knowledge*: for every probabilistic polynomial time verifier V^*, there exists a probabilistic simulator *Sim* running in expected polynomial time such that for every $x \in \mathscr{L}$,

$$(\mathrm{P}, \mathrm{V}^*)(x) \equiv Sim(x).$$

$(\mathrm{P}, \mathrm{V})(x)$ is a random variable representing the output of V at the end of an interaction with P, then the zero-knowledge property holds if for any probabilistic polynomial time V^*, the output of V^* after an interaction with P is the same one of the simulator.

For a full explanation on this model see [Gol01] for interactive proofs and [GMR89] for zero-knowledge.

F.B. A technical Lemma on Distributions

Lemma F – 2. *Let X be a discrete random variable at statistical distance ϵ from the uniform distribution over $\mathbf{Z}/ab\mathbf{Z}$ for positive integers a and b such that $\gcd(a, b) = 1$. And let X_a (resp. X_b) be the random variable defined as $X_a := X \mod a$ (resp. $X_b := X \mod b$). Then the random variables X_a and X_b are less than ϵ close to the uniform distributions in $\mathbf{Z}/a\mathbf{Z}$ and $\mathbf{Z}/b\mathbf{Z}$ respectively. Moreover, even knowing X_b, X_a remains at statistical distance less than 2ϵ of the uniform distribution in $\mathbf{Z}/a\mathbf{Z}$ (and vice versa).*

Proof. Let $\mathscr{C}$ be an algorithm which takes as input a tuple $(a, b, x) \in \mathbf{N}^2 \times \mathbf{Z}/ab\mathbf{Z}$, which can either be a sample of the distribution:

$$\mathscr{U} := \{a, b, x | \gcd(a, b) = 1 \wedge x \xleftarrow{\$} \mathbf{Z}/ab\mathbf{Z}\}$$

or a sample of:

$$\mathscr{V} := \{a, b, x | \gcd(a, b) = 1 \ \wedge \ x \hookleftarrow \mathscr{D}\},$$

where $\mathscr{D}$ is a distribution at statistical distance ϵ of the uniform distribution over $\mathbf{Z}/ab\mathbf{Z}$, and outputs a bit. Since distributions $\mathscr{U}$ and $\mathscr{V}$ are at statistical distance ϵ, for any such algorithm $\mathscr{C}$, it holds that:

$$|\Pr[\mathscr{C}(\mathscr{U}) \to 1] - \Pr[\mathscr{C}(\mathscr{V}) \to 1]| \leq \epsilon.$$

We further denote $\mathscr{U}_{\mathscr{A}} := \{a, b, x_b, x_a | (a, b, x) \hookleftarrow \mathscr{V};\ x_b \leftarrow x \mod b;\ x_a \xleftarrow{\$} \mathbf{Z}/a\mathbf{Z}\}$ and $\mathscr{V}_{\mathscr{A}} := \{a, b, x_b, x_a \,|(a, b, x) \hookleftarrow \mathscr{V};\ x_b \leftarrow x \mod b;\ x_a \leftarrow x \mod a\}$.

Consider any algorithm $\mathscr{A}$ which takes as input a sample (a, b, x_b, x_a^*) of either $\mathscr{U}_{\mathscr{A}}$ or $\mathscr{V}_{\mathrm{A}}$, and outputs a bit β'. $\mathscr{A}$'s goal is to tell whether x_a^* is sampled uniformly at random from $\mathbf{Z}/a\mathbf{Z}$ or if $x_a^* \leftarrow x \mod a$. We demonstrate that if $\mathscr{A}$ has significant probability in distinguishing both input types, then $\mathscr{C}$ can use $\mathscr{A}$ to distinguish distributions $\mathscr{U}$ and $\mathscr{V}$. We describe the steps of $\mathscr{C}$ below:

```
$\underline{\mathscr{C}(a,b,x):}$
1. Set $x_b \leftarrow x \mod b$
2. Sample $\beta^* \overset{\$}{\leftarrow} \{0,1\}$
3. If $\beta^* = 0$, then $x_a^* \overset{\$}{\leftarrow} \mathbf{Z}/a\mathbf{Z}$
4. Else if $\beta^* = 1$, then $x_a^* \leftarrow x \mod a$
5. $\beta' \leftarrow \mathscr{A}(a,b,x_b,x_a^*)$
6. If $\beta = \beta'$ return 1
7. Else return 0.
```

If $\mathscr{C}$ gets as input an element of $\mathscr{U}$ whatever the value of β^*, x_a^* follows the uniform distribution modulo a and is independent of x_b. So $\mathscr{A}$'s success probability in outputting β' equal to β^* is 1/2.

$$\Pr[\mathscr{A}(a,b,x_b,x_a^*) \to \beta^* | (a,b,x) \hookleftarrow \mathscr{U}] = 1/2$$

and so

$$\Pr[\mathscr{C}(\mathscr{U}) \to 1] = 1/2$$

On the other hand if $(a,b,x) \hookleftarrow \mathscr{V}$, then $\mathscr{C}$ outputs 1 if $\mathscr{A}$ guesses the correct bit β^* (when its inputs are either in $\mathscr{U}_{\mathscr{A}}$ or $\mathscr{V}_{\mathscr{A}}$ as expected).

$$\Pr[\mathscr{C}(\mathscr{V}) \to 1] = \Pr[\mathscr{A} \to \beta^* | (a,b,x) \hookleftarrow \mathscr{V}]$$

And so

$$\begin{aligned}|\Pr[\mathscr{C}(\mathscr{U}) \to 1] - \Pr[\mathscr{C}(\mathscr{V}) \to 1]| &= \left|\Pr[\mathscr{A} \to \beta^* | (a,b,x) \hookleftarrow \mathscr{V}] - 1/2\right| \\ &= 1/2 \cdot |\Pr[\mathscr{A}(\mathscr{U}_{\mathscr{A}}) \to 1] - \Pr[\mathscr{A}(\mathscr{V}_{\mathrm{A}}) \to 1]|.\end{aligned}$$

Since distributions $\mathscr{U}$ and $\mathscr{V}$ are at statistical distance ϵ, it holds that $|\Pr[\mathscr{C}(\mathscr{U}) \to 1] - \Pr[\mathscr{C}(\mathscr{V}) \to 1]| \leq \epsilon$, and so for any algorithm $\mathscr{A}$ as above:

$$|\Pr[\mathscr{A}(\mathscr{U}_{\mathscr{A}}) \to 1] - \Pr[\mathscr{A}(\mathscr{V}_{\mathrm{A}}) \to 1]| \leq 2\epsilon.$$

Thus the statistical distance between $\mathscr{U}_{\mathscr{A}}$ and $\mathscr{V}_{\mathscr{A}}$ is smaller than 2ϵ, which implies that even given $x \mod b$, the value of $x \mod a$ remains at negligible statistical distance 2ϵ of the uniform distribution modulo a, which concludes the proof. □

F.C. Proof of Theorem F − 2: protocol of Figure F.8 is a ZKPoK

Proof. We prove completeness, soundness and zero-knowledge. Completeness follows easily by observing that when $((\mathsf{pk},(c_1,c_2),\mathrm{Q}_1);(x_1,r)) \in \mathrm{R}_{\mathsf{CL-DL}}$, for any $k \in \{0,1\}$ the values computed by an honest prover will indeed verify the four relations checked by the verifier. For soundness, the protocol is in fact special sound. Indeed notice that for committed values t_1, t_2, t_3, if a prover P^* can answer correctly for two different values of k, he must be able to answer to challenges 0 and 1 with u_1, u_2 and u_1', u_2', where u_1 and u_1' are smaller than $\mathrm{A} + \mathrm{S} - 1$, such that $u_2\mathrm{P} = u_2'\mathrm{P} - \mathrm{Q}$, $\mathsf{pk}^{u_1} f^{u_2} c_2 = \mathsf{pk}^{u_1'} f^{u_2'}$ and $g_q^{u_1} c_1 = g_q^{u_1'}$. Let $\sigma_1 \leftarrow u_1' - u_1$, $\sigma_2 \leftarrow u_2' - u_2 \mod q$; we obtain $g_q^{\sigma_1} = c_1$, $\sigma_2\mathrm{P} = \mathrm{Q}$ and $\mathsf{pk}^{\sigma_1} f^{\sigma_2'} = c_2$. Thus P^* can easily compute $x_1 \leftarrow \sigma_2 \mod q$ and $r \leftarrow \sigma_1$ in $\mathbf{Z}$.

While this gives a soundness error of 1/2, the soundness is amplified to $2^{-\ell}$ by repeating the protocol sequentially ℓ times.

For zero-knowledge, we must exhibit a simulator $\mathscr{S}$ which, given the code of some verifier V^*, produces a transcript indistinguishable from that which would be produced between V^* and an honest prover P (proving the knowledge of a tuple in $\mathrm{R}_{\mathsf{CL-DL}}$) without knowing the witnesses (x_1, r) for $(\mathsf{pk}, (c_1, c_2), \mathrm{Q}_1)$ in the relation $\mathrm{R}_{\mathsf{CL-DL}}$.

The potentially malicious verifier may use an adaptive strategy to bias the choice of the challenges to learn information about (r, x_1). This implies that challenges may not be randomly chosen, which must be taken into account in the security proof.

We describe an expected polynomial time simulation of the communication between a prover P and a malicious verifier V^* for one round of the proof. Since the simulated round may not be the first round, we assume V^* has already obtained data, denoted by hist, from previous interactions with P. Then P sends the commitments t_1, t_2, t_3 and V^* chooses – possibly using hist and t_1, t_2, t_3 – the challenge $k(t_1, t_2, t_3, \mathsf{hist})$.

Description of the simulator

Consider the simulator $\mathscr{S}$ which simulates a given round of identification as follows:

1. $\mathscr{S}$ chooses random values $\bar{k} \in \{0,1\}$, $\bar{u}_1 \in [\mathrm{S}-1, \mathrm{A}-1]$ and $\bar{u}_2 \in \mathbf{Z}/q\mathbf{Z}$.
2. $\mathscr{S}$ computes $\bar{t}_1 \leftarrow \mathsf{pk}^{\bar{u}_1} f^{\bar{u}_2} / c_2^{\bar{k}}$; $\bar{t}_2 \leftarrow [\bar{u}_2]\mathrm{P} - [\bar{k}]\mathrm{Q}$ and $\bar{t}_3 \leftarrow g_q^{\bar{u}_1} / c_1^{\bar{k}}$, and sends $\bar{t}_1$, $\bar{t}_2$ and $\bar{t}_3$ to V^*.
3. $\mathscr{S}$ receives $k(\bar{t}_1, \bar{t}_2, \bar{t}_3, \mathsf{hist})$ from V^*.
4. If $k(\bar{t}_1, \bar{t}_2, \bar{t}_3, \mathsf{hist}) \neq \bar{k}$ then return to step 1, else return $(\bar{t}_1, \bar{t}_2, \bar{t}_3, \bar{k}, \bar{u}_1, \bar{u}_2)$.

To demonstrate that the proof is indeed zero-knowledge, we need to justify that the distribution output by the simulator is statistically close to that output in a real execution of the protocol, and that the simulation runs in expected polynomial time.

Intuitively, sampling the randomness r from a large enough distribution – *i.e.*, as long as $\mathrm{S} << \mathrm{A}$ – ensures that the distribution of t_1, t_2, t_3 in a real execution is statistically close[6] to that in a simulated execution.

The analysis of the above statistical distance Σ between the distribution of tuples output by the simulator and that of tuples output by a real execution of the protocol is

[6] The distributions cannot be distinguished by any algorithm, even using an infinite computational power, but only accessing a polynomial number of triplets of both distributions

quite tedeous and similar to that of [GPS06]. We do not provide the details here but applying their analysis to our setting allows us obtain the following bound:

$$\Sigma < \frac{8S}{A}.$$

Thus the real and simulated distributions are statistically indistinguishable if S/A is negligible.

Running time of the Simulator

We now need to ensure that the simulator runs in expected polynomial time. To see this, notice that step 3 outputs a tuple $(\bar{t_1}, \bar{t_2}, \bar{t_3}, \bar{k}, \bar{u}_1, \bar{u}_2)$ if $k(\bar{t_1}, \bar{t_2}, \bar{t_3}, \mathsf{hist}) = \bar{k}$. Since $\bar{k}$ is sampled at random from $\{0,1\}$, the expected number of iterations of the loop is 2. Therefore the complexity of the simulation of all ℓ rounds is $O(\ell)$.

Thus if $\ell S/A$ is negligible and ℓ is polynomial, the protocol is statistically zero-knowledge. □

F.D. Lindell's new interactive assumption

In order to prove the security of his 2-party ECDSA, Lindell introduced in [Lin17] the following ad hoc interactive assumption, called Paillier-EC assumption. It is defined via the following random experiment.

```
**Experiment** $\mathbf{Exp}_{\mathcal{A}}(1^\lambda)$
$(pk, sk) \leftarrow \mathsf{Paillier.KeyGen}(1^\lambda)$
$(\omega_0, \omega_1) \xleftarrow{\$} \mathbf{Z}/q\mathbf{Z}$, $Q \leftarrow \omega_0 P$
$b^\star \xleftarrow{\$} \{0,1\}$, $c^\star \leftarrow \mathsf{Paillier.Enc}(1^\lambda, pk, \omega_{b^\star})$
$b \leftarrow \mathcal{A}^{\mathcal{O}_{c^\star}(\cdot,\cdot,\cdot)}(pk, c^\star, Q)$
if $b = b^\star$ then return 1
  else return 0
```

where $1 \leftarrow \mathcal{O}_{c^\star}(c', \alpha, \beta)$ if and only if $\mathsf{Paillier.Dec}(1^\lambda, sk, c') = \alpha + \beta\omega_{b^\star} \mod q$ and $\mathcal{O}$ stops after the first time it returns 0.

The Paillier-EC assumption is hard if for every probabilistic polynomial-time adversary $\mathcal{A}$ there exists a negligible function ν such that $\Pr[\mathbf{Exp}_{\mathcal{A}}(1^\lambda) = 1] \le \frac{1}{2} + \nu(n)$.

F.E. Two-Party ECDSA from HSM

P_1	$\mathit{KeyGen}(\mathbb{G}, \mathrm{P}, q, \widehat{\mathrm{G}}, g_q)$	P_2
$x_1 \overset{\$}{\leftarrow} \mathbf{Z}/q\mathbf{Z}$ $\mathrm{Q}_1 \leftarrow x_1\mathrm{P}$		
	$\xrightarrow{(\text{com−prove},1,\mathrm{Q}_1,x_1)} \mathscr{F}^{\mathrm{R}_{\mathrm{DL}}}_{\text{zk−com}} \xrightarrow{(\text{proof−receipt},1)}$	
P_1 aborts if (proof, 2, Q_2) not received.		$x_2 \overset{\$}{\leftarrow} \mathbf{Z}/q\mathbf{Z}$ $\mathrm{Q}_2 \leftarrow x_1\mathrm{P}$
	$\xleftarrow{(\text{proof},2,\mathrm{Q}_2)} \mathscr{F}^{\mathrm{R}_{\mathrm{DL}}}_{\text{zk}} \xleftarrow{(\text{prove},2,\mathrm{Q}_2,x_2)}$	
	$\xrightarrow{(\text{decom−proof},1)} \mathscr{F}^{\mathrm{R}_{\mathrm{DL}}}_{\text{zk−com}} \xrightarrow{(\text{decom−proof},1,\mathrm{Q}_1)}$	
$x, \rho \hookleftarrow \mathscr{D}_q$ $h \leftarrow g_q^x$		
$c_{key} = (c_{key,1}, c_{key,2}) \leftarrow (g_q^{\rho}, h^{\rho} f^{x_1})$	$\xrightarrow{(\text{prove},3,(h,c_{key},\mathrm{Q}_1),(x_1,\rho))} \mathrm{L}_{\text{CL−DL}} \xrightarrow{(\text{proof},3,(h,c_{key},\mathrm{Q}_1))}$	P_2 aborts unless (decomp − proof ,1, Q_1), (proof, 3, $(h, c_{key}, \mathrm{Q}_1)$) received and $h \in \widehat{\mathrm{G}}$.
$\mathrm{Q} \leftarrow x_1\mathrm{Q}_2$		$\mathrm{Q} \leftarrow x_2\mathrm{Q}_1$

P_1	$\mathit{Sign}(m, \text{sid})$	P_2
$k_1 \overset{\$}{\leftarrow} \mathbf{Z}/q\mathbf{Z}$ $\mathrm{R}_1 \leftarrow k_1\mathrm{P}$		
	$\xrightarrow{(\text{com−prove},\text{sid}\|1,\mathrm{R}_1,k_1)} \mathscr{F}^{\mathrm{R}_{\mathrm{DL}}}_{\text{zk−com}} \xrightarrow{(\text{proof−receipt},\text{sid}\|1)}$	
		$k_2 \overset{\$}{\leftarrow} \mathbf{Z}/q\mathbf{Z}$ $\mathrm{R}_2 \leftarrow k_2\mathrm{P}$
P_1 aborts if (proof, $\mathit{sid}\|2$, R_2) not received.	$\xleftarrow{(\text{proof},\text{sid}\|2,\mathrm{R}_2)} \mathscr{F}^{\mathrm{R}_{\mathrm{DL}}}_{\text{zk}} \xleftarrow{(\text{prove},\text{sid}\|2,\mathrm{R}_2,k_2)}$	
	$\xrightarrow{(\text{decom−proof},\text{sid}\|1)} \mathscr{F}^{\mathrm{R}_{\mathrm{DL}}}_{\text{zk−com}} \xrightarrow{(\text{decom−proof},\text{sid}\|1,\mathrm{R}_1)}$	P_2 aborts if (decomp − proof ,$\mathit{sid}\|1$, R_1) not received.
$\mathrm{R} = (r_x, r_y) = (r_x, r_y)k_1\mathrm{R}_2$		$m' \leftarrow \mathrm{H}(m)$ $\mathrm{R} = (r_x, r_y) \leftarrow k_2\mathrm{R}_1$ $r \leftarrow r_x \mod q$ $\tau \hookleftarrow \mathscr{D}_q$ $c_1 = (c_{1,1}, c_{1,2}) \leftarrow (g_q^{\tau}, h^{\tau} f^{k_2^{-1}m'})$ $c_2 = (c_{2,1}, c_{2,2}) \leftarrow (c_{key,1}^{k_2^{-1}rx_2}, c_{key,2}^{k_2^{-1}rx_2})$ $c_3 = (c_{3,1}, c_{3,2}) \leftarrow (c_{1,1}c_{2,1}, c_{1,2}c_{2,2})$
	$\xleftarrow{\quad c_3 \quad}$	
$\alpha \leftarrow \text{Solve}(c_{3,2}/c_{3,1}^x)$ $\hat{s} \leftarrow \alpha \cdot k_1^{-1}$ $s \leftarrow \min(\hat{s}, q - \hat{s})$ If not Verif(Q, m, (r, s)) P_1 aborts Else Return (r, s)		

Figure F.10: Two-Party ECDSA from HSM

Bibliography

[ABDP15] M. Abdalla, F. Bourse, A. De Caro, and D. Pointcheval. Simple functional encryption schemes for inner products. In *PKC 2015*, *LNCS* 9020, pages 733–751. Springer, Heidelberg, March / April 2015. 3, 37, 139, 140

[ABDP16] M. Abdalla, F. Bourse, A. De Caro, and D. Pointcheval. Better security for functional encryption for inner product evaluations. Cryptology ePrint Archive, Report 2016/011, 2016. http://eprint.iacr.org/2016/011. 139, 140

[ABP$^+$17] S. Agrawal, S. Bhattacherjee, D. H. Phan, D. Stehlé, and S. Yamada. Efficient public trace and revoke from standard assumptions: Extended abstract. In *ACM CCS 2017*, pages 2277–2293. ACM Press, October / November 2017. 37, 139

[ABSV15] P. Ananth, Z. Brakerski, G. Segev, and V. Vaikuntanathan. From selective to adaptive security in functional encryption. In *CRYPTO 2015, Part II*, *LNCS* 9216, pages 657–677. Springer, Heidelberg, August 2015. 138

[Adl94] L. M. Adleman. The function field sieve. In *Algorithmic Number Theory*, pages 108–121, Berlin, Heidelberg, 1994. Springer Berlin Heidelberg. 142

[AGVW13] S. Agrawal, S. Gorbunov, V. Vaikuntanathan, and H. Wee. Functional encryption: New perspectives and lower bounds. In *CRYPTO 2013, Part II*, *LNCS* 8043, pages 500–518. Springer, Heidelberg, August 2013. 144

[AJL$^+$12] G. Asharov, A. Jain, A. López-Alt, E. Tromer, V. Vaikuntanathan, and D. Wichs. Multiparty computation with low communication, computation and interaction via threshold FHE. In *EUROCRYPT 2012*, *LNCS* 7237, pages 483–501. Springer, Heidelberg, April 2012. 104

[AJMV98] E. Allender, J. Jiao, M. Mahajan, and V. Vinay. Non-commutative arithmetic circuits: depth reduction and size lower bounds. *Theoretical Computer Science*, 209(1):47 – 86, 1998. 105

[ALS16] S. Agrawal, B. Libert, and D. Stehlé. Fully secure functional encryption for inner products, from standard assumptions. In *CRYPTO 2016, Part III*, *LNCS* 9816, pages 333–362. Springer, Heidelberg, August 2016. 3, 37, 38, 139, 140, 141, 142, 143, 152, 153, 155, 156, 158, 161, 164, 167, 171, 172, 173

[ALSZ15] G. Asharov, Y. Lindell, T. Schneider, and M. Zohner. More efficient oblivious transfer extensions with security for malicious adversaries. In *EUROCRYPT 2015, Part I*, *LNCS* 9056, pages 673–701. Springer, Heidelberg, April 2015. 104

[AM94] L. M. Adleman and K. S. McCurley. Open problems in number theoretic complexity, ii. In *Proceedings of the First International Symposium on Algorithmic Number Theory*, ANTS-I, pages 291–322, London, UK, UK, 1994. Springer-Verlag. 64

[Arn14] F. Arnault. Formes quadratiques de discriminants emboîtés. *arXiv e-prints*, page arXiv:1402.0344, Feb 2014. 14

[BBBF18] D. Boneh, J. Bonneau, B. Bünz, and B. Fisch. Verifiable delay functions. In *CRYPTO 2018, Part I*, *LNCS* 10991, pages 757–788. Springer, Heidelberg, August 2018. 10, 40, 185, 202

[BBF18] D. Boneh, B. Bünz, and B. Fisch. A survey of two verifiable delay functions. Cryptology ePrint Archive, Report 2018/712, 2018. https://eprint.iacr.org/2018/712. 42, 211

[BBHM02] I. Biehl, J. Buchmann, S. Hamdy, and A. Meyer. A signature scheme based on the intractability of computing roots. *Designs, Codes and Cryptography*, 25(3):223–236, Mar 2002. 10, 41

[BBL17] F. Benhamouda, F. Bourse, and H. Lipmaa. CCA-secure inner-product functional encryption from projective hash functions. In *PKC 2017, Part II*, *LNCS* 10175, pages 36–66. Springer, Heidelberg, March 2017. 38, 41, 139, 142, 143

[BBT94] I. Biehl, J. Buchmann, and C. Thiel. Cryptographic protocols based on discrete logarithms in real-quadratic orders. In *CRYPTO'94*, *LNCS* 839, pages 56–60. Springer, Heidelberg, August 1994. 11

[BCP03] E. Bresson, D. Catalano, and D. Pointcheval. A simple public-key cryptosystem with a double trapdoor decryption mechanism and its applications. In *ASIACRYPT 2003*, *LNCS* 2894, pages 37–54. Springer, Heidelberg, November / December 2003. 24, 25, 27, 30, 81, 82, 83, 84, 85, 87, 91, 92, 93, 94, 146, 150

[BDH99] D. Boneh, G. Durfee, and N. Howgrave-Graham. Factoring N = prq for large r. In *CRYPTO'99*, *LNCS* 1666, pages 326–337. Springer, Heidelberg, August 1999. 47, 64, 75

[BDJ$^+$06] P. Bogetoft, I. Damgård, T. Jakobsen, K. Nielsen, J. Pagter, and T. Toft. A practical implementation of secure auctions based on multiparty integer computation. In *FC 2006*, *LNCS* 4107, pages 142–147. Springer, Heidelberg, February / March 2006. 104

[BDOZ11] R. Bendlin, I. Damgård, C. Orlandi, and S. Zakarias. Semi-homomorphic encryption and multiparty computation. In *EUROCRYPT 2011*, *LNCS* 6632, pages 169–188. Springer, Heidelberg, May 2011. 104

[BDW90] J. Buchmann, S. Düllmann, and H. C. Williams. On the complexity and efficiency of a new key exchange system. In *EUROCRYPT'89*, *LNCS* 434, pages 597–616. Springer, Heidelberg, April 1990. 9, 82, 99

[Bea92] D. Beaver. Foundations of secure interactive computing. In *CRYPTO'91*, *LNCS* 576, pages 377–391. Springer, Heidelberg, August 1992. 104

[Ben88] J. C. Benaloh. *Verifiable Secret-Ballot Election*. PhD thesis, Yale University, 1988. 23, 80

[Ber11] D. J. Bernstein. List decoding for binary goppa codes. In *Coding and Cryptology*, pages 62–80, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. 19, 65, 74

[BG10] A. Bernard and N. Gama. Smallest reduction matrix of binary quadratic forms. In *Algorithmic Number Theory*, pages 32–49, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. 21

[BGJS16] S. Badrinarayanan, V. Goyal, A. Jain, and A. Sahai. Verifiable functional encryption. In *ASIACRYPT 2016, Part II*, *LNCS* 10032, pages 557–587. Springer, Heidelberg, December 2016. 138

[BGN05] D. Boneh, E.-J. Goh, and K. Nissim. Evaluating 2-DNF formulas on ciphertexts. In *TCC 2005*, *LNCS* 3378, pages 325–341. Springer, Heidelberg, February 2005. 81

[BGS07] A. Bostan, P. Gaudry, and E. Schost. Linear recurrences with polynomial coefficients and application to integer factorization and cartier-manin operator. *SIAM J. Comput.*, 36(6):1777–1806, February 2007. 66, 75, 76

[BH01] J. Buchmann and S. Hamdy. A survey on IQ cryptography. In *Public Key Cryptography and Computational Number Theory*, pages 1–15. De Gruyter Proceedings in Mathematics, 2001. 9, 10, 82, 99, 203

[BH03] M. L. Bauer and S. Hamdy. On class group computations using the number field sieve. In *ASIACRYPT 2003*, *LNCS* 2894, pages 311–325. Springer, Heidelberg, November / December 2003. 8, 185

[Bia10] J.-F. Biasse. Improvements in the computation of ideal class groups of imaginary quadratic number fields. *Advances in Mathematics of Communications*, 4(2):141–154, 2010. 7

[BJS10] J.-F. Biasse, M. J. Jacobson, and A. K. Silvester. Security estimates for quadratic field based cryptosystems. In *ACISP 10*, *LNCS* 6168, pages 233–247. Springer, Heidelberg, July 2010. 7, 9, 82, 93, 98, 99, 142, 172, 204, 209

[BKV19] W. Beullens, T. Kleinjung, and F. Vercauteren. Csi-fish: Efficient isogeny based signatures through class group computations. Cryptology ePrint Archive, Report 2019/498, 2019. https://eprint.iacr.org/2019/498. 7, 40

[BLS12] R. Broker, K. Lauter, and A. V. Sutherland. Modular polynomials via isogeny volcanoes. *Mathematics of Computation*, 81(278):1201–1231, 2012. 13, 40

[BMM00] J. Buchmann, M. Maurer, and B. Möller. Cryptography based on number fields with large regulator. *Journal de théorie des nombres de Bordeaux*, 12(2):293–307, 2000. 11

[BO13] M. Bellare and A. O'Neill. Semantically-secure functional encryption: Possibility results, impossibility results and the quest for a general definition. In *CANS 13*, *LNCS* 8257, pages 218–234. Springer, Heidelberg, November 2013. 144

[Bou17] F. Bourse. *Functional Encryption for Inner-Product Evaluations*. PhD thesis, PSL Research University, France, 2017. 139, 142, 143

[Boy86] C. Boyd. Digital multisignature. *Cryptography and Coding*, pages 241–246, 1986. 182

[BPT04] I. Biehl, S. Paulus, and T. Takagi. Efficient undeniable signature schemes based on ideal arithmetic in quadratic orders. *Designs, Codes and Cryptography*, 31(2):99–123, Feb 2004. 14, 47, 48, 52, 61, 78

[Bre00] R. P. Brent. Public key cryptography with a group of unknown order. Technical report, Oxford University, Oxford, UK, UK, 2000. 81

[BST02] J. Buchmann, K. Sakurai, and T. Takagi. An IND-CCA2 public-key cryptosystem with fast decryption. In *ICISC 01*, *LNCS* 2288, pages 51–71. Springer, Heidelberg, December 2002. 14, 47, 48, 52, 55, 56, 61

[BSW11] D. Boneh, A. Sahai, and B. Waters. Functional encryption: Definitions and challenges. In *TCC 2011*, *LNCS* 6597, pages 253–273. Springer, Heidelberg, March 2011. 138, 143, 144

[BTV04] J. Buchmann, T. Takagi, and U. Vollmer. Number field cryptography. In *In High Primes and Misdemeanours: Lectures in Honour of the 60th birthday of Hugh Cowie Williams*, *Fields Institute Communications* 41, pages 111–125. AMS, 2004. 41, 64

[BTW95] J. Buchmann, C. Thiel, and H. Williams. Short representation of quadratic integers. In *Computational Algebra and Number Theory*, pages 159–185, Dordrecht, 1995. Springer Netherlands. 52, 54, 69, 70, 89

[BV07] J. Buchmann and U. Vollmer. *Binary Quadratic Forms. An Algorithmic Approach*. Springer, 2007. 5, 9, 10, 67, 68, 69, 70, 71, 82, 97, 98, 99

[BV14] Z. Brakerski and V. Vaikuntanathan. ”efficient fully homomorphic encryption from (standard) lwe”. *SIAM Journal on Computing*, 43(2):831–871, 2014. 81

[BW88] J. Buchmann and H. C. Williams. A key-exchange system based on imaginary quadratic fields. *Journal of Cryptology*, 1(2):107–118, June 1988. 1, 9, 46, 61, 64, 82, 99

[BW90] J. Buchmann and H. C. Williams. A key exchange system based on real quadratic fields. In *CRYPTO'89*, *LNCS* 435, pages 335–343. Springer, Heidelberg, August 1990. 10

[Can00] R. Canetti. Security and composition of multiparty cryptographic protocols. *Journal of Cryptology*, 13(1):143–202, January 2000. 104

[Cas08] G. Castagnos. Two generic constructions of probabilistic cryptosystems and their applications. In *SCN 08*, *LNCS* 5229, pages 92–108. Springer, Heidelberg, September 2008. 2

[CC87] D. V. Chudnovsky and G. V. Chudnovsky. *Approximations and complex multiplication according to Ramanujan*, pages 596–622. Springer New York, New York, NY, 1987. 75

[CC07] G. Castagnos and B. Chevallier-Mames. Towards a DL-based additively homomorphic encryption scheme. In *ISC 2007*, *LNCS* 4779, pages 362–375. Springer, Heidelberg, October 2007. 24, 80

[CC18] P. Chaidos and G. Couteau. Efficient designated-verifier non-interactive zero-knowledge proofs of knowledge. In *EUROCRYPT 2018, Part III*, *LNCS* 10822, pages 193–221. Springer, Heidelberg, April / May 2018. 33

[CCL$^+$19] G. Castagnos, D. Catalano, F. Laguillaumie, F. Savasta, and I. Tucker. Two-Party ECDSA from Hash Proof Systems and Efficient Instantiations. In *CRYPTO 19*, 2019. 3, 10, 23, 24, 27, 28, 30, 33, 35, 36, 37, 41, 181

[CDJ$^+$16] D. Chaum, D. Das, F. Javani, A. Kate, A. Krasnova, J. de Ruiter, and A. T. Sherman. cmix: Anonymization by high-performance scalable mixing. *25th USENIX Security Sym-posium*, 2016. 33

[CDN01] R. Cramer, I. Damgård, and J. B. Nielsen. Multiparty computation from threshold homomorphic encryption. In *EUROCRYPT 2001*, *LNCS* 2045, pages 280–299. Springer, Heidelberg, May 2001. 34, 104, 105, 109, 131

[CF15] D. Catalano and D. Fiore. Using linearly-homomorphic encryption to evaluate degree-2 functions on encrypted data. In *ACM CCS 2015*, pages 1518–1529. ACM Press, October 2015. 82

[CGS97] R. Cramer, R. Gennaro, and B. Schoenmakers. A secure and optimally efficient multi-authority election scheme. In *EUROCRYPT'97*, *LNCS* 1233, pages 103–118. Springer, Heidelberg, May 1997. 94

[CH89] R. A. Croft and S. P. Harris. Public-key cryptography and reusable shared secret. *Cryptography and Coding*, pages 189–201, 1989. 182

[Chi] Chia. https://www.chia.net/. 40, 210

[Chi89] A. L. Chistov. The complexity of constructing the ring of integers of a global field. *Dolk. Akad. Nauk. SSSR*, 306:1063–1067, 1989. 64

[CHN99] J.-S. Coron, H. Handschuh, and D. Naccache. ECC: Do we need to count? In *ASIACRYPT'99*, *LNCS* 1716, pages 122–134. Springer, Heidelberg, November 1999. 81

[CIL17] G. Castagnos, L. Imbert, and F. Laguillaumie. Encryption switching protocols revisited: Switching modulo p. In *CRYPTO 2017, Part I*, *LNCS* 10401, pages 255–287. Springer, Heidelberg, August 2017. 3, 27, 33, 34, 35, 103, 142, 175, 176, 185, 202

[CJLN09] G. Castagnos, A. Joux, F. Laguillaumie, and P. Q. Nguyen. Factoring pq^2 with quadratic forms: Nice cryptanalyses. In *ASIACRYPT 2009*, *LNCS* 5912, pages 469–486. Springer, Heidelberg, December 2009. 2, 3, 5, 7, 13, 17, 18, 19, 20, 21, 63, 81

[CKY09] J. Camenisch, A. Kiayias, and M. Yung. On the portability of generalized Schnorr proofs. In *EUROCRYPT 2009*, *LNCS* 5479, pages 425–442. Springer, Heidelberg, April 2009. 41, 131, 185, 211

[CL84] H. Cohen and H. W. Lenstra Jr. Heuristics on class groups. In *Number Theory*, pages 26–36, Berlin, Heidelberg, 1984. Springer Berlin Heidelberg. 76

[CL09] G. Castagnos and F. Laguillaumie. On the security of cryptosystems with quadratic decryption: The nicest cryptanalysis. In *EUROCRYPT 2009*, *LNCS* 5479, pages 260–277. Springer, Heidelberg, April 2009. 2, 3, 5, 12, 17, 18, 45, 64, 65, 69, 78, 81, 82, 89, 92, 141, 184

[CL12] G. Castagnos and F. Laguillaumie. Homomorphic encryption for multiplications and pairing evaluation. In *SCN 12*, *LNCS* 7485, pages 374–392. Springer, Heidelberg, September 2012. 81

[CL15] G. Castagnos and F. Laguillaumie. Linearly homomorphic encryption from DDH. In *CT-RSA 2015*, *LNCS* 9048, pages 487–505. Springer, Heidelberg, April 2015. 2, 3, 5, 9, 10, 13, 23, 24, 25, 26, 27, 28, 30, 35, 79, 106, 109, 120, 121, 122, 128, 130, 140, 141, 145, 146, 147, 149, 152, 171, 172, 174, 175, 176, 184, 185, 201, 202, 203, 204

[CLT18] G. Castagnos, F. Laguillaumie, and I. Tucker. Practical fully secure unrestricted inner product functional encryption modulo p. In *ASIACRYPT 2018, Part II*, *LNCS* 11273, pages 733–764. Springer, Heidelberg, December 2018. 3, 23, 24, 25, 27, 28, 30, 33, 38, 41, 137, 185, 201, 202, 205

[CNP99] J.-S. Coron, D. Naccache, and P. Paillier. Accelerating okamoto-uchiyama public-key cryptosystem. *Electronics Letters*, 35(4):291–292, Feb 1999. 48

[Coh00] H. Cohen. *A course in computational algebraic number theory*. Springer-Verlag, 2000. 5, 8, 49, 50, 55, 67, 98, 99, 172, 203

[Cop97] D. Coppersmith. Small solutions to polynomial equations, and low exponent RSA vulnerabilities. *Journal of Cryptology*, 10(4):233–260, September 1997. 19, 64, 65, 72

[Cox99] D. A. Cox. *Primes of the form* $x^2 + ny^2$. John Wiley & Sons, 1999. 5, 11, 12, 48, 49, 50, 51, 52, 67, 89, 100

[CP01] R. Crandall and C. Pomerance. *Prime numbers: a computational perspective.* Springer-Verlag, 2001. 65

[CPP06] B. Chevallier-Mames, P. Paillier, and D. Pointcheval. Encoding-free ElGamal encryption without random oracles. In *PKC 2006*, *LNCS* 3958, pages 91–104. Springer, Heidelberg, April 2006. 24, 80, 81, 87

[CPP16] G. Couteau, T. Peters, and D. Pointcheval. Encryption switching protocols. In *CRYPTO 2016, Part I*, *LNCS* 9814, pages 308–338. Springer, Heidelberg, August 2016. 3, 34, 104, 105, 106, 107, 109, 113, 114, 116, 118, 124, 125, 128, 130, 131

[CS98] R. Cramer and V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *CRYPTO'98*, *LNCS* 1462, pages 13–25. Springer, Heidelberg, August 1998. 142, 150, 190

[CS02] R. Cramer and V. Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In *EUROCRYPT 2002*, *LNCS* 2332, pages 45–64. Springer, Heidelberg, April / May 2002. 28, 41, 142, 189, 190, 192

[CS03] J. Camenisch and V. Shoup. Practical verifiable encryption and decryption of discrete logarithms. In *CRYPTO 2003*, *LNCS* 2729, pages 126–144. Springer, Heidelberg, August 2003. 28, 30, 146, 150, 210

[CW05] K. Cheng and H. Williams. Some results concerning certain periodic continued fractions. *Acta Arithmetica*, 117:247–264, 01 2005. 15, 77

[Deg58] G. Degert. Über die bestimmung der grundeinheit gewisser reell quadratischer zahlkörper. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 22(1):92–97, Dec 1958. 65

[Des88] Y. Desmedt. Society and group oriented cryptography: A new concept. In *CRYPTO'87*, *LNCS* 293, pages 120–127. Springer, Heidelberg, August 1988. 182

[DF90] Y. Desmedt and Y. Frankel. Threshold cryptosystems. In *CRYPTO'89*, *LNCS* 435, pages 307–315. Springer, Heidelberg, August 1990. 108, 109, 182

[DF02] I. Damgård and E. Fujisaki. A statistically-hiding integer commitment scheme based on groups with hidden order. In *ASIACRYPT 2002*, *LNCS* 2501, pages 125–142. Springer, Heidelberg, December 2002. 42, 81, 131, 211

[DIJ$^+$13] A. De Caro, V. Iovino, A. Jain, A. O'Neill, O. Paneth, and G. Persiano. On the achievability of simulation-based security for functional encryption. In *CRYPTO 2013, Part II*, *LNCS* 8043, pages 519–535. Springer, Heidelberg, August 2013. 144

[DJ01] I. Damgård and M. Jurik. A generalisation, a simplification and some applications of Paillier's probabilistic public-key system. In *PKC 2001*, *LNCS* 1992, pages 119–136. Springer, Heidelberg, February 2001. 23, 80, 94

[DJS12] V. Dixon, M. J. Jacobson Jr., and R. Scheidler. Improved exponentiation and key agreement in the infrastructure of a real quadratic field. In *LATINCRYPT 2012*, *LNCS* 7533, pages 214–233. Springer, Heidelberg, October 2012. 11

[DJS19] P. Das, M. J. Jacobson Jr., and R. Scheidler. Improved efficiency of a linearly homomorphic cryptosystem. In *Codes, Cryptology and Information Security*. Springer, To appear, 2019. 31, 40

[DK02] I. Damgård and M. Koprowski. Generic lower bounds for root extraction and signature schemes in general groups. In *EUROCRYPT 2002*, *LNCS* 2332, pages 256–271. Springer, Heidelberg, April / May 2002. 131

[DKLs18] J. Doerner, Y. Kondi, E. Lee, and a. shelat. Secure two-party threshold ECDSA from ECDSA assumptions. In *2018 IEEE Symposium on Security and Privacy*, pages 980–997. IEEE Computer Society Press, May 2018. 35, 182, 184

[DKLs19] J. Doerner, Y. Kondi, E. Lee, and a. shelat. Threshold ECDSA from ECDSA assumptions: The multiparty case. In *2019 IEEE Symposium on Security and Privacy*, pages 416–431. IEEE Computer Society Press, may 2019. 41, 182

[DM10] I. Damgård and G. L. Mikkelsen. Efficient, robust and constant-round distributed RSA key generation. In *TCC 2010*, *LNCS* 5978, pages 183–200. Springer, Heidelberg, February 2010. 122, 136

[DN03] I. Damgård and J. B. Nielsen. Universally composable efficient multiparty computation from threshold homomorphic encryption. In *CRYPTO 2003*, *LNCS* 2729, pages 247–264. Springer, Heidelberg, August 2003. 104

[DT06] I. Damgård and R. Thorbek. Linear integer secret sharing and distributed exponentiation. In *PKC 2006*, *LNCS* 3958, pages 75–90. Springer, Heidelberg, April 2006. 42

[DZ13] I. Damgård and S. Zakarias. Constant-overhead secure computation of Boolean circuits using preprocessing. In *TCC 2013*, *LNCS* 7785, pages 621–641. Springer, Heidelberg, March 2013. 104

[ElG85] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31:469–472, 1985. 104

[FGK$^+$13] D. M. Freeman, O. Goldreich, E. Kiltz, A. Rosen, and G. Segev. More constructions of lossy and correlation-secure trapdoor functions. *Journal of Cryptology*, 26(1):39–74, January 2013. 41

[Fon08] F. Fontein. The infrastructure of a global field of arbitrary unit rank. *Mathematics of Computation*, 80:2325–2357, 09 2008. 9

[FP03] P.-A. Fouque and G. Poupard. On the security of RDSA. In *EUROCRYPT 2003*, *LNCS* 2656, pages 462–476. Springer, Heidelberg, May 2003. 10

[FPL16] FPLLL development team. *fplll, a lattice reduction library*, 2016. Available at https://github.com/fplll/fplll. 78

[FPS01] P.-A. Fouque, G. Poupard, and J. Stern. Sharing decryption in the context of voting or lotteries. In *FC 2000*, *LNCS* 1962, pages 90–104. Springer, Heidelberg, February 2001. 109

[Gal02] S. D. Galbraith. Elliptic curve Paillier schemes. *Journal of Cryptology*, 15(2):129–138, March 2002. 40, 80

[Gen09] C. Gentry. Fully homomorphic encryption using ideal lattices. In *41st ACM STOC*, pages 169–178. ACM Press, May / June 2009. 80, 81

[GG18] R. Gennaro and S. Goldfeder. Fast multiparty threshold ECDSA with fast trustless setup. In *ACM CCS 2018*, pages 1179–1194. ACM Press, October 2018. 41, 182

[GGHZ16] S. Garg, C. Gentry, S. Halevi, and M. Zhandry. Functional encryption without obfuscation. In *TCC 2016-A, Part II*, *LNCS* 9563, pages 480–511. Springer, Heidelberg, January 2016. 138

[GGN16] R. Gennaro, S. Goldfeder, and A. Narayanan. Threshold-optimal DSA/ECDSA signatures and an application to bitcoin wallet security. In *ACNS 16*, *LNCS* 9696, pages 156–174. Springer, Heidelberg, June 2016. 35, 41, 42, 182

[GJKR96] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin. Robust threshold DSS signatures. In *EUROCRYPT'96*, *LNCS* 1070, pages 354–371. Springer, Heidelberg, May 1996. 182

[Gjø05] K. Gjøsteen. Symmetric subgroup membership problems. In *PKC 2005*, *LNCS* 3386, pages 104–119. Springer, Heidelberg, January 2005. 140, 146, 147

[GKP+13a] S. Goldwasser, Y. T. Kalai, R. A. Popa, V. Vaikuntanathan, and N. Zeldovich. How to run turing machines on encrypted data. In *CRYPTO 2013, Part II*, *LNCS* 8043, pages 536–553. Springer, Heidelberg, August 2013. 138

[GKP+13b] S. Goldwasser, Y. T. Kalai, R. A. Popa, V. Vaikuntanathan, and N. Zeldovich. Reusable garbled circuits and succinct functional encryption. In *45th ACM STOC*, pages 555–564. ACM Press, June 2013. 138

[GM84] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984. 23, 80, 105, 106, 124

[GM05] S. D. Galbraith and J. F. McKee. Pairings on elliptic curves over finite commutative rings. In *Cryptography and Coding*, pages 392–409, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. 40

[GM09] G. Gavin and M. Minier. Oblivious multi-variate polynomial evaluation. In *INDOCRYPT 2009*, *LNCS* 5922, pages 430–442. Springer, Heidelberg, December 2009. 105

[GMR89] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989. 212

[GMW87] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game or A completeness theorem for protocols with honest majority. In *19th ACM STOC*, pages 218–229. ACM Press, May 1987. 104

[Gol01] O. Goldreich. *Foundations of Cryptography: Basic Tools*. Cambridge University Press, Cambridge, UK, 2001. 212

[GPS06] M. Girault, G. Poupard, and J. Stern. On the fly authentication and signature schemes based on groups of unknown order. *Journal of Cryptology*, 19(4):463–487, October 2006. 185, 206, 215

[GPV08] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *40th ACM STOC*, pages 197–206. ACM Press, May 2008. 135, 145, 174

[GVW12] S. Gorbunov, V. Vaikuntanathan, and H. Wee. Functional encryption with bounded collusions via multi-party computation. In *CRYPTO 2012*, *LNCS* 7417, pages 162–179. Springer, Heidelberg, August 2012. 138

[GW08] J. E. Gower and S. S. Wagstaff. Square form factorization. *Math. Comput.*, 77:551–588, 2008. 65

[Gé18] A. Gélin. On the complexity of class group computations for large degree number fields. *arXiv e-prints*, page arXiv:1810.11396, Oct 2018. 41

[HJPT98] D. Hühnlein, M. J. Jacobson Jr., S. Paulus, and T. Takagi. A cryptosystem based on non-maximal imaginary quadratic orders with fast decryption. In *EUROCRYPT'98*, *LNCS* 1403, pages 294–307. Springer, Heidelberg, May / June 1998. 12, 14, 46, 48, 50, 52, 57, 61, 90, 92, 99

[HJW03] D. Hühnlein, M. J. Jacobson Jr., and D. Weber. Towards practical non-interactive public-key cryptosystems using non-maximal imaginary quadratic orders. *Designs, Codes and Cryptography*, 30(3):281–299, Nov 2003. 52

[HL10] C. Hazay and Y. Lindell. *Efficient Secure Two-Party Protocols: Techniques and Constructions*. Springer, 1st edition, 2010. 188

[HM89] J. L. Hafner and K. S. McCurley. A rigorous subexponential algorithm for computation of class groups. *J. Amer. Math. Soc.*, 2(4):837–850, 1989. 7, 47

[HM00a] S. Hamdy and B. Möller. Security of cryptosystems based on class groups of imaginary quadratic orders. In *ASIACRYPT 2000*, *LNCS* 1976, pages 234–247. Springer, Heidelberg, December 2000. 8, 82, 98, 99

[HM00b] D. Hühnlein and J. Merkle. An efficient NICE-Schnorr-type signature scheme. In *PKC 2000*, *LNCS* 1751, pages 14–27. Springer, Heidelberg, January 2000. 14, 47, 48, 52, 61

[HMT98] D. Hühnlein, A. Meyer, and T. Takagi. Rabin and RSA analogues based on non-maximal imaginary quadratic orders. In *ICISC 98*, LNCS, pages 221–240. KIISC, December 1998. 15, 47, 61

[HO09] B. Hemenway and R. Ostrovsky. Lossy trapdoor functions from smooth homomorphic hash proof systems. *Electronic Colloquium on Computational Complexity (ECCC)*, 16:127, 01 2009. 41, 184, 191

[HO12] B. Hemenway and R. Ostrovsky. Extended-DDH and lossy trapdoor functions. In *PKC 2012*, *LNCS* 7293, pages 627–643. Springer, Heidelberg, May 2012. 27, 148

[How97] N. Howgrave-Graham. Finding small roots of univariate modular equations revisited. In *Crytography and Coding*, pages 131–142, Berlin, Heidelberg, 1997. Springer Berlin Heidelberg. 72

[How01] N. Howgrave-Graham. Approximate integer common divisors. In *Cryptography and Lattices*, pages 51–66, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg. 64

[HP01] D. Hühnlein and S. Paulus. On the implementation of cryptosystems based on real quadratic number fields. In *SAC 2000*, *LNCS* 2012, pages 288–302. Springer, Heidelberg, August 2001. 11

[HPT99] M. Hartmann, S. Paulus, and T. Takagi. NICE - new ideal coset encryption. In *CHES'99*, *LNCS* 1717, pages 328–339. Springer, Heidelberg, August 1999. 14, 45, 46, 48, 52, 61, 64, 76, 81, 90

[HT99] D. Hühnlein and T. Takagi. Reducing logarithms in totally non-maximal imaginary quadratic orders to logarithms in finite fields. In *ASIACRYPT'99*, *LNCS* 1716, pages 219–231. Springer, Heidelberg, November 1999. 15

[Hüh99] D. Hühnlein. Efficient implementation of cryptosystems based on non-maximal imaginary quadratic orders. In *SAC 1999*, *LNCS* 1758, pages 147–162. Springer, Heidelberg, August 1999. 14, 15, 46, 48, 51, 52, 61

[Hüh01] D. Hühnlein. Faster generation of NICE-Schnorr-type signatures. In *CT-RSA 2001*, *LNCS* 2020, pages 1–12. Springer, Heidelberg, April 2001. 14, 47, 48, 51, 52, 61

[IJS10] L. Imbert, M. J. Jacobson Jr., and A. Schmidt. Fast ideal cubing in imaginary quadratic number and function fields. *Advances in Mathematics of Communications*, 4(2):237–260, 2010. 40, 210

[Jac98] M. Jacobson, Jr. Applying sieving to the computation of quadratic class groups. *Mathematics of Computation*, 68:859–867, 05 1998. 7

[Jac00] M. J. Jacobson Jr. Computing discrete logarithms in quadratic orders. *Journal of Cryptology*, 13(4):473–492, September 2000. 98, 128, 172, 204

[Jac04] M. J. Jacobson Jr. The security of cryptosystems based on class semigroups of imaginary quadratic non-maximal orders. In *ACISP 04*, *LNCS* 3108, pages 149–156. Springer, Heidelberg, July 2004. 100

[JJ00] É. Jaulmes and A. Joux. A NICE cryptanalysis. In *EUROCRYPT 2000*, *LNCS* 1807, pages 382–391. Springer, Heidelberg, May 2000. 14, 45, 46, 47, 48, 61, 78, 100

[JL13] M. Joye and B. Libert. Efficient cryptosystems from 2^k-th power residue symbols. In *EUROCRYPT 2013*, *LNCS* 7881, pages 76–92. Springer, Heidelberg, May 2013. 23, 80

[JLW95] M. J. Jacobson Jr., R. F. Lukes, and H. C. Williams. An investigation of bounds for the regulator of quadratic fields. *Experimental Mathematics*, 4(3):211–225, 1995. 67

[Jou09] A. Joux. *Algorithmic Cryptanalysis*. CRC Press, 2009. 74

[JSW06] M. J. Jacobson Jr., R. Scheidler, and H. C. Williams. An improved real-quadratic-field-based key exchange procedure. *Journal of Cryptology*, 19(2):211–239, April 2006. 11

[JSW08] M. J. Jacobson Jr., R. Scheidler, and D. Weimer. An adaptation of the NICE cryptosystem to real quadratic orders. In *AFRICACRYPT 08*, *LNCS* 5023, pages 191–208. Springer, Heidelberg, June 2008. 1, 2, 15, 21, 62, 64, 65, 76, 77, 81

[JvdP02] M. J. Jacobson and A. J. van der Poorten. Computational aspects of nucomp. In *Algorithmic Number Theory*, pages 120–133, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg. 40

[JW09] M. J. Jacobson Jr. and H. C. Williams. *Solving the Pell Equation*. Springer, 2009. 5, 9, 11

[KAF$^+$10] T. Kleinjung, K. Aoki, J. Franke, A. K. Lenstra, E. Thomé, J. W. Bos, P. Gaudry, A. Kruppa, P. L. Montgomery, D. A. Osvik, H. J. J. te Riele, A. Timofeev, and P. Zimmermann. Factorization of a 768-bit RSA modulus. In *CRYPTO 2010*, *LNCS* 6223, pages 333–350. Springer, Heidelberg, August 2010. 7

[Kap73] P. Kaplan. Divisibilité par 8 du nombre des classes des corps quadratiques dont le 2-groupe des classes est cyclique, et réciprocité biquadratique. *J. Math. Soc. Japan*, 25(4):596–608, 10 1973. 98

[KDL$^+$17] T. Kleinjung, C. Diem, A. K. Lenstra, C. Priplata, and C. Stahlke. Computation of a 768-bit prime field discrete logarithm. In *EUROCRYPT 2017, Part I*, *LNCS* 10210, pages 185–201. Springer, Heidelberg, April / May 2017. 7

[KL14] J. Katz and Y. Lindell. *Introduction to Modern Cryptography, Second Edition*. Chapman & Hall/CRC, 2nd edition, 2014. 108

[Kle16] T. Kleinjung. Quadratic sieving. *Mathematics of Computation*, 85:1861–1873, 2016. 7

[KM03] H. Kim and S.-J. Moon. Public-key cryptosystems based on class semigroups of imaginary quadratic non-maximal orders. In *ACISP 03*, *LNCS* 2727, pages 488–497. Springer, Heidelberg, July 2003. 100

[KS08] V. Kolesnikov and T. Schneider. Improved garbled circuit: Free XOR gates and applications. In *ICALP 2008, Part II*, *LNCS* 5126, pages 486–498. Springer, Heidelberg, July 2008. 118

[KSW08] J. Katz, A. Sahai, and B. Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In *EUROCRYPT 2008*, *LNCS* 4965, pages 146–162. Springer, Heidelberg, April 2008. 139

[KY01] A. Kiayias and M. Yung. Secure games with polynomial expressions. In *Automata, Languages and Programming, 28th International Colloquium, ICALP 2001, Crete, Greece, July 8-12, 2001, Proceedings*, *Lecture Notes in Computer Science* 2076, pages 939–950. Springer, 2001. 105

[KY02] J. Katz and M. Yung. Threshold cryptosystems based on factoring. In *ASIACRYPT 2002*, *LNCS* 2501, pages 192–205. Springer, Heidelberg, December 2002. 125, 127, 131

[KY19] S. Katsumata and S. Yamada. Non-zero inner product encryption schemes from various assumptions: LWE, DDH and DCR. In *PKC 2019, Part II*, *LNCS* 11443, pages 158–188. Springer, Heidelberg, April 2019. 37

[Len87] H. W. Lenstra Jr. Factoring integers with elliptic curves. *Annals of Mathematics*, 126(3):649–673, 1987. 47

[Lin16] Y. Lindell. How to simulate it - A tutorial on the simulation proof technique. Cryptology ePrint Archive, Report 2016/046, 2016. http://eprint.iacr.org/2016/046. 187

[Lin17] Y. Lindell. Fast secure two-party ECDSA signing. In *CRYPTO 2017, Part II*, *LNCS* 10402, pages 613–644. Springer, Heidelberg, August 2017. 3, 35, 36, 181, 182, 183, 184, 185, 191, 192, 193, 195, 196, 207, 208, 210, 215

[Lip12] H. Lipmaa. Secure accumulators from euclidean rings without trusted setup. In *ACNS 12*, *LNCS* 7341, pages 224–240. Springer, Heidelberg, June 2012. 10, 42

[LL93] A. K. Lenstra and H. W. Lenstra Jr., editors. *The Development of the Number Field Sieve*, *Lecture Notes in Mathematics* 1554. Springer-Verlag, Berlin, 1993. 47

[LLL82] A. K. Lenstra, H. W. Lenstra Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, Dec 1982. 73

[LMS18] R. W. F. Lai, G. Malavolta, and D. Schröder. Homomorphic secret sharing for low degree polynomials. In *ASIACRYPT 2018, Part III*, *LNCS* 11274, pages 279–309. Springer, Heidelberg, December 2018. 33

[LN18] Y. Lindell and A. Nof. Fast secure multiparty ECDSA with practical distributed key generation and applications to cryptocurrency custody. In *ACM CCS 2018*, pages 1837–1854. ACM Press, October 2018. 41, 182

[LP11] Y. Lindell and B. Pinkas. Secure two-party computation via cut-and-choose oblivious transfer. In *TCC 2011*, *LNCS* 6597, pages 329–346. Springer, Heidelberg, March 2011. 104

[LPS08] Y. Lindell, B. Pinkas, and N. P. Smart. Implementing two-party computation efficiently with security against malicious adversaries. In *SCN 08*, *LNCS* 5229, pages 2–20. Springer, Heidelberg, September 2008. 104

[LT13] H. Lipmaa and T. Toft. Secure equality and greater-than tests with sublinear online complexity. In *ICALP 2013, Part II*, *LNCS* 7966, pages 645–656. Springer, Heidelberg, July 2013. 117

[LTSC14] H. W. Lim, S. Tople, P. Saxena, and E.-C. Chang. Faster secure arithmetic computation using switchable homomorphic encryption. Cryptology ePrint Archive, Report 2014/539, 2014. http://eprint.iacr.org/2014/539. 105

[Luc02] S. Lucks. A variant of the Cramer-Shoup cryptosystem for groups of unknown order. In *ASIACRYPT 2002*, *LNCS* 2501, pages 27–45. Springer, Heidelberg, December 2002. 150

[LZPL15] Y. Lu, R. Zhang, L. Peng, and D. Lin. Solving linear equations modulo unknown divisors: Revisited. In *ASIACRYPT 2015, Part I*, *LNCS* 9452, pages 189–213. Springer, Heidelberg, November / December 2015. 19

[Mar03] J. Martinet. *Perfect Lattices in Euclidean Spaces*. Grundlehren der mathematischen Wissenschaften 327. Springer-Verlag Berlin Heidelberg, 1 edition, 2003. 161, 170

[May10] A. May. Using LLL-reduction for solving RSA and factorization problems. ISC, pages 315–348. Springer, Heidelberg, 2010. 72

[McC89] K. S. McCurley. Cryptographic key distribution and computation in class groups. In *Number Theory and Applications (Proc. NATO Advanced Study Inst. on Number Theory and Applications, Banff, 1988)*, Boston, 1989. Kluwer. 27, 46, 61, 172

[McK99] J. McKee. Speeding fermat's factoring method. *Math. Comput.*, 68(228):1729–1737, October 1999. 66

[Mil07] J. Milan. Factoring Small Integers: An Experimental Comparison. working paper or preprint, November 2007. 66

[MNP01] A. Meyer, S. Neis, and T. Pfahler. First implementation of cryptographic protocols based on algebraic number fields. In *Information Security and Privacy*, pages 84–103, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg. 41

[MNPS04] D. Malkhi, N. Nisan, B. Pinkas, and Y. Sella. Fairplay - secure two-party computation system. In *Proceedings of the 13th USENIX Security Symposium, August 9-13, 2004, San Diego, CA, USA*, pages 287–302. USENIX, 2004. 104

[MR92] S. Micali and P. Rogaway. Secure computation (abstract). In *CRYPTO'91*, *LNCS* 576, pages 392–404. Springer, Heidelberg, August 1992. 104

[MR04a] P. D. MacKenzie and M. K. Reiter. Two-party generation of DSA signatures. *Int. J. Inf. Sec.*, 2(3-4):218–239, 2004. 35, 182, 183

[MR04b] D. Micciancio and O. Regev. Worst-case to average-case reductions based on Gaussian measures. In *45th FOCS*, pages 372–381. IEEE Computer Society Press, October 2004. 145

[MR07] D. Micciancio and O. Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM J. Comput.*, 37(1):267–302, 2007. 121, 135, 157, 161, 166, 170

[Ngu91] P. Q. Nguyen. *La Géométrie des Nombres en Cryptologie*. PhD thesis, École Normale Supérieure, 1991. 161, 170

[NNOB12] J. B. Nielsen, P. S. Nordholt, C. Orlandi, and S. S. Burra. A new approach to practical active-secure two-party computation. In *CRYPTO 2012*, *LNCS* 7417, pages 681–700. Springer, Heidelberg, August 2012. 104

[NP06] M. Naor and B. Pinkas. Oblivious polynomial evaluation. *SIAM J. Comput.*, 35(5):1254–1281, 2006. 105

[NS98] D. Naccache and J. Stern. A new public key cryptosystem based on higher residues. In *ACM CCS 98*, pages 59–66. ACM Press, November 1998. 23, 80

[Oka86] T. Okamoto. Fast public-key cryptosystem using congruent polynomial equations. *Electronics Letters*, 22(11):581–582, May 1986. 64

[Oka90] T. Okamoto. A fast signature scheme based on congruential polynomial operations. *IEEE Transactions on Information Theory*, 36(1):47–53, Jan 1990. 64

[O'N10] A. O'Neill. Definitional issues in functional encryption. Cryptology ePrint Archive, Report 2010/556, 2010. http://eprint.iacr.org/2010/556. 138, 144

[OP01] T. Okamoto and D. Pointcheval. REACT: Rapid Enhanced-security Asymmetric Cryptosystem Transform. In *CT-RSA 2001*, *LNCS* 2020, pages 159–175. Springer, Heidelberg, April 2001. 14, 47

[OU98] T. Okamoto and S. Uchiyama. A new public-key cryptosystem as secure as factoring. In *EUROCRYPT'98*, *LNCS* 1403, pages 308–318. Springer, Heidelberg, May / June 1998. 23, 64, 80

[Pai99] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *EUROCRYPT'99*, *LNCS* 1592, pages 223–238. Springer, Heidelberg, May 1999. 1, 23, 48, 80, 81, 82, 84, 87, 89, 104, 139, 146

[PAR18] PARI Group, Univ. Bordeaux. *PARI/GP version* 2.11.1, 2018. available from http://pari.math.u-bordeaux.fr/. 30, 40, 207

[Ped91] T. P. Pedersen. A threshold cryptosystem without a trusted party (extended abstract) (rump session). In *EUROCRYPT'91*, *LNCS* 547, pages 522–526. Springer, Heidelberg, April 1991. 122

[Per01] R. Peralta. Elliptic curve factorization using a "partially oblivious" function. In *Cryptography and Computational Number Theory*, pages 123–128, Basel, 2001. Birkhäuser Basel. 64

[PO96] R. Peralta and E. Okamoto. Faster factoring of integers of a special form. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E79.A(4):489–493, 1996. 64

[PSSW09] B. Pinkas, T. Schneider, N. P. Smart, and S. C. Williams. Secure two-party computation is practical. In *ASIACRYPT 2009*, *LNCS* 5912, pages 250–267. Springer, Heidelberg, December 2009. 104

[PT98] S. Paulus and T. Takagi. A generalization of the Diffie-Hellman problem and related cryptosystems allowing fast decryption. In *ICISC 98*, LNCS, pages 211–220. KIISC, December 1998. 1, 14, 46, 48, 52, 61, 64, 76

[PT00] S. Paulus and T. Takagi. A new public-key cryptosystem over a quadratic order with quadratic decryption time. *Journal of Cryptology*, 13(2):263–272, March 2000. 12, 14, 46, 47, 48, 50, 51, 52, 53, 55, 61, 64, 67, 76, 81

[PW08] C. Peikert and B. Waters. Lossy trapdoor functions and their applications. In *40th ACM STOC*, pages 187–196. ACM Press, May 2008. 41

[Say] M. Sayles. `https://github.com/maxwellsayles/libqform`. 40

[SBW91] R. Scheidler, J. Buchmann, and H. C. Williams. Implementation of a key exchange protocol using some real quadratic fields. In *EUROCRYPT'90*, *LNCS* 473, pages 98–109. Springer, Heidelberg, May 1991. 11

[SBW94] R. Scheidler, J. Buchmann, and H. C. Williams. A key-exchange protocol using real quadratic fields. *Journal of Cryptology*, 7(3):171–199, September 1994. 11

[Sch82] R. Schoof. Quadratic fields and factorization. *Computational Methods in Number Theory, MC-Tracts*, 154/155:235–286, 1982. 57, 66

[Sch90] C.-P. Schnorr. Efficient identification and signatures for smart cards. In *CRYPTO'89*, *LNCS* 435, pages 239–252. Springer, Heidelberg, August 1990. 47

[Sch91a] C.-P. Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, January 1991. 192

[Sch91b] A. Schönhage. Fast reduction and composition of binary quadratic forms. In *Proceedings of the 1991 International Symposium on Symbolic and Algebraic Computation*, ISSAC '91, pages 128–133, New York, NY, USA, 1991. ACM. 7, 94

[Sch08] R. Schoof. Computing arakelov class groups. In *Math. Sci. Res. Inst. Publ*, *Algorithmic number theory: lattices, number fields, curves and cryptograph* 44, page 447–495. Cambridge Univ. Press, 2008. 9

[Sep] Sepior. http://www.sepior.com. 35, 182

[Ser] I. D. P. Services. https://security.intuit.com/. 35, 182

[SG98] V. Shoup and R. Gennaro. Securing threshold cryptosystems against chosen ciphertext attack. In *EUROCRYPT'98*, *LNCS* 1403, pages 1–16. Springer, Heidelberg, May / June 1998. 182

[SG02] V. Shoup and R. Gennaro. Securing threshold cryptosystems against chosen ciphertext attack. *Journal of Cryptology*, 15(2):75–96, March 2002. 109

[Sha84] A. Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO'84*, *LNCS* 196, pages 47–53. Springer, Heidelberg, August 1984. 138

[Sho00] V. Shoup. Practical threshold signatures. In *EUROCRYPT 2000*, *LNCS* 1807, pages 207–220. Springer, Heidelberg, May 2000. 182

[SP05] D. Schielzeth and M. E. Pohst. On real quadratic number fields suitable for cryptography. *Experiment. Math.*, 14(2):189–197, 2005. 9, 82, 99

[SS10] A. Sahai and H. Seyalioglu. Worry-free encryption: functional encryption with public keys. In *ACM CCS 2010*, pages 463–472. ACM Press, October 2010. 138

[Str76] V. Strassen. Einige Resultate über Berechnungskomplexität. *Jahresbericht der Deutschen Mathematiker Vereinigung*, 78(1):1–8, 1976. 75

[SW05] A. Sahai and B. R. Waters. Fuzzy identity-based encryption. In *EUROCRYPT 2005*, *LNCS* 3494, pages 457–473. Springer, Heidelberg, May 2005. 138

[Tak98] T. Takagi. Fast RSA-type cryptosystem modulo $p^k q$. In *CRYPTO'98*, *LNCS* 1462, pages 318–326. Springer, Heidelberg, August 1998. 64

[Tho12] E. Thomé. *Algorithmic Number Theory and Applications to the Cryptanalysis of Cryptographical Primitives*. Habilitation à diriger des recherches, Université de Lorraine, December 2012. 128

[TJB13] T. Tassa, A. Jarrous, and Y. Ben-Ya'akov. Oblivious evaluation of multivariate polynomials. *J. Mathematical Cryptology*, 7(1):1–29, 2013. 105

[TSCS13] S. Tople, S. Shinde, Z. Chen, and P. Saxena. AUTOCRYPT: enabling homomorphic computation on servers to protect sensitive web content. In *ACM CCS 2013*, pages 1297–1310. ACM Press, November 2013. 105

[TW12] B. Terelius and D. Wikström. Efficiency limitations of S-protocols for group homomorphisms revisited. In *SCN 12*, *LNCS* 7485, pages 461–476. Springer, Heidelberg, September 2012. 42, 211

[Unb] Unboundtech. `https://www.unboundtech.com/`. 35, 182

[Van92] S. Vanstone. Responses to nist's proposal. *Communications of the ACM*, 35:50–52, July 1992. (communicated by John Anderson). 186

[VSBR83] L. G. Valiant, S. Skyum, S. Berkowitz, and C. Rackoff. Fast parallel computation of polynomials using few processors. *SIAM Journal on Computing*, 12(4):641–644, 1983. 104

[VV07] B. Vallée and A. Vera. Lattice reduction in two dimensions : analyses under realistic probabilistic models. In *Proc. of AofA'07*, pages 181–216, 2007. 56

[Wat15] B. Waters. A punctured programming approach to adaptively secure functional encryption. In *CRYPTO 2015, Part II*, *LNCS* 9216, pages 678–697. Springer, Heidelberg, August 2015. 138

[Wei04] D. Weimer. An adaptation of the nice cryptosystem to real quadratic orders. Master's thesis, Technische Universität Darmstadt, 2004. 12, 67, 77

[Wes19] B. Wesolowski. Efficient verifiable delay functions. In *EUROCRYPT 2019, Part III*, *LNCS* 11478, pages 379–407. Springer, Heidelberg, May 2019. 10, 40, 42, 185, 202

[WWD18] H. Wang, Z. Wang, and J. Domingo-Ferrer. Anonymous and secure aggregation scheme in fog-based public cloud computing. *Future Generation Computer Systems*, 78:712 – 719, 2018. 33

[WWP+11] L. Wang, L. Wang, Y. Pan, Z. Zhang, and Y. Yang. Discrete logarithm based additively homomorphic encryption and secure data aggregation. *Information Sciences*, 181(16):3308 – 3322, 2011. 80

[Yao82] A. C.-C. Yao. Protocols for secure computations (extended abstract). In *23rd FOCS*, pages 160–164. IEEE Computer Society Press, November 1982. 104

[YWPZ08] Q. Ye, H. Wang, J. Pieprzyk, and X.-M. Zhang. Efficient disjointness tests for private datasets. In *ACISP 08*, *LNCS* 5107, pages 155–169. Springer, Heidelberg, July 2008. 105