Factoring pq^2 with Quadratic Forms: Nice Cryptanalyses

Guilhem Castagnos[⋆1], Antoine Joux[2,3], Fabien Laguillaumie[4], and
Phong Q. Nguyen[5]

[1] Institut de Mathématiques de Bordeaux – Université Bordeaux 1
guilhem.castagnos@math.u-bordeaux1.fr
[2] PRISM – Université de Versailles St-Quentin-en-Yvelines
[3] DGA
antoine.joux@m4x.org
[4] GREYC – Université de Caen Basse-Normandie
fabien.laguillaumie@info.unicaen.fr
[5] INRIA and ENS, France
http://www.di.ens.fr/~pnguyen/

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 $\mathbb{Q}(\sqrt{p})$ is unusually small, just like in *real*-NICE.

Keywords: Public-key Cryptanalysis, Factorisation, Binary Quadratic Forms, Homogeneous Coppersmith's Root Finding, Lattices.

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

[⋆] This work was done while this author was with the PRISM – Université de Versailles.

was initiated by Buchmann and Williams' key exchange [BW88], and which includes NICE[1] cryptosystems [HPT99,PT99,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 $N \in \mathbb{N}$ as input, compute $(r, s) \in \mathbb{N}^2$ 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 (other than N) 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 Coppersmith's root finding method [Cop97]. Very recently, Castagnos and Laguillaumie [CL09] showed that the public key in the imaginary version [HPT99,PT99, 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, PT99, 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:

i. The hint is an ideal equivalent to a secret ideal of norm q^2 in 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].
ii. The hint is the knowledge that the *regulator* of the quadratic field $\mathbb{Q}(\sqrt{p})$ is unusually small, just like in REAL-NICE. Roughly speaking, the regulator is a

[1] for *New Ideal Coset Encryption*

real number which determines how “dense” the units of the ring of integers of the number field $\mathbb{Q}(\sqrt{p})$ 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 $N = pq^2$, 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 [Ber08] 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 $\tilde{O}(N^{1/4})$ (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).

2 Background on Quadratic Fields and Quadratic Forms

2.1 Quadratic Fields

Let $D \neq 0, 1$ be a squarefree integer and consider the quadratic number field $K = \mathbb{Q}(\sqrt{D})$. 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 $\mathbb{Z}$-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 $\mathbb{Z} + \omega_K\mathbb{Z}$, 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 $\mathcal{O} = \mathbb{Z} + f\omega_K\mathbb{Z}$. 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 $\mathrm{N}(\mathfrak{a}) = |\mathcal{O}_\Delta/\mathfrak{a}|$. A *fractional* ideal is a subset $\mathfrak{a} \subset K$ such that $d\mathfrak{a}$ is an ideal of $\mathcal{O}_\Delta$ for $d \in \mathbb{N}$. 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 $I(\mathcal{O}_\Delta, f)$ the subgroup of $I(\mathcal{O}_\Delta)$ 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 $R_\Delta = \log(\varepsilon_\Delta)$ is the *regulator* of $\mathcal{O}_\Delta$. The following important bounds on the regulator of a real quadratic field can be found in [JLW95]:

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

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 $\mathsf{REAL\text{-}NICE}$.

Proposition 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.*

i. If $\mathfrak{A}$ is an $\mathcal{O}_{\Delta_K}$-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.
ii. If $\mathfrak{a}$ is an $\mathcal{O}_{\Delta_f}$-ideal prime to f, then $\mathfrak{a}\mathcal{O}_{\Delta_K}$ is an $\mathcal{O}_{\Delta_K}$-ideal prime to f of the same norm.
iii. The map $\varphi_f : I(\mathcal{O}_{\Delta_f}, f) \to I(\mathcal{O}_{\Delta_K}, f)$, $\mathfrak{a} \mapsto \mathfrak{a}\mathcal{O}_{\Delta_K}$ is an isomorphism.

The map φ_f from Proposition 1 induces a surjection $\bar{\varphi}_f : C(\mathcal{O}_{\Delta_f}) \twoheadrightarrow C(\mathcal{O}_{\Delta_K})$ which can be efficiently computed (see [PT00]). In our settings, we will use a prime conductor $f = q$ and consider $\Delta_q = q^2\Delta_K$, 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])

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

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

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 $\mathsf{SL}_2(\mathbb{Z})$ (cf. Section 5.2 of [Coh00]). Moreover, quadratic forms are more suited to an algorithmic point of view. Every ideal $\mathfrak{a}$ of $\mathcal{O}_\Delta$ can be written as $\mathfrak{a} = m\left(a\mathbb{Z} + \frac{-b+\sqrt{\Delta}}{2}\mathbb{Z}\right)$ with $m \in \mathbb{Z}$, $a \in \mathbb{N}$ and $b \in \mathbb{Z}$ 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 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 $M \in \mathsf{SL}_2(\mathbb{Z})$ with $M = \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix}$, and $f = [a,b,c]$, a binary quadratic form, then $f.M$ 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 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 $\mathsf{SL}_2(\mathbb{Z})$. We will call *matrix of the reduction*, the matrix M such that $g = f.M$. The reduction procedure yields a *uniquely* determined reduced form in the class modulo $\mathsf{SL}_2(\mathbb{Z})$.

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

Definition 3. *The form $f = [a, b, c]$ of positive discriminant Δ is* 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 $\mathsf{SL}_2(\mathbb{Z})$. 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 4. *Let f be an indefinite binary quadratic form, the* cycle *of f is the sequence $(\rho^i(g))_{i\in\mathbb{Z}}$ 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 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 $\mathbf{1}_\Delta$.*

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 $M_k = \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} \in \mathsf{SL}_2(\mathbb{Z})$ 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 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 [CL09, Theorem 2] 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[1] (indeed, the root finding algorithm of Section 3 recovers roots up to $|\Delta_q|^{1/9}$):

Heuristic 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], $|M_k| < 2 \max\{q^2, (k^2 + p)/4\}/\sqrt{pq^2}$. As f_k is normalised, $|k| \leq q$ and $|M_k| < 2q/\sqrt{p} \approx |\Delta_q|^{1/6}$. Note that we cannot reach such a bound with our root finding algorithm. Experimentally, for random k, $|M_k|$ can be much smaller. For example, if the bit-size λ of p and q equals 100, the mean value of $|M_k|$ 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

[1] In the full version, we prove a probabilistic version of Heuristic 1.

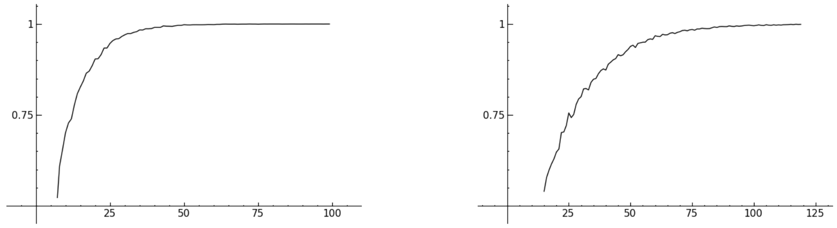
reduced forms. Applied to $\hat{f}_k = [a, b, c]$, this means that we expect a and c to be close to $|\Delta_q|^{1/2}$. Now, recall that $q^2 = \hat{f}_k(\delta, -\gamma) = a\delta^2 - b\delta\gamma + c\gamma^2$, which leads to δ and γ close to $\sqrt{q^2/a} = q/\sqrt{a} \approx q/|\Delta_q|^{1/4} \approx |\Delta_q|^{1/12}$. Thus, we expect that $|M_k| \leq |\Delta_q|^{1/12}$. And this explains why we obtained experimentally the bound $|\Delta_q|^{1/11.7}$. Figure 1(a) shows a curve obtained by experimentation, which gives the probability that $|M_k| < |\Delta_q|^{1/9}$ for random k, in function of λ. This curve also supports our heuristic.

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

Theorem 1. *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 $\varepsilon^t_{\Delta_K} = \varepsilon_{\Delta_q}$. Then the principal ideals of $\mathcal{O}_{\Delta_q}$ generated by $q\varepsilon^i_{\Delta_K}$ 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 $\varepsilon^i_{\Delta_K} \mod q$.*

Proof. Let $\alpha_i = q\varepsilon^i_{\Delta_K}$ 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 $\varepsilon^i_{\Delta_K} = x_i + y_i\omega_K$. Then $\alpha_i = qx_i + y_i q\Delta_K(1-q)/2 + y_i\frac{1}{2}(\Delta_q + \sqrt{\Delta_q})$, and α_i is an element of $\mathcal{O}_{\Delta_q}$. Let m_i, a_i and b_i be three integers such that $\mathfrak{a}_i = m_i\left(a_i\mathbb{Z} + \frac{-b_i+\sqrt{\Delta_q}}{2}\right)$. 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 $\mathbb{Z} + (\Delta_q + \sqrt{\Delta_q})/2\mathbb{Z}$, $\alpha_i\mathcal{O}_{\Delta_q}$ is generated by α_i and $\alpha_i(\Delta_q + \sqrt{\Delta_q})/2$ as a $\mathbb{Z}$-module. So a simple calculation gives that $m_i = \gcd(y_i, q(x_i + y_i\Delta_K/2))$. As $\varepsilon^i_{\Delta_K}$ is not an element of $\mathcal{O}_{\Delta_q}$, we have $\gcd(y_i, q) = 1$ so $m_i = \gcd(y_i, x_i + y_i\Delta_K/2)$. The same calculation to find m'_i for the ideal $\varepsilon^i_{\Delta_K}\mathcal{O}_{\Delta_K}$ reveals that $m_i = m'_i$. As $\varepsilon^i_{\Delta_K}\mathcal{O}_{\Delta_K} = \mathcal{O}_{\Delta_K}$ 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 $2qx_i + qy_i\Delta_K \equiv -b_i y_i \pmod{2a_i}$. 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 $k(i) \equiv -2x_i/y_i - \Delta_K \pmod q$. 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 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])



(a) Imaginary case

(b) Real case

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

which is close to the number of iterations of Rho between two forms. One has $d(\mathbf{1}_{\Delta_q}, f_{k(i)}) = iR_{\Delta_K}$. 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 $0 < jR_{\Delta_K} - 2\log q$, then as $jR_{\Delta_K} = d(f_{k(i)}, f_{k(i+j)}) = d(f_{k(i)}, \hat{f}_{k(i)}) + d(\hat{f}_{k(i)}, \hat{f}_{k(i+j)}) + d(\hat{f}_{k(i+j)}, f_{k(i+j)})$, from the triangle inequality, one has $jR_{\Delta_K} < 2\log(q) + |d(\hat{f}_{k(i)}, \hat{f}_{k(i+j)})|$. So, $|d(\hat{f}_{k(i)}, \hat{f}_{k(i+j)})| > jR_{\Delta_K} - 2\log q > 0$. 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 2, $d(\mathbf{1}_{\Delta_q}, \hat{f}_{k(i)}) \approx iR_{\Delta_K}$.

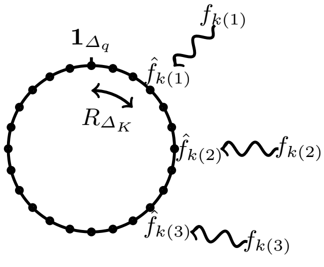


Fig. 2. Repartition of the forms $\hat{f}_{k(i)}$ along the principal cycle

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

Heuristic 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 $|M_k| < \Delta_q^{1/9}$, can be found in $\mathcal{O}(R_{\Delta_K})$ 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 $|M_k| < \Delta_q^{1/6.25}$ (resp. $|M_k| < \Delta_q^{1/5.44}$).

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 [May07]). Our problem is the following: Given $f(x,y) = x^2 + bxy + cy^2$ a (monic) binary quadratic form and $N = pq^2$ an integer of unknown factorisation, find $(x_0, y_0) \in \mathbb{Z}^2$ such that $f(x_0, y_0) \equiv 0 \pmod{q^2}$, while $|x_0|, |y_0| \leq M$, where $M \in \mathbb{N}$. 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 1. *Let $B \in \mathbb{N}$ and $g(x,y) \in \mathbb{Z}[x,y]$ be a homogeneous polynomial of total degree δ. Let $M > 0$ be a real number and suppose that $||g(x,y)|| < \frac{B}{\sqrt{\delta+1}M^\delta}$ then for all $x_0, y_0 \in \mathbb{Z}$ such that $g(x_0,y_0) \equiv 0 \pmod{B}$ and $|x_0|, |y_0| \leq M$, $g(x_0, y_0) = 0$.*

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

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

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

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

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

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

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

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 $y^{\delta t}, xy^{\delta t-1}, \dots, x^{\delta t-1}y, x^{\delta t}$. Let $\beta \in [0,1]$ such that $M = N^\beta$. The product of the diagonal elements gives $\det(L) = N^{\delta m(m+1)/2}$. If we omit the quantities that do not depend on N, to satisfy the inequality of Lemma 1 with the root bound M, the LLL bound from Lemma 2 implies that we must have

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

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 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 (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 $\beta \approx 1/5.44$.

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 [Ber08].

4 A $\tilde{O}(p^{1/2})$-Deterministic Factoring Algorithm for 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 $p \equiv 1 \pmod 4$.

4.1 The Algorithm

Roughly speaking, if $\Delta_q = N = pq^2$, our factoring algorithm, depicted in Fig. 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 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: $p, q$
1. $h \leftarrow \mathbf{1}_N$
2. while $(x_0, y_0)$ not found do
    2.1. $h \leftarrow \mathsf{Rho}(h)$
    2.2. $x_0/y_0 \leftarrow \mathsf{HomogeneousCoppersmith}(h, N, m, t)$
3. $q \leftarrow \mathsf{Sqrt}(\mathsf{Gcd}(h(x_0, y_0), N))$
4. return $(N/q^2, q)$
```

Fig. 3. Factoring $N = pq^2$

4.2 Heuristic Correctness and Analysis of Our Algorithm

Assuming Heuristic 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 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 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 $O\left(\mathsf{M}_{int}(\sqrt{b} \log N) + \log b \mathsf{M}_{int}(\log N) \log \log N\right)$ bit operations and space $O(\sqrt{b} \log N)$ bits.*

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.

5 Cryptanalysis of the NICE Cryptosystems

Hartmann, Paulus and Takagi proposed the elegant *NICE* encryption scheme (see [HPT99,PT99,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 $\varDelta_q = -pq^2$ 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.

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 $\varDelta_q = pq^2$, 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 [HPT99,JSW08].

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 $\varDelta_K = p$ and $\varDelta_q = pq^2$. They are chosen such that the ratio $R_{\varDelta_q}/R_{\varDelta_K}$ is of order of magnitude of q and that $R_{\varDelta_K}$ is bounded by a polynomial in $\log(\varDelta_K)$. To ensure the first property, it is sufficient to choose q such that $q - \left(\frac{\varDelta_K}{q}\right)$ 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 $\mathbb{Z}$, $a \neq 0$ 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 $\varDelta_q = pq^2$ 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 $\varDelta_q$. The message is carefully embedded (and padded) into a primitive $\mathcal{O}_{\varDelta_q}$-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 $\mathcal{O}_{\Delta_q}$-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 $\bar{\varphi}_q([\mathfrak{mh}])$. 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 1 *Algorithm 3 recovers the secret key of REAL-NICE in polynomial time in the security parameter under Heuristic 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 $S(X) = 2725^2X^2 + 2 \cdot 3815X + 2$ produces a suitable 256-bit prime p for the value $X_0 = 103042745825387139695432123167592199$. This prime has a regulator $R_{\Delta_K} \simeq 90.83$. The second 256-bit prime q is chosen following the recommendations from [Wei04]. This leads to a the discriminant

$$\begin{aligned}\Delta_q = \; & 287369388233100448733807161422820733961868439067574632747926387341440606028305 10\\ & 807386691634892735925990545294422710538698324853636823418921245006784003227198 42\\ & 63278692833860326257638544601057379571931906787755152745236263303465093\end{aligned}$$

Our algorithm recovers the prime

$$q = 6037210547149963441719285917385366345612301526720776965323555809278 1188395563$$

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 $-\frac{215551161171099644 5623}{354487427713477865 8948}$, where x_0 and y_0 satisfy $\frac{\log(\Delta_q)}{\log(|x_0|)} \simeq 10.8$ and $\frac{\log(\Delta_q)}{\log(|y_0|)} \simeq 10.7$.

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 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 3.

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

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

$$\begin{aligned}
\Delta_q &= -10011336194028467500739190370826191745653724259466749151493405394642199279551 68\\
&\quad 1821676008364075219870972619973270184386441185324964453536572880202249818566559 2\\
&\quad 983708546453282107912775914256762913490132215200222246716212360016561209 23
\end{aligned}$$

$$\begin{aligned}
a &= 57022687708942583181685884381175588713007831807699951950927158957551737003 99\\
&\quad 141486895731384747
\end{aligned}$$

$$\begin{aligned}
b &= 33612360405827547849585862980179491106487317456059301646668195696067550297 73\\
&\quad 074415823039847007
\end{aligned}$$

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

$$\begin{aligned}
h(x_0, y_0) &= 5363123171977038839829609999282338450991746328236957351089\\
&\quad 4245774887056120365979002534633233830227721465513935614971\\
&\quad 5939077126809522499818706407364011207 29 = q^2.
\end{aligned}$$

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 [Ste].

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.

References

[AM94] L. M. Adleman and K. S. McCurley. *Open problems in number theoretic complexity, II.* Proc. of ANTS-I, Springer LNCS Vol. 877, 291–322 (1994)

[BDH99] D. Boneh, G. Durfee and N. Howgrave-Graham. *Factoring $N = p^r q$ for large r.* Proc. of Crypto'99, Springer LNCS Vol. 1666, 326–337 (1999)

[Ber08] D. J. Bernstein. *List decoding for binary Goppa codes.* Preprint available at http://cr.yp.to/papers.html#goppalist (2008)

[BGS07] A. Bostan, P. Gaudry and É. Schost. *Linear Recurrences with Polynomial Coefficients and Application to Integer Factorization and Cartier-Manin Operator.* SIAM J. Comput., 36(6), 1777–1806 (2007)

[BPT04] I. Biehl, S. Paulus and T. Takagi. *Efficient Undeniable Signature Schemes based on Ideal Arithmetic in Quadratic Orders.* Des. Codes Cryptography 31(2), 99–123 (2004)

[BTV04] J. Buchmann, T. Takagi and U. Vollmer. *Number Field Cryptography.* High Primes & Misdemeanours: Lectures in Honour of the 60th Birthday of Hugh Cowie Williams, van der Poorten and Stein, eds., vol. 41 of Fields Institute Communications, AMS 111–125 (2004)

[BTW95] J. Buchmann, C. Thiel and H. C. Williams. *Short Representation of Quadratic Integers.* Proc. of CANT'92, Math. Appl. 325, Kluwer Academic Press, 159–185 (1995)

[BV07] J. Buchmann and U. Vollmer. *Binary Quadratic Forms. An Algorithmic Approach.* Springer (2007)

[BW88] J. Buchmann and H. C. Williams. *A Key-Exchange System based on Imaginary Quadratic Fields.* J. Cryptology, 1, 107–118 (1988)

[CC87] D. V. Chudnovsky and G. V. Chudnovsky, *Approximations and Complex Multiplication According to Ramanujan*, in Ramanujan Revisited: Proceedings, Boston, MA: Academic Press, 375–472, (1987)

[Chi89] A. L. Chistov. *The complexity of constructing the ring of integers of a global field.* Dolk. Akad. Nauk. SSSR, 306, 1063–1067 (1989). English translation: Soviet. Math. Dolk. 39, 597–600 (1989)

[CL84] H. Cohen and H.W. Lenstra, Jr. *Heuristics on class groups.* Springer LNM Vol. 1052, 26–36 (1984)

[CL09] G. Castagnos and F.Laguillaumie. *On the Security of Cryptosystems with Quadratic Decryption: The Nicest Cryptanalysis.* Proc. of Eurocrypt'09, Springer LNCS Vol. 5479, 260-277 (2009)

[Coh00] H. Cohen. *A Course in Computational Algebraic Number Theory.* Springer (2000).

[Cop97] D. Coppersmith. *Small Solutions to Polynomial Equations, and Low Exponent RSA Vulnerabilities.* J. Cryptology. 10 (4), 233–260 (1997)

[Cox99] D. A. Cox. *Primes of the form $x^2 + ny^2$.* John Wiley & Sons (1999)

[CP01] R. Crandall and C. Pomerance. *Prime Numbers: A Computational Perspective.* Springer (2001)

[CW05] K.H.F. Cheng and H.C. Williams. *Some Results Concerning Certain Periodic Continued Fractions.* Acta Arith. 117, 247–264 (2005)

[Deg58] G. Degert. *Uber die Bestimmung der Grundeinheit gewisser reell- quadratischer Zhalkörper.* Abh. Math. Sem. Univ. Hanburg, 22, 92–97 (1958)

[GW08] J. E. Gower and S. S. Wagstaff, Jr. *Square form factorization.* Math. Comput. 77(261), 551-588 (2008)

[How97] N. Howgrave-Graham. *Finding small roots of univariate modular equations revisited.* Proc. of IMA-C2, Springer LNCS Vol. 1355, 131–142 (1997)

[How01] N. Howgrave-Graham. *Approximate Integer Common Divisors* Proc. of CALC '01, Springer LNCS Vol. 2146, 51–66 (2001)

[HPT99] M. Hartmann, S. Paulus and T. Takagi. *NICE - New Ideal Coset Encryption.* Proc. of CHES'99, Springer LNCS Vol. 1717, 328–339 (1999)

[JJ00] É. Jaulmes and A. Joux. *A NICE Cryptanalysis.* Proc. of Eurocrypt'00, Springer LNCS Vol. 1807, 382–391 (2000)

[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

[Jou09] A. Joux. *Algorithmic Cryptanalysis.* CRC Press (2009)

[JSW08] M. J. Jacobson Jr., R. Scheidler and D. Weimer. *An Adaptation of the NICE Cryptosystem to Real Quadratic Orders.* Proc. of Africacrypt'08, Springer LNCS Vol. 5023, 191-208 (2008)

[LLL82] A. K. Lenstra, H. W. Lenstra Jr., and L. Lovász, *Factoring Polynomials with Rational Coefficients.* Math. Ann. 261, 515–534, (1982)

[May07] A. May. *Using LLL-Reduction for Solving RSA and Factorization Problems: A Survey.* LLL+25 Conference in honour of the 25th birthday of the LLL algorithm (2007).

[McK99] J. McKee. *Speeding Fermat's factoring method.* Math. Comput. 68(228), 1729–1737 (1999)

[Mil07] J. Milan. *Factoring Small Integers: An Experimental Comparison.* INRIA report available at http://hal.inria.fr/inria-00188645/en/ (2007)

[Oka86] T. Okamoto. *Fast public-key cryptosystem using congruent polynomial equations.* Electronic Letters 22(11), 581–582 (1986)

[Oka90] T. Okamoto. *A fast signature scheme based on congruential polynomial operations.* IEEE Transactions on Information Theory 36(1), 47–53 (1990)

[OU98] T. Okamoto and S. Uchiyama. *A New Public-Key Cryptosystem as Secure as Factoring.* Proc. of Eurocrypt'98, Springer LNCS Vol. 1403, 308–318 (1998)

[Per01] R. Peralta. *Elliptic curve factorization using a "partially oblivious" function.* Cryptography and computational number theory, Progr. Comput. Sci. Appl. Logic. 20. Birkhäuser, 123–128 (2001).

[PO96] R. Peralta and E. Okamoto. *Faster Factoring of Integers of a Special Form.* IEICE Trans. Fundamentals, E79-A, 4, 489–493 (1996).

[PT99] S. Paulus and T. Takagi. *A generalization of the Diffie-Hellman problem and related cryptosystems allowing fast decryption.* Proc. of ICISC'98, 211–220 (1999)

[PT00] S. Paulus and T. Takagi. *A New Public-Key Cryptosystem over a Quadratic Order with Quadratic Decryption Time.* J. Cryptology, 13(2), 263–272 (2000)

[Sch82] R. Schoof. *Quadratic fields and factorization.* Computational Methods in Number Theory, MC-Tracts 154/155, 235–286 (1982)

[Ste] D. Stehlé. *fplll-3.0.*
Available at http://perso.ens-lyon.fr/damien.stehle/#software

[Str76] V. Strassen. *Einige Resultate über Berechnungskomplexität.* Jber. Deutsch. Math.-Verein., 78, 1–8 (1976/77)

[Tak98] T. Takagi. *Fast RSA-Type Cryptosystem Modulo* $p^k q$. Proc. of Crypto'98, Springer LNCS Vol. 1462, 318–326 (1998)

[Wei04] D. Weimer. *An Adaptation of the NICE Cryptosystem to Real Quadratic Orders.* Master's thesis, Technische Universität Darmstadt (2004)