Bounded Fully Homomorphic Encryption from Monoid Algebras

Mugurel Barcau Vicenţiu Paşol

Abstract

We present a new method that produces bounded FHE schemes (see Definition 3), starting with encryption schemes that support one algebraic operation. We use this technique to construct examples of encryption schemes that, theoretically can handle any algebraic function on encrypted data.[1]

1 Introduction

Homomorphic encryption was introduced in [28] as a machinery for arbitrary computations over encrypted data. Since then, several attempts to produce such encryption schemes have been made. Lately, a lot of attention has been drawn to this domain especially for the following two reasons: first, globalization of information impose the need to work with huge amount of data that is input for different users, but no information has to leak to a potential competitor; second, a big breakthrough in this direction was obtained by C. Gentry in [14].

1.1 Short History

In the early 90's, the authors of [13] and [3] independently proposed an algorithm, called Polly Cracker, which was capable of performing algebraic computations on the encrypted data without revealing the encrypted information. However, few years later, the algorithm proved to be insecure and no modifications of the algorithm could solve this inconvenient (cf. [11]). In the late 90's, a secure and efficient algorithm to encrypt messages (NTRU) was proposed in [25]. The algorithm has the same ring homomorphic feature as the previous one, but only few operations can be performed on the encrypted data (with the parameters suggested to be implemented, only few additions and no multiplications are allowed). A major breakthrough was obtained by C. Gentry in his Ph.D. thesis [14], where he used the error-based encryption technique for the purpose of achieving fully homomorphic encryption (FHE) schemes. Gentry applied the cryptosystem proposed in [19] to the case of ideal lattices, thus producing a scheme which accommodates a much larger (but bounded) number of computations on the encrypted data (see also [15]). This *leveled* feature comes from the fact that the algorithm is an error-based one, so that only circuits which keep the noise very low can be applied to the encrypted data. He also proved that if an encryption scheme can

[1]The content of this article is protected under the law by the U.S. Patent Appln. No. 14/936,097 and European Patent Appln. No. EP 15193706.7

handle its own decryption algorithm (augmented by a single gate), in other words the scheme is *bootstrappable*, then one can use it in a limiting process to produce a FHE scheme. The word limiting is important from the practical implementation point of view because, in this sense, one can achieve only leveled FHE, i.e. one has to prescribe from the beginning the maximum degree (or the depth) of the polynomials to be computed on the encrypted data. Since then an entire collection of fully homomorphic encryption schemes were constructed [38], [35], [10], the methodology in Gentry's seminal work being followed.

A second generation of schemes started with the work of Brakerski and Vaikuntanathan [6], who established FHE in a simpler way, based on the learning with errors (LWE) assumption. To accommodate multiplication and hence obtain (leveled) fully homomorphic encryption schemes, the authors had to introduce the so-called re-linearization technique. Also, the construction was refined using a modulus switching technique to obtain better efficiency (see also [8]). Currently, perhaps the simplest (leveled) FHE scheme based on the learning with errors assumption is by Brakerski [7] who builded on Regev's public key encryption scheme [29]. The most recent achievement in this direction was obtained in [18], where the authors were able to construct a simpler (leveled) FHE scheme based on the LWE assumption by removing the extensive and complicated step that involves the re-linearization procedure. It is worth mentioning here that the error-based technique was borrowed by the authors of [1] to construct a more secure variant of Polly Cracker. Their cryptosystem can be seen both as a high-dimensional generalization of Regev's LWE-based scheme and as a noisy generalization of the Polly Cracker-style cryptosystems (cf. [24]). Also the re-linearization technique was used to produce a more secure NTRU cryptosystem [37] (for a survey on recent developments on NTRU cryptosystems see [36]).

The current state of the art in terms of FHE implementation is represented by a recent software library (HElib), developed by Halevi and Shoup, which is available for public trial at https://github.com/shaih/HElib. HElib is an implementation of the RLWE encryption scheme described in [8], along with many other optimizations [22]. To achieve FHE, the authors implemented a new recryption procedure with running times around 6 minutes [23]. The fact that bootstrapping takes such a large amount of time makes this implementation of FHE unattractive. Future work in this direction will focus on minimizing the running time of the FHE bootstrapping procedure [12].

1.2 Short discussion about practical encryption

Unfortunately, all error-based encryption schemes proposed until now are far from being practical. But what exactly means *practical*? The security/efficiency of an algorithm cannot just be considered as a function of the security parameter and prove polynomial asymptotics about it. It is not feasible from the practical point of view. A practical implementation has only a finite and well established range for the security parameter. A polynomial in the security parameter can take much larger values than an exponential in the practical range if, for example, the coefficients of that polynomial are large enough. It is also true that the practical range of the security/efficiency parameter (for a given encryption scheme) varies in time, i.e. an algorithm which proved unfeasible from the practical point of view 10 years ago, now can be implemented successfully. This is why we promote and advocate a strategy to

redefine the concepts of security and efficiency in a way to become practical for implementing algorithms. The notions should depend on the computing power and the power to stock and transmit information at a given time so one knows for example what "practical" means for an algorithm to be implemented and be able to predict the security/efficiency in the future if for example one assumes Moore's law.

1.3 Our results

In 2013 at the UCI conference "Workshop on Lattices with Symmetry", C. Gentry asked if one can actually realize at least a leveled FHE but with an algorithm that is not error-based. In our attempt to answer Gentry's question we rediscovered an idea that goes back to Grigoriev and Ponomarenko in [21], which from the mathematical point of view is very natural. If one has an encryption scheme which supports one operation (such encryption schemes are abundant in the literature), then one can use the group algebra theory to obtain an encryption scheme that can handle any algebraic function on the encrypted data. However, the algorithms presented in [21] do not produce FHE, i.e. ring homomorphic encryptions over the field $\mathbf{F}_2$.
Our scheme is more general and flexible enough to overcome the weaknesses of Grigoriev and Ponomarenko' construction, proposing a blueprint to produce bounded FHE schemes, which are less restrictive than (compact) FHE schemes (see Definition 3 bellow). We also give in this paper several examples and analyze their properties.

We point out that the compactness property was introduced in order to avoid trivial encryptions. But one has to notice that boundedness also eliminates the possibility of encrypting "trivially". Moreover, we shall prove that from a practical point of view, the theoretical notions of compactness and boundedness are not different, and to be fair, none of them guaranties efficiency.

1.4 Layout of the paper

In Section 2 we review background information from the theory of homomorphic encryption schemes, introduce the notion of bounded schemes and prove that from the practical point of view it is equivalent with the notion of compact schemes. In section 3, we give background information about monoid algebras. In Section 4 we describe our main construction, which gives the general recipe for producing ring HE schemes starting with encryption schemes that have homomorphic properties with respect to one operation. We give examples of this construction in Sections 5 and 6. The examples we give are, in some sense, complementary to each other and reflect important instances of our blueprint. We conclude the paper in the last section.

2 Homomorphic Encryption Schemes

The homomorphic encryption schemes in their generality were treated by different authors and many treaties. We refer to [33] for a monograph treatment of the subject and to [2] for a treatment of their security behavior. General encryption schemes are composed of three

algorithms: KeyGen, Enc, Dec and two sets: the plaintext space $\mathcal{P}$ and the ciphertext space $\mathcal{C}$. Basically, one generates, given a security parameter λ, a secret and a public key (sk, pk) by KeyGen, then the next two algorithms describe how to associate to a plaintext $m \in \mathcal{P}$ a ciphertext $c \in \mathcal{C}$ using the public key pk and viceversa, using the secret key sk, how to associate to a ciphertext $c \in \mathcal{C}$ a plaintext $m = \mathrm{Dec}(c)$, such that $\mathrm{Dec}(\mathrm{Enc}(m)) = m$. Since in our work the key generation is made explicitly, we assume in what follows the existence of this algorithm in all of the encryption schemes without mentioning it in the notation. We will write for short such an encryption scheme by the quadruple ($\mathcal{C}$, $\mathcal{P}$, Enc, Dec).

Definition 1. Let AS be an algebraic structure, such as semigroup, monoid, group, ring, etc. An AS homomorphic encryption scheme $(G, H, \mathrm{Enc}, \mathrm{Dec})$ is an encryption scheme such that both plaintext space H and ciphertext space G are endowed with the algebraic structure AS and such that $\mathrm{Dec} : G \to H$ is an AS-homomorphism.

We note that different authors give different names for such an encryption scheme, all of them bearing the name homomorphic. Some authors give the name group homomorphic encryption schemes to the above definition while they don't require the plaintext (respectively ciphertext) space to be an actual group but only to have operations compatible with the decryption algorithm (which would actually correspond to semi-group homomorphic encryption schemes in our definition).

It is also worth to mention that many of the encryption schemes already treated in the literature are in fact group homomorphic schemes (RSA, ElGamal, Paillier, Goldwasser-Micali, Benaloh, etc.), but one can produce other schemes where the spaces have only a monoid structure or even a semigroup structure.

Practical encryption schemes require additional constraints on the algorithms KeyGen, Enc and Dec such that the encryption and decryption processes are both feasible, secure and efficient. For an AS-homomorphic encryption scheme the algorithms that compute the algebraic structure on both plaintext and ciphertext spaces need also be efficient. This means that there exists a fourth algorithm, called the *Evaluation* algorithm and denoted by $\mathrm{Eval}(f, \bar{c})$, which takes as input any circuit f that can be represented only by AS-operations and a corresponding number of ciphertexts and outputs a ciphertext. This algorithm is required to be compatible with the decryption algorithm, i.e. $\mathrm{Dec}(\mathrm{Eval}(f, \bar{c})) = f(\mathrm{Dec}(\bar{c}))$.

Definition 2. An AS homomorphic encryption scheme $(G, H, \mathrm{Enc}, \mathrm{Dec})$ is called a *leveled* AS homomorphic encryption scheme if the decryption algorithm is correct only for a certain number of AS operations made on G.

We remark that so far all of the proposed secure ring homomorphic encryption schemes are "error"-based schemes which make them leveled for practical purposes. Even if the scheme is bootstrapable (see [15]), the practical implementation of such a scheme is leveled, because the construction of a fully homomorphic encryption scheme out of a bootstrapable one is a limiting process. However, the authors of [17] where able to obtain a "pure" fully homomorphic encryption scheme by constructing a circular ladder of encryption keys, under an additional assumption called circular security. The idea was implemented in HElib (see [23] for more details).

The above definitions imply a parametrized set, $HE(\lambda)$, of encryption schemes.

Definition 3 (see also [6], Definition 3.4.)**.** A homomorphic encryption scheme, $HE(\lambda)$, is called *compact* if there exists a function $s = s(\lambda)$ such that:

1. The output length of $\text{Eval}_{HE(\lambda)}(C, c)$ is at most $s(\lambda)$ bits long (regardless of the circuit C or the number of inputs c).

2. As a function of λ, s is asymptotically polynomial.

A scheme where only condition 1. holds will be called *bounded.*

We would like to point out that all the ring homomorphic encryption schemes constructed in this paper are bounded.

Remark 4. *The initial name we used for bounded homomorphic encryption schemes was quasi-compact. But this name was already used in Gentry's thesis (see also [9] for the definition in quantum homomorphic encryption), to denote a class of homomorphic encryption schemes for which $s = s(\lambda, depth(C))$ is polynomial both in the security parameter and in the depth of the circuit. As one can easily see, the two notions are complementary in the sense that a bounded and quasi-compact scheme is compact. The quasi-compact schemes make sense from the practical point of view, but from a theoretical point of view they can never be "embedded" into a compact scheme if they are not already. The meaning of this sentence will become apparent in the next theorem.*

The following theorem shows that any bounded encryption scheme can always be fit into a compact one, as long as there exists such a compact scheme.

Theorem 5. *Let $S(\lambda)$ be a bounded scheme and assume that there exists a compact scheme $S'(\lambda)$. Let $I = [\lambda_0, \lambda_1]$ be the practical (any finite for that matter) range of the security parameter. Then, there exists a compact scheme S'' such that $S''(\lambda) = S(\lambda)$ for all $\lambda \in I$.*

Proof. The proof is constructible. We define $S''(\lambda)$ to be $S(\lambda)$ if $\lambda \in I$ and to be $S'(\lambda)$ else. Since the asymptotics of the scheme S'' is the same as the asymptotics of the scheme S' and for a finite range of λ, any function can be bounded polynomially, we get our conclusion. □

We end the section by recalling that the notion of fully homomorphic encryption scheme (see for example [15]) is equivalent to the ring homomorphic encryption scheme, where the plaintext space is the field with two elements $\mathbf{F}_2$. Indeed, this is due to the fact that any boolean circuit with XOR and AND gates can be written as a polynomial over $\mathbf{F}_2$ with the XOR and AND gates replaced by addition and multiplication (for more details see [16]).

3 Monoid Algebras

In this section we recall how one can associate to any monoid and any commutative ring with unity a certain algebra and we will explore its properties. Let $(M, \cdot)$ be a monoid and let R be a commutative ring with unity. As an R-module the *monoid algebra* $R[M]$ is free with a basis consisting of the symbols $[x], x \in M$, and the multiplication defined by the

R-bilinear extension of $[x] \cdot [y] = [xy]$. Therefore every element of $a \in R[M]$ has a unique representation

$$a = \sum_{x \in M} a_x [x] \tag{1}$$

in which $a_x = 0$ for all but finitely many $x \in M$, and the product of $a, b \in R[M]$ is given by

$$ab = \sum_{x \in M} \left(\sum_{yz=x} a_y b_z \right) [x]. \tag{2}$$

Notice that the identity element of $R[M]$ with respect to multiplication is $1[e]$ where e is the identity element of M. If M is a group then the monoid algebra above is called a *group algebra*. Notice that the R-algebra $R[M]$ is commutative if and only if M is commutative.

Example 6. *If M is the free monoid in one generator then $R[M] \cong R[X]$ as R-algebras, whereas if G is the free group in one generator then $R[G] \cong R[X, \frac{1}{X}]$ as R-algebras.*

An *R-character* of a monoid M is a monoid homomorphism $\chi : M \to A$ from M to the multiplicative monoid of an R-algebra A, i.e. $\chi(xy) = \chi(x)\chi(y)$, for all $x, y \in M$, and $\chi(e) = 1$. The monoid algebra $R[M]$ is characterized up to isomorphism by the following universality property: for every R-character $\chi : M \to A$, there exists a unique R-algebra homomorphism $R[M] \to A$ extending χ. If we also denote by χ the extension $R[M] \to A$ then:

$$\chi \left(\sum_{x \in M} a_x [x] \right) = \sum_{x \in M} a_x \chi(x). \tag{3}$$

Let M, N be monoids, and $\phi : M \to N$ be a monoid homomorphism. Then ϕ induces an R-algebra homomorphism $\phi_R : R[M] \to R[N]$ via

$$\phi_R(\sum_{x \in M} a_x [x]) = \sum_{x \in M} a_x [\phi(x)]. \tag{4}$$

Notice that formula (4) defines ϕ_R as the R-linear extension of ϕ.

For any R-algebra A, there is a natural R-algebra homomorphism $\epsilon : R[A] \to A$ given by

$$\epsilon \left(\sum_{x \in R} r_x [x] \right) = \sum_{x \in R} r_x x. \tag{5}$$

4 Blueprint

Let R be a ring, which we will assume throughout the paper to be finite. Let also (G, H, E, D) be a monoid homomorphic encryption scheme. Consider an efficiently computable R-character $\chi : H \to A$, where A is a finite R-algebra, and consider also the monoid algebra $R[G]$. As explained above, the monoid homomorphism $D : G \to H$ induces the R-algebra homomorphism $D_R : R[G] \to R[H]$. At the same time the R-character χ induces the R-algebra

homomorphism $\chi_R : R[H] \to R[A]$. We define the R-algebra homomorphism Dec as the composition $\mathrm{Dec} = \epsilon \circ \chi_R \circ D_R : R[G] \to R[H] \to R[A] \to A$. It is easy to check that Dec is defined by the following formula:

$$\mathrm{Dec}\left(\sum_{g\in G} a_g[g]\right) = \sum_{g\in G} a_g \chi(D(g)).$$

Let us denote by S the image of χ in A. For Dec to remain secure, one needs the assumption that $|S| \geq 2$, i.e. χ is not the trivial character, a condition always assumed in our blueprint. We shall suppose that the pair (A, S) satisfies the following condition: there exist a *fixed* k-tuple $(r_1, \ldots, r_k) \in R^k$, where $k \geq 1$, such that the set containing the elements of the form $\sum_{i=1}^k r_i s_i$ with $s_i \in S$, $\forall i$ (not necessarily distinct) is the whole R-algebra A. Notice that a necessary condition for the existence of such a tuple is that A is generated as an R-module by S. If this is not the case then A may be replaced by its R-submodule generated by S. Indeed, since S is closed under multiplication the R-submodule of A generated by S is an R-subalgebra of A. On the other hand, this necessary condition is not sufficient to ensure the existence of a tuple as above, however if S contains 0 the two conditions are equivalent because A is a finite ring.
Now, we describe the ring homomorphic encryption scheme $(R[G], A, \mathrm{Enc}, \mathrm{Dec})$:

- `Enc`: Let S be the image of χ in A and consider a fixed tuple $(r_1, \ldots, r_k) \in R^k$, where $k \geq 2$, such that the set containing the elements of the form $\sum_{i=1}^k r_i s_i$ with $s_i \in S$ is the whole R-algebra A. For a plaintext $m \in A$ consider $(h_1, \ldots, h_k) \in H^k$ such that $m = \sum_{i=1}^k r_i \chi(h_i)$. Then
$$\mathrm{Enc}(m) = \sum_{i=1}^k r_i [E(h_i)].$$
- `Dec`: The decryption algorithm is given by:
$$\mathrm{Dec}\left(\sum_{g\in G} a_g[g]\right) = \sum_{g\in G} a_g \chi(D(g)).$$

The main result of this section is the following:

Theorem 7. *The encryption scheme* $(R[G], R, \mathrm{Enc}, \mathrm{Dec})$ *is a ring homomorphic encryption scheme.*

Proof. As we have seen above, given the homomorphic properties of D and χ we get that Dec is actually a ring homomorphism. The security of the scheme is the same as the security of the monoid encryption scheme (G, H, E, D) since no information and no additional security was revealed or added through the steps describing the encryption algorithm. The choice of the generating set $(r_1, \ldots, r_k)$ described in `Enc` ensures that the semantic security of the ring homomorphic encryption scheme reduces to the semantic security of the monoid homomorphic encryption scheme.

One should make the difference (as we will notice in some examples bellow) between the probability of plaintexts generated by choosing random elements in S and producing the plaintext $\sum_{i=1}^{k} r_i s_i$ and the probability of a certain plaintext to be encrypted. In essence the choice of the set $(r_1, \ldots, r_k)$ ensures that no plaintext is left outside the encryption process. The bigger the set S is inside R the smaller the number k can be chosen.

The efficiency of the encryption scheme is k times less than the efficiency of the monoid homomorphic encryption scheme since basically the length of the ciphertext obtained by `Enc` is approximately k times the length of a ciphertext obtained by `E`. In particular, the encryption algorithm has polynomial size in the security parameter and the output has polynomial length in the same parameter if and only if the monoid homomorphic encryption scheme has. The decryption algorithm `Dec` has also the same efficiency as the algorithm D in the monoid homomorphic encryption scheme.

Having fixed the encryption scheme, the length of the ciphertexts obtained by performing algebraic computations is *finite* since all computations take place in $R[G]$ which is a finite ring, i.e. the scheme is *bounded*.

In conclusion, all of the algebraic properties as well as the properties required for privacy and security are satisfied by the ring homomorphic encryption scheme constructed above if one starts with a private and secure monoid homomorphic encryption scheme. □

Remark 8. *Even though the ciphertext resulted by computing a polynomial on cyphertexts remains finite, its length is growing up to a certain point exponentially. The maximal length of an element in $R[G]$ is huge for practical purposes. Therefore, for implementation in cloud computing, one needs an additional process to ensure a practical finiteness of an output after an algebraic manipulation on ciphertexts.*

Remark 9. *In general the plaintext space $\mathcal{P}$ need not be the whole R-algebra A, but just a subring of it. Therefore one can encrypt only the desired plaintexts. We will actually need this remark in the next examples.*

Remark 10. *The idea of producing FHE schemes out of a group homomorphic encryption scheme was also taken under consideration in [27]. However, the resulted encryption schemes therein are not bounded.*

We end this section by mentioning that the blueprint works word-by-word for the case of semi-group homomorphic encryption schemes. The resulting scheme is a (non-unital) ring homomorphic encryption scheme. Also, even if we start with a *compact* monoidal encryption scheme, the above blueprint produces only a *bounded* ring homomorphic encryption scheme.

5 A ring HE scheme using Monoid Algebra

In this section we discuss the first example of a FHE scheme based on the above blueprint. Hereafter the ring R is the field $\mathbf{F}_2$. If (G, H, E, D) is a group homomorphic encryption scheme, since H is a group, the image of any character $\chi : H \to A$ is also a group, so that if $A = \mathbf{F}_2$ then any character is trivial. This is the reason why we need to consider an $\mathbf{F}_2$-algebra A different from $\mathbf{F}_2$ itself and a group encryption scheme (G, H, E, D) such that

there exist a nontrivial $\mathbf{F}_2$-character $\chi : H \to A$ (in particular $\gcd(|H|, |A^\times|) > 1$). The simplest (but not the most efficient) example of such situation is the following variant of Benaloh cryptosystem (cf. [5]), which is an extension of the Goldwasser-Micali cryptosystem (cf. [20]).

We proceed to the explicit description of the group encryption scheme we are referring to. Choose p, q two large primes such that $p \equiv 1 \pmod 3$, $p \not\equiv 1 \pmod 9$, $q \equiv 1 \pmod 3$, and let $N = p \cdot q$. Let $G := (\mathbf{Z}/N\mathbf{Z})^\times$ be the group of invertible elements mod N, and let $\pi_p : (\mathbf{Z}/N\mathbf{Z})^\times \to (\mathbf{Z}/p\mathbf{Z})^\times$ and $\pi_q : (\mathbf{Z}/N\mathbf{Z})^\times \to (\mathbf{Z}/q\mathbf{Z})^\times$ be the projection maps. Fix two primitive third roots of unity: $\omega_p \in (\mathbf{Z}/p\mathbf{Z})^\times$ and $\omega_q \in (\mathbf{Z}/q\mathbf{Z})^\times$, i.e. let $\omega_p = g_p^{\frac{p-1}{3}}$, where g_p is a generator of the cyclic group $(\mathbf{Z}/p\mathbf{Z})^\times$, and similarly for ω_q. We let $\phi : (\mathbf{Z}/p\mathbf{Z})^\times \to \mathbf{Z}/3\mathbf{Z}$ be the group homomorphism defined by: $\phi(x) = i$ if and only if $x^{\frac{p-1}{3}} = \omega_p^{i \cdot \frac{p-1}{3}}$. The morphism ϕ is efficiently computable because raising x to the power $\frac{p-1}{3}$ can be done in $\log p$ steps, and then by Fermat's Little Theorem $x^{\frac{p-1}{3}}$ is a third root of unity modulo p, therefore $x^{\frac{p-1}{3}} \in \{1, \omega_p, \omega_p^2\}$, so that ϕ is well defined if and only if $p \not\equiv 1 \pmod 9$.

For encryption, let η be the unique element of G, such that $\pi_p(\eta) = \omega_p$ and $\pi_q(\eta) = \omega_q$. The group encryption scheme $(G, \mathbf{Z}/3\mathbf{Z}, E, D)$ is given as follows:

- `Setup`(1^λ): Choose two large enough primes (to ensure semantic security) $p = p(\lambda), q = q(\lambda)$ such that $p \equiv 1 \pmod 3$, $p \not\equiv 1 \pmod 9$, and $q \equiv 1 \pmod 3$.
- `PublicKeygen`: Set $N = pq$. Fix a primitive third root of unity modulo p, say ω_p, and a primitive third root of unity modulo q, say ω_q. Let $\eta \in G$ be such that: $\pi_p(\eta) = \omega_p$ and $\pi_q(\eta) = \omega_q$. The public key is the pair $(N, \gamma := \eta \cdot u^3)$, where u is a random element of G.
- `SecretKeygen`: The secret key is the prime p.
- `E`: To encrypt $m \in \mathbf{Z}/3\mathbf{Z}$, choose a random $y \in G$ and let $E(m) = \gamma^m y^3$.
- `D`: The decryption of $c \in G$ is given by $D(c) = \phi(\pi_p(c))$.

To describe the associated FHE scheme, let $\mu_3 = \{1, \omega, \omega^2\}$ be the group of third roots of unity of $\overline{\mathbf{F}}_2$ (the algebraic closure of $\mathbf{F}_2$). Then the field with four elements consists of $\mathbf{F}_4 = \{0, 1, \omega, \omega^2\}$ (here ω satisfies the equation $\omega^2 + \omega + 1 = 0$). Let $A = \mathbf{F}_4$ and let the character $\chi : \mathbf{Z}/3\mathbf{Z} \to \mathbf{F}_4$ be defined by $m \mapsto \omega^m$. Notice that $\chi : \mathbf{Z}/3\mathbf{Z} \to \mathbf{F}_4^\times$ is an isomorphism of groups so that we shall denote by χ^{-1} its inverse. Now the encryption and decryption algorithms of the FHE scheme are given by:

- `Enc`: To encrypt the bit $m = 0$ encode ω twice using the above scheme to get $E_1(\chi^{-1}(\omega)), E_2(\chi^{-1}(\omega))$ and then set $Enc(0) = [E_1(\chi^{-1}(\omega))] + [E_2(\chi^{-1}(\omega))]$.

  For $m = 1$ let $E_1(\chi^{-1}(\omega))$ and $E_2(\chi^{-1}(\omega^2))$ be encryptions of ω and ω^2 respectively, and set $Enc(1) = [E_1(\chi^{-1}(\omega))] + [E_2(\chi^{-1}(\omega^2))]$.
- `Dec`: For $c = \sum_{g \in G} r_g[g] \in \mathbf{F}_2[G]$, let $Dec(c) = \sum_{g \in G} r_g \chi(D(g))$.

One sees easily that the homomorphic properties and the security of the scheme are exactly as in the main theorem.

Remark 11. *We notice that this encryption has another interesting feature. It detects with high probability if the message sent for encryption has been altered during the transmission. Concretely, for every bit encrypted as* $[g_1] + [g_2]$*, if one changes randomly the elements* g_1 *and* g_2 *there is a* $1/2$ *probability that the altered ciphertext does not decrypt to* 0 *or* 1 *(but rather to* ω *or* ω^2*). Of course, for a large text that has been tempered during the encryption, the probability of detecting if the message has been altered or not is very high. This feature is very useful in many applications and none of the "error"-based encryptions have it.*

6 A symmetric bounded FHE scheme

6.1 $k = 1$

As we mentioned above, there is a big amount of efficiency in the monoid algebra blueprint if one starts with a nontrivial (surjective) monoid homomorphic encryption $(G, (\mathbf{F}_2, \cdot), E, D)$. In fact, we shall see in a moment that the efficiency of the monoid algebra encryption is the same as for the monoid homomorphic encryption (i.e. one can choose $k = 1$ in the blueprint encryption algorithm), while the size of the ciphertext which is an output of a circuit grows by the number of *additions* in the associated polynomial. This phenomenon is exactly the opposite to the growth in the size of the ciphertext encrypted by RLWE based schemes or other previous methods (where the costly operation is the multiplication).

We start by describing a monoid homomorphic encryption scheme, which is the most natural example for a scheme as in the blueprint with $k = 1$. Unfortunately, this scheme, in its incipient form, is vulnerable for the statistical attacks described bellow. We will describe afterwords an improvement which takes care of this vulnerability. We choose to present the simple scheme for the sake of the reader to follow better the ideas involved.

Let G be the set $\mathbf{F}_2^n$, with the monoid structure defined by component-wise multiplication. Basically, G is the commutative monoid generated by n idempotents with no "extra" relations. The scheme is defined as follows:

- `Setup`(1^λ): Choose the dimension parameter $n = n(\lambda)$, and the integers $d = d(\lambda)$, $s = s(\lambda)$ such that $n = 2sd$.
- `SecretKeygen`: Choose a subset S of $\{1, 2, ..., n\}$ of size s. Set the secret key to be S.
- `E`: To encrypt a bit $m \in \mathbf{F}_2$, choose d random numbers $i_1, i_2, ..., i_d$ from the set $\{1, 2, ..., n\}$ such that there are exactly m of them in the secret key set S. Set $E(m)$ to be the vector in G, whose components corresponding to the indices $i_1, i_2, ..., i_d$ are equal to 0 and the others are equal to 1.
- `D`: To decrypt a cyphertext c using the secret key S, set $D(c) = 0$ if c has at least one component equal to 0 corresponding to a index from S and $D(c) = 1$, otherwise.

Correctness. It is an easy exercise to check that $D(c_1 \cdot c_2) = D(c_1) \cdot D(c_2)$, so that the above scheme is a compact monoid homomorphic encryption scheme.

Security. We discuss first the security of the scheme under brute-force attack. To achieve 2^λ security against brute-force attacks we require the parameters n, d, s to satisfy the following conditions $s, d = \Theta(\lambda)$, and since $n = 2sd$ we obtain $n = \Theta(\lambda^2)$. Using brute-force attack an adversary needs to try $\binom{n}{s}$ subsets of $\{1, 2, ..., n\}$ in order to find the secret key S. Since by Stirling's formula $\binom{n}{s} = 2^{\omega(\lambda \log \lambda)}$ we obtain the required security. A more skilfull adversary that has access to an encrypted text can try the following attack: compute the relative frequency of occurrence of 0 on the ith component of ciphertexts. Of course, if the text is large enough, these relative frequencies are approximately equal to the corresponding probabilities. If $i \in S$ then the probability that the ith component equals 0 is: $\frac{1}{2} \cdot \frac{1}{s} + \frac{1}{2} \cdot 0 = \frac{1}{2s}$. On the other hand, if $i \notin S$ then the probability equals $\frac{1}{2} \cdot \frac{d}{n-s} + \frac{1}{2} \cdot \frac{d-1}{n-s} = \frac{2d-1}{2(n-s)}$. These two expressions must be equal, otherwise the adversary learns whether a component is in S or not, which gives non-negligible information about S (in fact it almost determines S). Equating the expressions we get that $n = 2sd$, which is the obstruction required in the `Setup`. Unfortunately, the above monoid encryption scheme is vulnerable to Known Plaintext Attack, Chosen Plaintext Attack and Chosen Ciphertext Attack (we refer to [4] for definitions). Indeed, a list of encryptions of 1 reveals information about certain components that are not in S, so that if the list is long enough then the secret key S can be determined with high accuracy. The scheme can be modified in order to achieve security against the above attacks but not without affecting the efficiency in cloud computing.

The above scheme as it is presented above, seems to take care also of the above mentioned statistical attack. Unfortunately, this is not the case for an attack based on a even more attentive analysis of the statistics of the encrypted text. As we saw above, the condition $n = 2sd$ imposed in the `Setup`, already guaranties that no information about the secret key (or the plaintext in that matter) can be drawn if one tries to use the mere statistical appearance of 0's and 1's in the plaintext by analyzing the 0 and 1 appearance in the cyphertext on a certain position. We call this, the first moment statistical attack. However, more information can be drawn if one considers pairs of indices (positions), or more generally, l-tuples of indices and compute for each of the 2^l possible configurations of 0 and 1 appearing on this tuples, the statistics in the cyphertext. We will call this the l^{th}-moment attack. Observing statistical differences can be useful in deducing information about the secret key with high probability. For example, in the above scheme, no two indices corresponding to the secret key are ever both equal to 0, so for such pair, the statistical appearance of $(0, 0)$ is mathematically 0, while for any other type of pair of indices, the statistical appearance of $(0, 0)$ is NOT zero.

On the other hand, if we can take care of the l^{th}-moment attack for any $l \geq 1$, then we can be sure that no other attack based on statistical analysis of the ciphertext can be designed since the statistics in this case is determined by its moments.

In what follows, we present the improved version of the above scheme. The algorithms that we will modify are the setup algorithm and the encryption algorithm. We shall discuss afterwords the security against the l^{th}-moment attack.

`Setup`(1^λ): Choose the dimension parameter $n = n(\lambda)$, and the integers $d = d(\lambda) < s =$

$s(\lambda)$ to be defined as follows: Use the Simplex algorithm to solve the following problem: set $P(k,u,v) := \frac{\binom{v-k}{u-k}}{\binom{v}{u}}$. Fix ϵ an acceptable statistical error (of order $1/D$, where D is the size of the document in clear).
Fix $X = (X(1), \ldots, X(d))$ a random variable with probabilities to be determined by the simplex algorithm. For each $r = 1, \ldots, \sqrt{2d}$ and each $i = 1, \ldots, r$, set the expression

$$E(X,i,r,s,d,n) := \sum_{k=1}^{d} X(k)(P(i,k,s)P(r-i,d-k,n-s) - P(r,d-k,n-s))$$

where $P(k,u,v)$ is set to be 0 if $k > \min(u,v)$. The system of approximate equations to be solved takes the form:

$$\left| E(X,i,r,s,d,n) - \frac{1}{2}P(r,d,n-s) \right| < \epsilon,$$

for all i, r in the given range, $X(k) \geq 0$ for all k and $\sum_{k=1}^{d} X(k) = \frac{1}{2}$.

`SecretKeygen`: Choose a subset Sk of $\{1, 2, ..., n\}$ of size s. Set the secret key to be Sk.

`E`: To encrypt $1 \in \mathbf{F}_2$, choose d random numbers $i_1, i_2, \ldots, i_d$ from the set $\{1, 2, ..., n\}$ such that none of them is in the secret key set, Sk. Set $E(1)$ to be the vector of length n with zeroes on the chosen positions and 1 everywhere else. To encrypt $0 \in \mathbf{F}_2$, choose k in $\{1, \ldots, d\}$ with probability $2X(k)$. Choose k random positions in Sk and $d-k$ random positions outside Sk and set $E(0)$ to be the vector of length n with zeroes on the chosen positions and 1 everywhere else.

`D`: To decrypt a cyphertext c using the secret key Sk, set $D(c) = 0$ if c has at least one component equal to 0 corresponding to an index from Sk and $D(c) = 1$, otherwise.

Efficiency. Now, let's see how this improves the efficiency in the FHE scheme obtained from the above monoid homomorphic scheme. First of all, we can take R to be equal to $\mathbf{F}_2$, $H = R$ and χ to be the identity character. In this case, k in the `Enc` algorithm can be taken to be equal to 1, so that the size of an encrypted message will be $\mathcal{O}(\lambda)$. For a polynomial P associated to a circuit, the size of the encryption $P(c_1, \ldots, c_r)$ will be at most $L(P) \cdot n$, where $L(P)$ is the number of monomials of the polynomial P. Indeed, since each ciphertext c_i has length n, each monomial will still have n bits since the multiplication is done within G. Next, we formally add the monomials resulting in a length at most $L(P) \cdot n$.

This feature is very effective compared to other leveled FHE in the literature because the size of an output of a circuit applied to fresh ciphertexts is linear in the number of monomials of the polynomial associated to the circuit, which for a lot of applications is much smaller than its total depth (number of additions *and* multiplications).

In fact, in practice, the above scheme is far more efficient than the mathematical prediction since in practice one works only within a fixed range of values for the security parameter, etc. We ran the above simplex algorithm where we fixed as acceptable statistical error

$\epsilon = 2^{-30}$ and found an acceptable range of solutions for $(d, s, n) = (22, 31, 1010)$. The solution is found almost instantaneously on a regular PC (our algorithm, which can be improved, found it in 0.17 sec on a 2.66 GHz I7, 8Gb RAM). In fact, the solution will guaranty that no r-tuple statistics will reveal significant information because already at the 6^{th} moment (which is guarantied by the simplex algorithm to reveal no information), the probability of encryption with a fixed 6-tuple is less than ϵ. The brute force attack is also impossible since $\binom{1010}{31}$ is of order 2^{196}, which is far beyond today's (or near future) capacity of processing. Therefore, for documents of size of Gb-order, our encryption is proved to be secure and sufficiently efficient.

6.2 $k = 2$

There is a simple way to improve the scheme so it becomes secure for KPA, CPA, etc: one can choose $k = 2$ in the blueprint, i.e. $Enc(0) := [E_1(0)] + [E_2(0)]$ or $[E_1(1)] + [E_2(1)]$ while $Enc(1) := [E_1(0)] + [E_2(1)]$ or $[E_1(1)] + [E_2(0))]$, where by "or", we mean uniformly chosen.

Now the scheme becomes resistant to KPA and CPA because one is unable to separate the components of $Enc(m)$ which correspond to $E(0)$ or to $E(1)$. Thus the only possible attacks would be those discussed above.

The drawback of this scheme, besides the fact that fresh ciphertexts have double sizes, is that the size of the ciphertext grows exponentially (*up to some finite point*) with the multiplicative depth of a circuit, making the practical implementation rather unattractive. The scheme behaves exactly as the scheme presented in section 5.

The ring homomorphic encryption scheme described above can be transformed into a public key encryption scheme using lists of encryptions of 0 and 1 (see [30]). However, the size of fresh ciphertexts will become too large to practically implement the scheme.

It becomes clear that for a practical implementation of encryption schemes produced by the blueprint presented above, one needs yet another algorithm, which may be called generically "compactification", that takes as input a large size ciphertext and outputs an equivalent ciphertext with lower (practically manageable) size. This process is similar in essence with the reliniarization/key switching processes in the RLWE-based algorithms. The drawback in the later algorithms is the growth of the noise which is solved using the techniques mentioned earlier. In our case the drawback is the size of the ciphertext after evaluation.

6.3 A previous example

The example in the previous section is not the first monoid homomorphic encryption scheme over the monoid $(\mathbf{F}_2, \cdot)$. The first one, to our knowledge, is the scheme presented in [31]. This scheme is secure but there is a downside; more precisely the scheme is capable to decode correctly any "AND" operation with overwhelming probability (more precisely, that is $1 - \frac{1}{2^t}$). In other words, the scheme does not satisfy the equality $D(x \cdot y) = D(x) \cdot D(y)$ for any ciphertexts x and y. To describe the scheme we shall use the group homomorphic encryption scheme of Goldwasser-Micali. Since the scheme is well known we decided not to recall it here (see [20] for details).

- `Setup`(1^λ): Choose two large primes $p = p(\lambda)$, $q = q(\lambda)$ to ensure security of the Goldwasser-Micali scheme. Choose $\ell = \ell(\lambda)$ of size $O(\lambda)$.

- `Keygen`: Compute $N = pq$. The public key is the same as in the Goldwasser-Micali scheme. The secret key is p or q.

- `E`: If $m = 1$ set $v = (0, ..., 0) \in \mathbf{F}_2^\ell$. If $m = 0$ set $v = (v_1, ..., v_n) \in \mathbf{F}_2^\ell$, where the components v_i are randomly chosen in $\{0, 1\}$, such that not all are equal to 0. Encrypt each component of v with the Goldwasser-Micali scheme to get a vector in $[(\mathbf{Z}/N\mathbf{Z})^\times]^\ell$.

- `D`: To recover the plaintext from the cyphertext $c \in [(\mathbf{Z}/N\mathbf{Z})^\times]^\ell$, first decrypt each component of c, and then if the obtained vector is the 0-vector the message decrypts to 1, else to 0.

It is easy to see that the semantic security of this scheme reduces down to the security of the Goldwasser-Micali scheme. Let us describe the AND operation on the ciphertext space. Let x and y be two ciphertexts then $z := \mathrm{AND}(x, y)$ is defined as follows:

1. Choose uniformly at random two $\ell \times \ell$ matrices over $\mathbf{F}_2$ until two nonsingular matrices $A = (a_{ij})$ and $B = (b_{ij})$ are found.

2. Let $x = (x_1, ...x_\ell)$, $y = (y_1, ...y_\ell)$. Compute $v = (v_1, ..., v_\ell)$, where

$$v_i = \prod_{j, a_{ij}=1} x_j \cdot \prod_{j, b_{ij}=1} y_j.$$

3. Pick uniformly at random $r_1, ..., r_\ell \in (\mathbf{Z}/N\mathbf{Z})^\times$ and set $z = (r_1^2 v_1, ..., r_\ell^2 v_\ell)$.

Correctness. It is an easy exercise to check that the probability

$$P\left(D(\mathrm{AND}(x, y)) = D(x) \cdot D(y)\right) > 1 - \frac{1}{2^\ell}.$$

We can apply our blueprint to this monoid encryption scheme to produce a ring homomorphic encryption scheme over $\mathbf{F}_2$. A simple computation shows that in the resulted scheme the decryption algorithm evaluates correctly a ciphertext obtained by a homomorphic evaluation of a circuit $\mathcal{C}$ with probability of size $1 - \frac{L}{2^l}$, where $L = L(P_\mathcal{C})$ is the number of monomials (of degree at least 2) of the polynomial attached to $\mathcal{C}$. Therefore, the resulted scheme is only leveled.

7 Conclusions

1) In the case of the encryption scheme described in Section 5 the maximal length of an output of a circuit is the cardinality of G . Using Goldwasser-Micali cryptosystem, in order to achieve secure encryption, one takes $n := \mathrm{Log}(|G|) = 2048$, therefore a maximal output would have $2048 \cdot 2^{2048}$ bits which is totally unpractical (see the discussion in Section 1.2). Our blueprint however, can use any group homomorphic encryption scheme. For a group homomorphic encryption schemes which is based on the discrete logarithm problem, the size

of the cyphertext space is required to be at least 2^{160} in size (see [32] and [34]), therefore in our blueprint the maximal size of an output would require around $160 \cdot 2^{160}$ bits which is again not practical, but is much closer to what we mean by practical. This size is actually reachable up to a small constant using ECC or HECC.

2) From a theoretical point of view, it is better to see the encryption process described in section 6 as a stochastic process in order to get better performance. However, in the practical range that our scheme works, it doesn't seem to get much better than the scheme presented above. One idea of making the encryption process to be stochastic is to attach to each of the n positions a probability vector, where the component k corresponds to the extraction probability of that position in the case the encryption has k zeroes corresponding to the secret key. The simplex algorithm becomes a stochastic simplex algorithm, which in practice is hard to compute. But this strategy doesn't necessarily get stuck; one can use dynamic procedures to compute on the run those probabilities: during the encryption process, modify accordingly the probabilities so the moment's deviation gets within the accepted error. (One could call this, AI encryption strategy).

3) The authors' work in progress can actually prove that in a very well defined manner these two examples of encryption schemes are essential examples of commutative ring homomorphic encryption schemes over $\mathbf{F}_2$. These two examples correspond to semigroup algebra homomorphic encryption schemes where the semigroup has only one idempotent, respectively the semigroup is made up only of idempotents.

Acknowledgments

This work was partially supported by the Romanian National Authority for Scientific Research (CNCS-UEFISCDI) under the project PN-II-PT-PCCA-2011-3 (ctr. 19/2012).

References

[1] Albrecht, M.R., Farshim, P., Faugère, J.-C., Perret, L.: *Polly Cracker, Revisited*, In Lee, D.H. ed., ASIACRYPT 2011, Lecture Notes in Computer Science, vol. 7073, Springer, 2011, pp. 179196.

[2] Armknecht, F., Katzenbeisser, S., Peter, A.: *Group Homomorphic Encryption: Characterizations, Impossibility Results, and Applications* , in Designs, Codes and Cryptography, vol. 67, no. 2, 2013, pp. 209 - 232.

[3] Barkee, B., Can, D.C., Ecks, J., Moriarty, T., Ree, R.F.: *Why you cannot even hope to use Gröbner bases in public key cryptography: An open letter to a scientist who failed and a challenge to those who have not yet failed*, Journal of Symbolic Computations 18(6), 1994, pp. 497501.

[4] Bellare, M., Desai, A., Jokipii, E., Rogaway, P.: *A concrete security treatment of symmetric encryption*, In Proceedings of the 38th Symposium on Foundations on Computer Science, IEEE, 1997, pp. 394 - 403.

[5] Benaloh, J.: *Dense Probabilistic Encryption*, in Proceedings of the Workshop on Selected Areas of Cryptography, 1994, pp. 120 - 128.

[6] Brakerski, Z., Vaikuntanathan, V.: *Efficient fully homomorphic encryption from (standard) LWE*, R. Ostrovsky editor, IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs 2011, pp. 97 - 106, unpublished, longer version available at eprint.iacr.org/2011/344.pdf .

[7] Brakerski, Z.: *Fully homomorphic encryption without modulus switching from classical GapSVP*, In CRYPTO 2012, pp. 868 - 886.

[8] Brakerski, Z., Gentry, C., Vaikuntanathan, V.: *(Leveled) fully homomorphic encryption without bootstrapping*, Proceedings of the 3rd Innovations in Theoretical Computer Science Conference, ITCS 2012, pp. 309 - 325.

[9] Broadbent,A., Jeffery, S.: *Quantum homomorphic encryption for circuits of low T-gate complexity*, arXiv:1412.8766v2

[10] Coron, J-S., Mandal, A., Naccache, D., Tibouchi, M.: *Fully homomorphic encryption over the integers with shorter public keys*, P. Rogaway editor, Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara 2011, Lecture Notes in Computer Science, vol. 6841, Springer, 2011, pp. 487 - 504.

[11] dit Vehel, F.L., Marinari, M.G., Perret, L., Traverso, C.: *A survey on Polly Cracker systems*, In: Gröbner Bases, Coding and Cryptography, Springer, 2009, pp. 285 - 305.

[12] Ducas, L., Micciancio, D.: *FHEW: Homomorphic Encryption Bootstrapping in less than a second*, Advances in Cryptology - EUROCRYPT 2015, Lecture Notes in Computer Science, Vol. 9056, 2015, pp. 617 - 640.

[13] Fellows, M., Koblitz, N.: *Combinatorial cryptosystems galore!*, In Finite Fields: Theory, Applications, and Algorithms, vol. 168 of Contemporary Mathematics, AMS 1994, pp. 51 - 61.

[14] Gentry, C.: *A fully homomorphic encryption scheme*, PhD thesis, Stanford University, 2009.

[15] Gentry, C.: *Fully homomorphic encryption using ideal lattices*, In STOC 2009, Proceedings of the 41st annual ACM symposium on Theory of computing, pp. 169 - 178.

[16] Gentry, C.: *Computing arbitrary functions of encrypted data*, Communications of the ACM, Vol. 53, Issue 3, March 2010, pp. 97 - 105.

[17] Gentry, C., Halevi, S., Smart N.P.: *Better Bootstrapping in Fully Homomorphic Encryption*, Public Key Cryptography - PKC 2012, Lecture Notes in Computer Science, Vol. 7293, 2012, pp. 1 - 16.

[18] Gentry, C., Sahai, A., Waters, B.: *Homomorphic Encryption from Learning with Errors: Conceptually-Simpler, Asymptotically-Faster, Attribute-Based*, Advances in Cryptology - CRYPTO 2013, Lecture Notes in Computer Science, Vol. 8042, 2013, pp. 75 - 92.

[19] Goldreich, O., Goldwasser, S., Halevi, S.: *Public-key cryptosystems from lattice reduction problems*, Advances in Cryptology - CRYPTO 1997, Vol. 1294 of Lecture Notes in Computer Science, pp. 112 - 131.

[20] Goldwasser, S., Micali, S.: *Probabilistic Encryption*, Journal of Computer and System Sciences 28, 1984, pp. 270 - 299.

[21] Grigoriev, D., Ponomarenko, I.: *Homomorphic Public-Key Cryptosystems over Groups and Rings*, Quaderni di Mathematica, Vol. 13, 2004, pp. 304 - 325.

[22] Halevi, S., Shoup, V.: *Algorithms in HElib*, Garay, J.A. and Gennaro, R. editors; CRYPTO 2014, Part I, Lecture Notes in Computer Science, Vol. 8616, Springer, 2014, pp. 554 - 571.

[23] Halevi, S., Shoup, V.: *Bootstrapping for HElib*, Advances in Cryptology - EUROCRYPT 2015, Lecture Notes in Computer Science, Vol. 9056, 2015, pp 641 - 670.

[24] Herold, G.: *Polly Cracker, Revisited, Revisited*, Proceedings of 15th International Conference on Practice and Theory in Public Key Cryptography, Lecture Notes in Computer Science, vol. 7293, Springer, 2012, pp. 17 - 33.

[25] Hoffstein, J., Pipher, J., Silverman, J.H.: *NTRU: A Ring Based Public Key Cryptosystem*, Algorithmic Number Theory (ANTS III), Portland, June 1998, Lecture Notes in Computer Science 1423 (J.P. Buhler, ed.), Springer-Verlag, Berlin, 1998, pp. 267 - 288.

[26] Hoffstein, J., Pipher, J., Silverman, J.H.: *NSS: an NTRU lattice-based signature scheme*, Advances in cryptology - EUROCRYPT 2001, Innsbruck, Lecture Notes in Computer Science, Vol. 2045, Springer, 2001, pp. 211 - 228.

[27] Melchor, C.A., Gaborit, P., Herranz, J.: *Additively Homomorphic Encryption with d-Operand Multiplications*, Advances in Cryptology - CRYPTO 2010, Lecture Notes in Computer Science, vol. 6223, Springer, 2010, pp. 138 - 154.

[28] Rivest, R., Adleman, L., Dertouzos, M.: *On data banks and privacy homomorphisms*, In Foundations of Secure Computation, Academic Press, 1978, pp. 169 - 177.

[29] Regev, O.: *On lattices, learning with errors, random linear codes, and cryptography*, In Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing, STOC 2005, pp. 84 - 93. .

[30] Rothblum, R.: *Homomorphic Encryption: From Private-Key to Public-Key*, Theory of Cryptography, Lecture Notes in Computer Science, Vol. 6597, 2011, pp. 219 - 234.

[31] Sander, T., Young, A., Yung, M.: *Non-Interactive CryptoComputing For NC^1*, Proceedings of the 40th IEEE Symposium on Foundations of Computer Science, FOCS 1999, pp. 554 - 566.

[32] Scholten, J., Vercauteren, F.: *An Introduction to Elliptic and Hyperelliptic Curve Cryptography and the NTRU Cryptosystem*, To appear in B. Preneel Ed., State of the Art in Applied Cryptography, COSIC '03, Lecture Notes in Computer Science, Springer, 2004.

[33] Sen, J.: *Homomorphic Encryption: Theory & Application*, the article is available at http://arxiv.org/abs/1305.5886.

[34] Shoup, V. *Lower bounds for discrete logarithms and related problems*, In Advances in cryptology, EUROCRYPT 1997, Lecture Notes in Computer Sciences, Vol. 1233, Springer, 1997, pp. 256 - 266.

[35] Smart, N., Vercauteren, F. *Fully homomorphic encryption with relatively small key and ciphertext sizes*, In P. Nguyen and D. Pointcheval, editors, Public Key Cryptography, vol. 6056 of Lecture Notes in Computer Science, Springer, 2010, pp. 420 - 443.

[36] Steinfeld, R.: *NTRU cryptosystem: Recent developments and emerging mathematical problems in finite polynomial rings*, In Algebraic Curves and Finite Fields, Cryptography and Other Applications, De Gruyter, 2014, pp. 179 - 212.

[37] Stehlé, D., Steinfeld, R.: *Making NTRU as Secure as Worst-Case Problems over Ideal Lattices*, Advances in Cryptology - EUROCRYPT 2011, Lecture Notes in Computer Science, Vol. 6632, 2011, pp 27 - 47.

[38] van Dijk, M., Gentry, C., Halevi, S., Vaikuntanathan, V.: *Fully hohomomorphic encryption over the integers*, In EUROCRYPT 2010, pp. 24 - 43.

Mugurel Barcau, Researcher, CertSign S.A., Bucharest, Romania
and
Institute of Mathematics of the Romanian Academy "Simion Stoilow", Str. Grivitei 21, Bucharest, Romania
E-mail address: barcau@yahoo.com

Vicenţiu Paşol, Researcher, CertSign S.A., Bucharest, Romania
and
Institute of Mathematics of the Romanian Academy "Simion Stoilow", Str. Grivitei 21, Bucharest, Romania *E-mail address*: vpasol@yahoo.com