Lattice-based homomorphic encryption of vector spaces

Carlos Aguilar Melchor
XLIM-DMI, Université de Limoges,
123 av. Albert Thomas,
87000, Limoges, France
Email: carlos.aguilar@unilim.fr

Guilhem Castagnos
GREYC, ENSICAEN,
Boulevard Maréchal Juin, BP 5186,
14032 Caen CEDEX, France
Email: guilhem.castagnos@info.unicaen.fr

Philippe Gaborit
XLIM-DMI, Université de Limoges,
123 av. Albert Thomas,
87000, Limoges, France
Email: gaborit@unilim.fr

***Abstract*— In this paper we introduce a new probabilistic lattice-based bounded homomorphic encryption scheme. For this scheme the sum of two encrypted messages is the encryption of the sum of two messages and the scheme is able to preserve a vector space structure of the message. The size of the public key is rather large ≈ 3Mb but the encryption and the decryption operations are very fast (of the same speed order than NTRU). The homomorphic operation, *i.e.* the addition of ciphertexts is dramatically fast compared to homomorphic schemes based on group theory like Paillier or Elgamal.**

I. Introduction

In 1982, Goldwasser and Micali proposed the first probabilistic cryptosystem and defined the adequate notion of security for this type of scheme: the notion of semantic security. After this system, based on quadratic residuosity, many probabilistic schemes following the same principle have been proposed: chronologically by Benaloh, Naccache and Stern, Okamoto and Uchiyama, and at last, the most achieved system has been proposed by Paillier, and then generalized by Damgård and Jurik (see [1] for all references).

All these schemes use quotients of Z. Their one-wayness is based on factoring and their semantic security is based on distinguishing prime residues. A specially interesting property of these schemes is that they are homomorphic. Indeed, if c_i is a valid encryption of m_i, with $i \in \{1,\ 2\}$, one can publicly compute a valid ciphertext of the message $m_1 + m_2$. For these schemes, this is done by computing the modular multiplication $c_1 c_2$, whose cost is a quadratic function of the modulus size. This property has many applications, for example, the systems of Paillier and Damgård and Jurik can be used to design electronic vote systems [1], for Private Information Retrieval [2], or for building Mix-nets [3].

In this paper we propose a new lattice based encryption system with bounded homomorphic property (in the sense that the number of possible homomorphic operations, even if large, is bounded). This scheme has a large public key but has the following features: 1) the scheme is the first non number theory based with homomorphic property which preserves vectorial structure, 2) the scheme is faster by an order 100-1000 than previously known homomorphic schemes, 3) the scheme is the first homomorphic scheme truly additive for both the message and the encryption.

More precisely our scheme operates over vectors instead of integers. We will say that a cryptosystem is (ℓ, r, N)-homomorphic if two conditions are met. First, a plaintext message will be a N-tuple of elements of a ring of characteristic r. Second, up to ℓ publicly computed ciphertexts can be combined to get a valid ciphertext. For example, Paillier's cryptosystem is $(\infty, N_{RSA}, 1)$-homomorphic. This property offers flexibility as one can adjust the parameters N, r and ℓ in order to fit very different applications. For example, for a multi-candidate election system with N candidates and r voters, a voter will vote for the i^{th} candidate by encrypting a N-tuple of the form $(0, 0, \ldots, 0, 1, 0, \ldots, 0)$, where the 1 is in the i^{th} position. It is easy to extend this process to more complex vote protocols like party list elections where voters choose for candidates that belong to the same set out of several sets.

Our scheme is based on noisy lattices and is very efficient from a computational point of view, as well for encryption as for decryption, and specially for the homomorphic operation which is a simple addition in a vector space over a finite field. This scheme is based on the same assumptions and problems as the Private Information Retrieval protocol presented in [4] by Aguilar and Gaborit. In particular, we prove that our scheme is semantically secure in the standard model under one of these assumption, where as this level of security is not achieved by NTRU (indistinguishability is only possible in the random oracle model, cf. [5]).

II. Description of the scheme

A. High-level overview

The encryption scheme we propose relies on the simple idea of controlled noise addition. The main idea is to start from a secret random $N \times 2N$ matrix M of rank N over a field $GF(p)$ and to hide the subspace it represents. This matrix is used to generate a set of different matrices obtained by multiplication on the left side by invertible random matrices. These matrices (which can also be seen as lattices by joining pI_{2N} for I_{2N} the identity $2N \times 2N$ matrix) are disturbed by the user by the introduction of noise in half of the matrices' columns (as shown in figure 1) to obtain respectively softly disturbed matrices (SDMs) and a hardly disturbed matrices (HDM).

The public key is composed of one HDM and n SDMs. To encrypt a vector, the user multiplies it by the HDM. Then, for each of the SDMs he generates a random vector with small coordinates and multiplies it by the corresponding matrix. Finally, he adds all the results. The encrypted message is hence an element of the hidden subspace added with the (large) noise induced by the encrypted message and the (small) noise induced by the SDMs. Using the knowledge of the hidden subspace matrix and the position of the unmodified columns

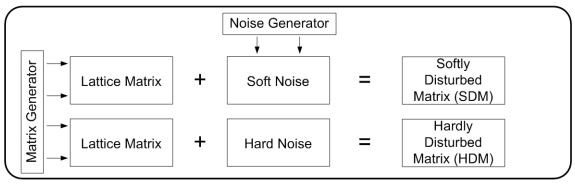

Fig. 1. Scheme overview.

of the HDM and SDMs, one can recover the noise associated to the encrypted message. From it, one separates the hard noise induced by the encrypted message from the smaller noise induced by the SDMs.

The scheme uses the same kind of idea as for the lattice-based NTRU cryptosystem: one considers a vector space over a field $GF(p)$ where the key idea is to control an error by keeping it non altered by any modular operation. The homomorphic properties of the scheme come directly from additive properties of the lattices. More particularly, the addition of two encrypted messages is the sum of two vectors of the hidden subspace plus two hard noises induced by the encrypted messages and two small noises induced by the SDMs. Choosing *ad hoc* parameters ensures that the small noise remains distinguishable from the hard noise.

B. Key generation

The scheme will have five global integer parameters: N, the number of coordinates of the plaintext vectors, r the characteristic of the ring over which they are constructed, ℓ the maximum number of homomorphic operations that can be done, n, the number of SDMs used for the public key, and ϵ_{max}, an upper bound for the coordinates of the random vectors used to insert noise.

Key generation

1) Note $l_0 = n \times N \times \epsilon_{max} + (N-1) \times r$ and set q as $2 \times l_0 \times (2\ell + 1)$ and p as a prime such that $p = q \times r + \epsilon$ with $\epsilon < l_0$.
2) Generate A and B, two random $N \times N$ matrices over $GF(p)$ such that A is invertible, and note $M = [A|B]$.
3) For each $i \in \{0, 1, \ldots, n\}$, compute a matrix $M_i^{''} = [A_i|B_i]$ by multiplying M to the left by a random invertible matrix P_i.
4) Generate the random scrambling matrix Δ as a $N \times N$ diagonal invertible matrix over $GF(p)$.
5) For each $i \in \{1, \ldots, n\}$ generate a soft noise matrix D_i, a $N \times N$ random matrix over $\{-1, 1\}$, and compute the softly disturbed matrix $M_i^{'} = [A_i|B_i + D_i\Delta]$.
6) Generate D_0, the hard noise matrix, by:
   - generating a soft noise matrix;
   - replacing each diagonal term by q.
7) Compute the hardly disturbed matrix $M_0^{'} = [A_0|B_0 + D_0\Delta]$.
8) Choose a random permutation of columns $\mathcal{P}(\cdot)$ and compute $M_i = \mathcal{P}(M_i^{'})$ for $i \in \{0, 1, \ldots, n\}$.
9) The $n+1$ matrices $\{M_0, \ldots, M_n\}$ compose the public key and the private key is the permutation $\mathcal{P}(\cdot)$, the hidden matrix M, and the scrambling matrix Δ.

As Aguilar and Gaborit propose in [4], instead of multiplying each soft and hard noise matrix by the noise scrambling matrix in this protocol, it would be computationally more efficient to generate each column i of the noise matrices as a random set over $\{-\delta_i, \delta_i\}$, noting $\delta_1, \ldots, \delta_N$, the diagonal terms of the noise scrambling matrix. To highlight the difference between soft noise and hard noise matrices and ease the protocol comprehension, we have decided to separate this in two steps, even if in a real implementation only one step would have to be done. Similarly, the random permutation $\mathcal{P}(\cdot)$ would not be applied to each matrix $M_i^{'}$, but directly to M at the end of step two, and all the disturbing process would be done taking into account this initial permutation.

C. Encryption

To encrypt, one considers a message vector m in Z_r^N. The encryption is done in the following way: first one multiplies the message m by the HDM M_0. Then, one disturbs this result by adding soft noise vectors. One selects n random vectors r_i, with coordinates smaller than ϵ_{max}. One then adds the soft noise vectors $r_i M_i$ to mM_0 to get the encrypted message.

Encryption

1) Input: a message $m \in Z_r^N$.
2) For each $i \in \{1, \ldots, n\}$ construct randomly the disturbing vectors r_i in $Z_{\epsilon_{max}}^N$.
3) Return $c = mM_0 + \sum_{i=1}^{n} r_i M_i$.

Of course, if the message is not a vector, two options are possible. One can maximize the transmission factor by splitting the message m in a vector of N $log(r)$-bit integers $(m_1, \ldots, m_N)$ or preserve the homomorphic property by using just one coordinate and ensuring that $r > m$.

The encryption can be seen as a linear action on the message disturbed by the addition of the noise vectors induced by the r_i's and the M_i's. The encrypted message is a vector c of dimension $2N$ over $GF(p)$.

D. Decryption

To recover the message m, the user will operate in two phases. First, he will recover the noise included in the vector (steps 1 and 2 of the Encryption protocol), and then he will unscramble and filter out this noise to obtain the message (steps 3 to 5).

Decryption

1) Input: a ciphertext $c \in GF(p)^{2N}$.
2) Compute the non-permuted noisy vector $c' = \mathcal{P}^{-1}(c)$.
3) Retrieve $e = c'_D - c'_U A^{-1} B$, the scrambled noise, c'_U and c'_D being resp. the undisturbed and disturbed halves of c'.
4) Compute the unscrambled noise $e' = e\Delta^{-1}$.
5) For each e'_j in $e' = [e'_1 \cdots e'_N]$, compute $e''_j = e'_j - \mu$ with $\mu := e'_j \mod q$ if $(e'_j \mod q) < q/2$ and $\mu := (e'_j \mod q) - q$, else.
6) For each $j \in \{1 \cdots N\}$, compute $m_j := e''_j q^{-1}$.
7) Return $m = (m_1, \ldots, m_N)$.

In the first step, the random column permutation is undone. Then, the N first coordinates of the vector and the initial matrix M are used to obtain what the N last coordinates (which have been

disturbed) should be without noise. These values are subtracted to the noisy ones and the scrambled noise is obtained (step 2).

This noise is composed of soft and hard noise, but it cannot be directly filtered because it was scaled up by the noise scrambling matrix. In step 3 the noise is therefore unscrambled. Finally, in step 4 the soft noise is filtered out, and in step 5 each coordinate is divided by the hard noise factor to obtain the message coordinates m_i.

By lack of space we do not give the precise proof of the correctness of the scheme but it relies on the fact that we choose parameters so that the action of the noise is controlled.

E. Homomorphic property

We begin this subsection with a formal definition of the concept of an (ℓ, r, N)-homomorphic scheme introduced in the introduction.

Definition 1: A (ℓ, r, N)-homomorphic public-key encryption scheme. A probabilistic public-key encryption scheme will be (ℓ, r, N)-homomorphic if its probabilistic encryption algorithm, Enc, and its decryption algorithm, Dec, satisfy the following conditions:

- The inputs of Enc are the elements of $(\mathcal{M}^N, +)$, where $\mathcal{M}$ is an additive group with r elements. The algorithm Enc outputs an element of $\mathcal{C}$, the ciphertext set.
- There is a public operation on $\mathcal{C}$, denoted $\oplus$, such that for all $k \leq \ell$ and for all k-tuple $(m_1, m_2, \ldots, m_k)$ of elements of $\mathcal{M}^N$,

$$\mathrm{Dec}(\mathrm{Enc}(m_1) \oplus \mathrm{Enc}(m_2) \oplus \cdots \oplus \mathrm{Enc}(m_k)) = m_1 + m_2 + \cdots + m_k.$$

Note that with this definition, $\mathrm{Enc}(m_1) \oplus \cdots \oplus \mathrm{Enc}(m_k)$ can be distinct from all the outputs of $\mathrm{Enc}(m_1 + \cdots + m_k)$, *i.e.* no randomness use in Enc on the input $m_1 + \cdots + m_k$ can lead to this value.

Theorem 2: The previous protocol is (ℓ, r, N)-homomorphic.

Proof. Our protocol is intrinsically additive. Indeed, a ciphertext generated with the public key is of the form $c = vMP + m[0|qI_n]\Delta P + s\Delta P \mod p$, for a given vector $v \in Z_p^N$, a plaintext $m \in Z_r^N$, and a soft noise vector s in $Z_{l_0}^N$,[1] M, P, q, p and l_0 being scheme parameters. Thus, when two such ciphertexts are added we obtain

$$c + c' = (v+v')MP + (m+m')[0|qI_n]\Delta P + (s+s')\Delta P \mod p.$$

Parameter p has been chosen such that $rq = p + \epsilon$ with $\epsilon \in Z_{l_0}^N$. Thus, if $m + m' > r$ we have $(m+m')q = (m+m'-r)q + p + \epsilon$ and therefore, $c + c'$ equals

$$(v+v')MP + (m+m' \mod r)[0|qI_n]\Delta P + s''\Delta P \mod p$$

with $s'' \in Z_{3l_0}^N$. If ℓ additions of ciphertexts are done, s'' will be in $Z_{(2\ell+1)\times l_0}^N$. As $q = 2 \times l_0 \times (2\ell + 1)$, the soft noise can be filtered out and the ciphertext decrypted. □

III. Security

The security of our scheme can be separated in three parts. A first part is the structural security which permits to break completely the system by finding the private key, a second part is message security (or one-wayness) and the third part is semantic security.

[1]If the ciphertext results from the addition of two ciphertexts, the coordinates of s may be larger.

A. Structural security

The structural security of our scheme can be related to the Hidden Lattice Problem, this problem is introduced in [4], the security of this problem is also studied in [4]. By lack of space we do not recall this problem extensively here and refer to the previous references, where it is shown that this problem can be related to NP-complete coding problems like [6] and that the best known attack against this problem is exponential in N, typically $\binom{2N}{N}$. Moreover it is also shown that lattice based attacks are very unlikely to be usable to solve this problem.

B. One-wayness against Chosen Plaintext Attack

Besides the structural security of our scheme which is studied in other papers, we now focus on specific security aspects of our scheme. We begin with a definition of a general problem, CKVP, to which the one-wayness of our scheme is related.

Definition 3 (Computational Knapsack Vector Problem): Let p_v be a large prime number. Let e and r be two integers with $e < r < p_v$. Consider a set of r_v different matrices, $M_0, M_1, \ldots, M_{r_v}$, of size $k_v \times n_v$. Let c be an element of the subset

$$\left\{ mM_0 + \sum_{i=1}^{r_v} r_i M_i, m \in Z_r^{k_v}, r_i \in Z_e^{k_v}, i \in \{1, \ldots, r_v\} \right\},$$

of $GF(p_v)^{n_v}$. Determine m in this expression.

Clearly, this problem is a generalization of the well known Knapsack problem: determine the expression of a given integer as a linear combination with small coefficients of a given basis. Breaking the one-wayness of our scheme can be reduced to an instance of CKVP with $p_v = p$, $e = \epsilon_{max}$, $r_v = n$, $k_v = N$, and $n_v = 2N$. Therefore, the assumption associated to the one-wayness of our scheme is that there exists no family of circuits with polynomially bounded size in N and $\log p$ able to solve CKVP with non-negligible advantage for the subset of instances associated to our scheme. We define it as the Computational Knapsack Vector Problem Assumption (CKVPA).

In opposition to the case of structural attacks, LLL is very natural in the case of breaking one-wayness due to the similarity of CKVP with the traditional Knapsack problem. Indeed, the encryption of a message $m = (m_1, \cdots, m_N)$ with a set of random vectors $r_i = (r_{i1}, \cdots, r_{iN})$ for $i \in \{1, \cdots, n\}$ can be seen as the linear action of the matrix M resulting from the row concatenation of the matrices $M_0, M_1, \ldots, M_n$ over the large vector $x = (m_1, \cdots, m_N, r_{11}, \cdots, r_{1N}, \cdots, r_{n1}, \cdots, r_{nN})$. We obtain:

$$c = xM,$$

and deduce straightforwardly that the vector x is included in one of the vectors of the lattice of dimension $(n+1)N$ and determinant p^{2N} generated by :

$$L_{HLP} = \begin{bmatrix} 1 & 0 & \cdots & 0 & & M_0 \\ 0 & \ddots & \ddots & \vdots & & \vdots \\ \vdots & \ddots & \ddots & 0 & & M_n \\ 0 & \cdots & 0 & 1 & & c \\ 0 & \cdots & 0 & 0 & & \\ p & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & \ddots & \ddots & & & \vdots \\ \vdots & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & \cdots & 0 & p \end{bmatrix}$$

Hence if the norm of x is small it is possible to recover it (and therefore the message m) by a lattice based attack.

Estimation of the attack cost: One major difficulty of LLL based attacks is to estimate the cost of recovering a shortest vector of a lattice. Indeed, for some parameters it is easy to prove that the computational cost is polynomial, but the probability of finding a shortest vector is very low. When the parameters are adjusted to have a reasonable chance of finding a shortest vector, evaluating the computational cost is hard, and heuristics seem to show that this complexity is far worse than polynomial. The NTRU cryptosystem has developed researches on this particular area. One of the difficulties is that the cost of the attack depends on "how" short is a short vector. Recall that the norm of a vector $x = (x_1, \ldots, x_n)$ is defined by:

$$\|x\| = \sqrt{\sum_1^n x_i^2}$$

the Gaussian heuristic [7] stands that the minimum norm $\mu(L)$ (the norm of the shortest vector) of a random lattice of dimension $\dim(L)$ and determinant $\det(L)$ satisfies:

$$\mu(L) \approx \sqrt{\frac{\dim(L)}{2\pi e}} \det(L)^{1/\dim(L)}$$

This result gives an *a priori* estimation of the expected minimum norm of a lattice. Now, we consider the problem of finding by LLL a shortest vector. Even if there is no theoretical result on this, heuristics and researchers of this research field (see [8], [9]) seem to confirm that finding by LLL a shortest vector of a lattice very close to the Gaussian heuristic has a complexity at least in $2^{d/6}$ (see [10] and [7] for a more precise heuristic) for d the dimension of the lattice. On the other hand, if the norm of the shortest vector is smaller than the Gaussian heuristic by a strong factor, the computational cost can be drastically smaller.

For instance, experiments on the NTRU lattices, where one searches for a shortest vector very close to the Gaussian heuristic, show that it possible to find a shortest vector up to dimension $200-250$ (which corresponds to the heuristic of $\approx 2^{45}$ operations), but that it is difficult beyond. In particular, NTRU recommends to use lattices of dimension at least 500 for their system. These orders of magnitude have also been observed on knapsack cryptosystems [10]. In fact, the only lattice based system for which it was possible to find shortest vectors for dimensions greater than 300 was the GGH system for which Nguyen could find shortest vectors up to dimension 350 [11], but it corresponds to the case where the shortest vector had a strong dividing factor compared to the Gaussian heuristic.

The HLP lattice has a structure similar to the knapsack lattice and the tests we realized have shown that computational complexity for dimensions up to 200 with HLP lattices is similar to the one of NTRU and knapsack lattices.

Even if the research on lattices and cryptology is recent (about 20 years) this area has been relatively stable and for instance the challenges proposed by NTRU (www.ntru.com) have not been broken. It is interesting to remark that a similar situation holds for error-correcting codes, where finding the minimum weight of a code is (as far as we know) polynomial, but depends exponentially on the actual weight of the searched codeword. For a random code there is an equivalent notion of the Gaussian heuristic: the Gilbert-Varshamov bound, and searching for a codeword which has a ratio to the expected Gilbert-Varshamov bound divides the time for finding it by the same factor. Although the similarity may not be exactly the same, it is not surprising that searching for a shortest vector far below the Gaussian heuristic takes far less computation.

In term of security for our system, we have to choose our parameters such that the vector x has norm at least of the same order than the Gaussian heuristic and a public key such that $(n+1) \times N$ is at least 500. For $N = 50$ this leads to a public key with one HDM and nine SDMs.

C. Semantic security against Chosen Plaintext Attack

In this subsection, we analyze the indistinguishability of our scheme. Due the homomorphic property of the cryptosystem, the problem of determining if a ciphertext c is an encryption of m or an encryption of m' can be reduced to the problem of distinguishing an encryption of $(0, \ldots, 0)$ from a random element of $GF(p)^{2N}$ (by subtracting from c an encryption of m). Therefore, we define a general problem, DKVP, to which the semantic security of our scheme is related.

Definition 4 (Decision Knapsack Vector Problem): Let p_v be a large prime number. Let e an integer with $e < p_v$. Consider a set of r_v different matrices, $M_1, \ldots, M_{r_v}$, of size $k_v \times n_v$. Let c be an element of $GF(p_v)^{n_v}$, determine if c can be expressed as a linear combination, $c = \sum_{i=1}^{r_v} r_i M_i$, where $r_i \in Z_e^{k_v}$ for all $i \in \{1, \ldots, r_v\}$.

Breaking the semantic security of our scheme can be reduced to an instance of DKVP with $p_v = p$, $e = \epsilon_{max}$, $r_v = n$, $k_v = N$, and $n_v = 2N$. Therefore, the assumption associated to the semantic security of our scheme is that there exists no family of circuits with polynomially bounded size in N and $\log p$ able to solve DKVP with non-negligible advantage for the subset of instances associated to our scheme. We define it as the Decision Knapsack Vector Problem Assumption (DKVPA). By lack of space we omit the details of the proof.

D. Parameters

1) Parameters and security: Our system is based on 4 parameters: l, r, N and p. Our system is very versatile, the only constraints are on its security and parameters have to be chosen to satisfy a good security. There are two main type of attacks to be protected from. The structural security and the chosen plaintext attack. To resist the structural attack one must choose $N \geq 50$, this parameter assures that searching for the non disturbed columns is of order $\binom{2N}{N} \geq \binom{100}{50} \approx 2^{100}$. In order to resist an attack on a characterization of a set of N columns by finding non invertible square $N \times N$ submatrix, one has to take $p \geq 2^{60}$ since in that case finding one such submatrix has a cost of 2^{80} operations: $1/p \approx 2^{60}$ (the probability of finding a non invertible random matrix on $GF(p)$) times 2^{20} the cost of computing one determinant.

The second type of attack against which the system has to be secured is the lattice based attack. The lattice L_{HLP} (say L in the following) has determinant p^{2N} and its expected minimal norm $\mu(L)$ is hence by the Gaussian heuristic:

$$\mu(L) \approx \sqrt{\frac{\dim(L)}{2\pi e}} \det(L)^{1/\dim(L)} = \sqrt{\frac{N(n+3)}{2\pi e}} p^{\frac{2}{n+3}}.$$

We saw that the search for short vectors very close to this expected bound had a heuristic complexity in $2^{dim(L)/6}$ hence first we must choose our parameters so that $dim(L) = (n+3)N \geq 500$ and second the norm of the target vector has to be very close to the expected value of $\mu(L)$. The target vector is $x =$

$(m_1, \cdots, m_N, r_{11}, \cdots, r_{nN}, 0, \cdots, 0)$ of length $(n+3)N$. One can suppose that the m_i can be zero, hence since $\frac{\epsilon_{max}}{2} \leq m_i \leq \epsilon_{max}$, one deduces: $\|x\| \geq \sqrt{n(\frac{\epsilon_{max}}{2})^2} = \sqrt{n}\frac{\epsilon_{max}}{2}$.

Hence we should choose ϵ_{max} such that:

$$\sqrt{\frac{N(n+3)}{\pi e}} p^{\frac{2}{n+3}} \approx \sqrt{nN}\frac{\epsilon_{max}}{2}$$

Eventually we obtain $\frac{\epsilon_{max}}{2} \geq \sqrt{\frac{2(n+3)}{n\pi e}} \approx 0.5\sqrt{\frac{n+3}{n}} p^{2/(n+3)}$.

Notice that at the difference from NTRU when having a norm too high may induce decryption failures, we do not have this problem here, since we are still able to decrypt by the conditions on .

2) Examples of parameters: We focus on three types of applications. Application in which one wants to preserve a vector space structure (over $GF(2)$ for instance), a second application in which one wants to add on a given coordinate 0 or 1 (a voting system) and eventually an application in which one wants to optimize the transmission rate and when one do not want to use homomorphic properties. In the following we choose parameters which satisfy the conditions of the system. To have a good lattice security we choose $n = 9$ for the three following examples, which gives a lattice of dimension 600 for the lattice based attack.

• Example 1: Preservation of a binary vector space structure, a $(2, 2^{38}, 50)$-homomorphic scheme.

Considering $GF(2)$ gives $r = 2$. Suppose one wants to be able to do 2^{38} addition of encrypted messages over $GF(2)^N$. One can take $p \approx 2^{60}$, from which one deduces $\epsilon_{max} = 1024$. One then obtains: $l_0 = 9.50.1024 + 50.2 \approx 2^{19}$, $q = 2^{21}.2^{38}(forl = 2^{38})$ and eventually $p \approx 2^{60}$.

• Example 2: Voting scheme, a $(2^{30}, 2^{30}, 50)$-homomorphic scheme.

Suppose one adds only 0 or 1, for 2^{30} voters. Each coordinate corresponds to a special candidate for the vote (here $N = 50$). We can take $\epsilon_{max} = 2^{14}$, $l_0 = 9.50.2^{14} + 50.1$ (in this special case one adds only 0 or 1 rather than r), $q = 2^{21}.2^{30} = 2^{51}$ and eventually $p \approx 2^{81}$.

• Example 3: Optimization of the transmission rate, a $(2^{30}, 1, 50)$-homomorphic scheme.

Suppose one wants to optimize the transmission rate then one do not use the homomorphic properties of the scheme ($l = 1$), and proceed as usually by encrypting blocks of messages. In this case we want the ratio $\frac{r}{p}$ to be the highest possible. Suppose we take a large ring with $r = 2^{30}$ one can then take $p = 2^{80}$ and $\epsilon_{max} = 2^{14}$. In that case the transmission rate is $\frac{1}{2}\frac{Log_2(r)}{Log_2(p)} = \frac{3}{16}$.

3) Performances and security: Our system is very fast, in particular adding two encrypted messages is the cost of an addition in $2N * Log_2(p)$ bits which gives respectively 6000,8000 and 8000 bits addition.

The size of the public is $2N^2(n+1)Log_2(p)$ which gives respectively $3Mb, 4Mb$ and $4Mb$.

The encryption speed is the cost of $(n+1)N$ additions of vectors of lengths $2N$ and $Log_2(p)$ bits which gives respectively plus the cost of multiplying by the random vectors with $Log_2(\epsilon_{max})$ bits, hence respectively $2^{25}, 2^{26}$ and 2^{26} bits operations. (Notice that these times may be divided by an order 10 if one chooses the random values r closed to a power of 2).

The decryption speed is the cost of a multiplication of two $N \times N$ matrices with elements in $GF(p)$: $2N^2.Log(2,p)^2$ respectively $2^{24}, 2^{25}$ and 2^{26} operations.

The security of three examples, is respectively at least $2^{80}, 2^{100}$ and 2^{100}.

IV. Conclusion

In this paper we presented a new lattice based homomorphic scheme faster than previously known schemes based on number theory, which moreover has the specificity to preserve vectorial structure. A natural question for which a complete treatment is beyond the scope of this short article is whether it is possible to find this homomorphic property in other lattice based schemes. The fastest lattice based cryptosystem is the NTRU cryptosystem, for this system it appears possible to have the additive homomorphic property but the number of possible homomorphic operations should be small compared to the system we introduced, since there is the risk of constructing spurious keys and since one wants to resist LLL based attack for both a message and the sum of several encrypted messages. Another scheme is the Regev scheme and its generalization [12], in that case it is also possible to obtain the same type of homomorphic properties but the large expansion factor makes it difficult to use in practice.

References

[1] Damgård, I., Jurik, M.J.: A Generalisation, a Simplification and some Applications of Paillier's Probabilistic Public-Key System. In: PKC' 01. Volume 1992 of LNSC series. (2001)

[2] Lipmaa, H.: An Oblivious Transfer Protocol with Log-Squared Communication. In: The 8th Information Security Conference (ISC'05). Volume 3650 of Lecture Notes in Computer Science., Springer-Verlag (2005) 314–328

[3] Nguyen, L., Safavi-Naini, R., Kurosawa, K.: Verifiable shuffles: a formal model and a Paillier-based three-round construction with provable security. Int. J. Inf. Secur. **5**(4) (2006) 241–255

[4] Aguilar Melchor, C., Gaborit, P.: A Lattice-Based Computationally-Efficient Private Information Retrieval Protocol. In: Western European Workshop on Research in Cryptology (WEWoRC'2007), Bochum, Germany. Book of Abstracts. (2007) 50–54, Extended version available on IACR eprints `http://eprint.iacr.org/2007/446`

[5] Nguyen, P.Q., Pointcheval, D.: Analysis and improvements of ntru encryption paddings. In: CRYPTO '02, Springer-Verlag (2002) 210–225

[6] Wieschebrink, C.: Two NP-complete Problems in Coding Theory with an Application in Code Based Cryptography. In: 2006 IEEE International Symposium on Information Theory. (2006) 1733–1737

[7] N. Howgrave-Graham, J.H.S., Whyte, W.: Estimated breaking times for ntru lattices. Technical report (2003)

[8] Phong Q. Nguyen and Jacques Stern: The two faces of lattices in cryptology. In: Cryptography and Lattices, International Conference, CaLC 2001, Providence, RI, USA, March 29-30, 2001, Revised Papers. Volume 2146 of Lecture Notes in Computer Science., Springer (2001) 146–180

[9] N. Howgrave-Graham, J. Hoffstein, J.P., Whyte, W.: On estimating the lattice security of ntru. Technical report (2005)

[10] Phong Q. Nguyen and Jacques Stern: Adapting density attack to low-weight knapsacks. In: AsiaCrypt. Lecture Notes in Computer Science, Springer (2005) 41–58

[11] Phong Q. Nguyen: Cryptanalysis of the Goldreich-Goldwasser-Halevi Cryptosystem from Crypto '97. In: CRYPTO '99. Volume 1666 of LNCS., Springer (1999) 288–304

[12] Regev, O.: New lattice based cryptographic constructions. In: Proceedings of the 35th Annual ACM Symposium on Theory of Computing, STOC'2003 (San Diego, California, USA, June 9-11, 2003), New York, ACM Press (2003) 407–416