An Efficient Fully Homomorphic Encryption Scheme

Ahmed El-Yahyaoui and Mohamed Dafir Ech-Cherif El Kettani
(Corresponding author: Ahmed El-Yahyaoui)
Information Security Research Team, CEDOC ST2I ENSIAS, Mohammed V University in Rabat, Morocco
(Email: ahmed_elyahyaoui@um5.ac.ma)
(Received Mar. 4, 2017; revised and accepted July 13, 2017)

Abstract

Cloud computing is a new paradigm of information technology and communication. Performing big and complex computations in a context of cloud computing and big data is highly appreciated today. Fully homomorphic encryption (FHE) is a powerful category of encryption schemes that allows working with the data in its encrypted form. It permits us to preserve confidentiality of our sensible data and to benefit from cloud computing powers. Currently, it has been demonstrated by many existing schemes that the theory is feasible but the efficiency needs to be dramatically improved in order to make it usable for real applications. One subtle difficulty is how to efficiently handle the noise. This article aims to introduce an efficient fully homomorphic encryption scheme based on a new mathematic structure that is noise free.

Keywords: Fully Homomorphic Encryption; Lipschitz Integers; Probabilistic Transform; Quaternion

1 Introduction

Fully homomorphic encryption is a type of encryption cryptosystems that support arbitrary computations on ciphertexts without ever needing to decrypt or reveal it. In a context of cloud computing and distributed computation, this is a highly precious power. In fact, a significant application of fully homomorphic encryption is to big data and cloud computing. In these two situations, the processed data often contains private information about individuals or corporate secrets that would cause great harm if they fell into the wrong hands. Generally, FHE is used in outsourcing complex computations on sensitive data stored in a cloud as it can be employed in specific applications for big data like secure search on encrypted dat and private information retrieval. It was an open problem, conjectured by Rivest, Adleman and Dertozous [14] in 1978, until the revolutionary work of Gentry in 2009 [8] which opens the curtain for the study of fully homomorphic encryption. In his thesis, Gentry proposed the first adequate fully homomorphic encryption scheme by exploiting properties of ideal lattices.

Gentry's construction is based on his bootstrapping theorem which provides that given a somewhat homomorphic encryption scheme (SWHE) that can evaluate homomorphically its own decryption circuit and an additional NAND gate, we can pass to a 'leveled' fully homomorphic encryption scheme and so obtain a FHE scheme by assuming circular security. The purpose of using bootstrapping technique is to allow refreshment of ciphertexts and reduce noise after its growth.

Gentry's construction is not a single algorithm but it is considered as a framework that inspires cryptologists to build new fully homomorphic encryption schemes [6, 9, 15, 17]. A FHE cryptosystem that uses Gentry's bootstrapping technique can be classified in the category of noise-based fully homomorphic encryption schemes [2]. If this class of cryptosystems has the advantage to be robust and more secure, it has the drawback to be not efficient in terms of runtime and ciphertext size. In several works followed Gentry's one, many techniques of noise management are invented to improve runtime efficiency and to minimise ciphertext and key size's (bootstrapping [8], key switching, modulus switching [3], re-linearization [4], flattening [10]), but the problematic of designing a practical and efficient fully homomorphic encryption scheme remains the same until now.

In the literature, we can come up with a second category of fully homomorphic encryption schemes called noise-free based [2], which do not need a technique of noise management to refresh ciphertexts. In a noise-free fully homomorphic encryption scheme, one can do infinity of operations on the same ciphertext without noise growing. This class of encryption schemes is known to be faster than the previous one, it involves simple operations to evaluate circuits on ciphertexts and do not require a noise management technique. However, it suffers from security problems, because the majority of designed schemes are cryptanalyzed today.

In this work, we will adopt the noise- free approach to design a new and efficient fully homomorphic encryption

scheme. We will try to overcome the problem of weak security through using the ring of quaternions and introducing a new method of coding integers in the domain of quaternions.

We propose a new noise-free fully homomorphic encryption scheme that uses the ring of Lipschitz's quaternions and permits computations on data encrypted under a symmetric key; a new method of coding integers (clear text) to Lipschitz's integers and a new approach to keep constant the free noise for any ciphertext after any operation. We present also an implementation of our fully homomorphic encryption scheme in JAVA programing language, the obtained results constitute a concrete proof and an effective demonstration to the performances of our scheme.

Our Techniques and Results: We propose a new noise-free fully homomorphic encryption scheme that uses the ring of Lipschitz's quaternions and permits computations on data encrypted under a symmetric key; a new method of coding integers (clear text) into Lipschitz quaternions and a new approach to keep constant the free noise for any ciphertext and after any operation. We present also an implementation of our results in JAVA programing language.

2 Mathematical Background

2.1 Quaternionique Field $\mathbb{H}$

A quaternion is a number in his generalized sense. Quaternions encompass real and complex numbers in a number system where multiplication is no longer a commutative law.

The Irish mathematician William Rowan Hamilton introduced the quaternions in 1843. They now find applications in mathematics, physics, computer science and engineering.

Mathematically, the set of quaternions $\mathbb{H}$ is a non-commutative associative algebra on the field of real numbers $\mathbb{R}$ generated by three elements i, j and k satisfying relations: $i^2 = j^2 = k^2 = i.j.k = -1$. Concretely, any quaternion q is written uniquely in the form: $q = a + bi + cj + dk$ where a, b, c and d are real numbers.

The operations of addition and multiplication by a real scalar are trivially done term to term, whereas the multiplication between two quaternions is termed by respecting the non-commutativity and the rules proper to i,j and k. For example, given $q = a + bi + cj + dk$ and $q' = a'+b'i+c'j+d'k$ we have $qq' = a_0+b_0i+c_0j+d_0k$ such that: $a_0 = aa' - (bb' + cc' + dd')$, $b_0 = ab' + a'b + cd' - c'd$, $c_0 = ac' - bd' + ca' + db'$ and $d_0 = ad' + bc' - cb' + a'd$.

The quaternion $\bar{q} = a - bi - cj - dk$ is the conjugate of $q.|q| = \sqrt{(q\bar{q})} = \sqrt{(a^2 + b^2 + c^2 + d^2)}$ is the module of q. The real part of q is $\Re(q) = (q + \bar{q})/2 = a$ and the imaginary part is $\Im(q) = (q - \bar{q})/2 = bi + cj + dk$.

A quaternion q is invertible if and only if its modulus is non-zero, and we have $q^{-1} = 1/|q|^2\bar{q}$.

2.2 Reduced Form of a Quaternion

Quaternion can be represented in a more economical way, which considerably alleviates the calculations and highlights interesting results. Indeed, it is easy to see that $\mathbb{H}$ is a $\mathbb{R}$-vectorial space of dimension 4, of which $(1, i, j, k)$ constitutes a direct orthonormal basis. We can thus separate the real component of the pure components, and we have for $q \in \mathbb{H}$,$q = (a, u)$ such that u is a vector of $\mathbb{R}^3$. So for $q = (a, u), q' = (a', v) \in \mathbb{H}$ and $\lambda \in \mathbb{R}$ we obtain:

1) $q + q' = (a + a', u + v)$ and $\lambda q = (\lambda a, \lambda u)$;
2) $qq' = (aa' - u.v, av + a'u + u \wedge v)$ Where $\wedge$ is the cross product of $\mathbb{R}^3$;
3) $\bar{q} = (a, -u)$ and $|q|^2 = a^2 + u^2$.

2.3 Ring of Lipschitz Integers

The set of quaternions defined as follows: $\mathbb{H}(\mathbb{Z}) = q = a + bi + cj + dk/a, b, c, d \in \mathbb{Z}$ Has a ring structure called the ring of Lipschitz integers. $\mathbb{H}(\mathbb{Z})$ is trivially non-commutative.

For r $n \in \mathbb{N}^*$, the set of quaternions: $\mathbb{H}(\mathbb{Z}/\mathrm{n}\mathbb{Z}) = \{q = a + bi + cj + dk/a, b, c, d \in \mathbb{Z}/\mathrm{n}\mathbb{Z}\}$ has the structure of a non-commutative ring.

A modular quaternion of Lipschitz $q \in \mathbb{H}(\mathbb{Z}/\mathrm{n}\mathbb{Z})$ is invertible if and only if its module and the integer n are coprime numbers, i.e $|q|^2 \wedge n = 1$.

2.4 Quaternionique Matrices $\mathbb{M}_2(\mathbb{H}(\mathbb{Z}/n\mathbb{Z}))$

The set of matrices $\mathbb{M}_2(\mathbb{H}(\mathbb{Z}/n\mathbb{Z}))$ describes the matrices with four inputs (two rows and two columns) which are quaternions of $\mathbb{H}(\mathbb{Z}/\mathrm{n}\mathbb{Z})$. This set has a non-commutative ring structure.

There are two ways of multiplying the quaternion matrices: the Hamiltonian product, which respects the order of the factors, and the octonionique product, which does not respect it.

The Hamiltonian product is defined as for all matrices with coefficients in a ring (not necessarily commutative). For example:

$$U = \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix}, V = \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{bmatrix}$$

$$\Rightarrow UV = \begin{bmatrix} u_{11}v_{11} + u_{12}v_{21} & u_{11}v_{12} + u_{12}v_{22} \\ u_{21}v_{11} + u_{22}v_{21} & u_{21}v_{12} + u_{22}v_{22} \end{bmatrix}$$

The octonionique product does not respect the order of the factors: on the main diagonal, there is commutativity of the second products and on the second diagonal there is commutativity of the first products.

$$U = \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix}, V = \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{bmatrix}$$

$$\Rightarrow UV = \begin{bmatrix} u_{11}v_{11} + v_{21}u_{12} & v_{12}u_{11} + u_{12}v_{22} \\ v_{11}u_{21} + u_{22}v_{21} & u_{21}v_{12} + v_{22}u_{22} \end{bmatrix}$$

In our article we will adopt the Hamiltonian product as an operation of multiplication of the quaternionique matrices.

2.5 Shur Complement and Inversibility of Quaternionique Matrices

Let $\mathcal{R}$ be an arbitrary associative ring, a matrix $M \in \mathcal{R}^{n\times n}$ is supposed to be invertible if $\exists N \in \mathcal{R}^{n\times n}$ such that $MN = NM = I_n$ where N is necessarily unique.

The Schur complement method is a very powerful tool for calculating inverse of matrices in rings. Let $M \in \mathcal{R}^{n\times n}$ be a matrix per block satisfying:

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \text{ such that } A \in \mathcal{R}^{k\times k}.$$

Suppose that A is invertible, we have:

$$M = \begin{bmatrix} I_k & 0 \\ CA^{-1} & I_{n-k} \end{bmatrix} \begin{bmatrix} A & 0 \\ 0 & A_s \end{bmatrix} \begin{bmatrix} I_k & A^{-1}B \\ 0 & I_{n-k} \end{bmatrix}$$

where $A_s = D - CA^{-1}B$ is the Schur complement of A in M.

The inversibility of A ensures that the matrix M is invertible if and only if A_s is invertible. The inverse of M is:

$$M^{-1} = \begin{bmatrix} I_k & -A^{-1}B \\ 0 & I_{n-k} \end{bmatrix} \begin{bmatrix} A^{-1} & 0 \\ 0 & A_s^{-1} \end{bmatrix} \begin{bmatrix} I_k & 0 \\ -CA^{-1} & I_{n-k} \end{bmatrix}$$

$$= \begin{bmatrix} A^{-1} + A^{-1}BA_s^{-1}CA^{-1} & -A^{-1}BA_s^{-1} \\ -A_s^{-1}CA^{-1} & A_s^{-1} \end{bmatrix}$$

For a quaternionique matrix:

$$M = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \in \mathcal{R}^{2\times 2} = \mathbb{M}_2(\mathbb{H}(\mathbb{Z}/n\mathbb{Z})).$$

where the quaternion a is invertible as well as its Schur complement $a_s = d - ca^{-1}b$ we have M is invertible and:

$$M^{-1} = \begin{bmatrix} a^{-1} + a^{-1}ba_s^{-1}ca^{-1} & -a^{-1}ba_s^{-1} \\ -a_s^{-1}ca^{-1} & a_s^{-1} \end{bmatrix}$$

Therefore, to generate an invertible quaternionique matrix randomly, it is sufficient:

- To choose randomly three quaternions a,b and c for which a is invertible.
- To select randomly the fourth quaternion d such that the Schur complement $a_s = d - ca^{-1}b$ of a in M is invertible.

3 Related Work

Generally, a fully homomorphic encryption scheme is defined as a quadruplet of algorithms $(Gen, Enc, Dec, Eval)$, which can be executed in a polynomial time, such as:

- $Gen(\lambda)$: Is a key generation algorithm, inputs a security parameter λ and outputs a pair of keys (sk, pk).
- $Enc(m, pk)$: Is an encryption algorithm, it takes as input a clear message m and a public key pk and outputs a ciphertext c.
- $Dec(c, sk)$: Is a decryption algorithm, takes as input a ciphertext c and a secret key sk and outputs the clear message.
- $Eval(C, c_1, .., c_n)$: Is an evaluation algorithm, takes as input a circuit C and ciphertexts $c_1, .., c_n$ and verifies $Dec(Eval(C, c_1, .., c_n), sk) = C(m_1, .., m_n)$.

After resisting roughly three decades, Rivest *et al.* conjecture was finally resolved in 2009 by Craig Gentry [8]. Indeed, Gentry gave a renaissance to the search for homomorphic cryptography by designing a fully homomorphic encryption scheme considered semantically secure. Gentry's design can be summarized into three main stages:

- Somewhat Homomorphic Encryption Scheme (SWHE): Gentry starts from a SWHE or simply homomorphic scheme that supports a limited number of homomorphic multiplications.
- Squashing the decryption circuit: Gentry reduces the complexity of the decryption circuit by publishing a set of vectors whose sum of a part of them is equal to the secret key. This so-called 'squash' scheme can evaluate, in addition to its SWHE capabilities, a NAND gate.
- Bootstrapping: The procedure of the bootstrap invented by Gentry consists in the evaluation of the circuit of decryption plus the NAND gate to obtain a so-called 'leveled' FHE which allows evaluating any circuit with a depth of the circuit defined at the beginning.

This first scheme is based on the addition of noise to clear to obtain the homomorphy of the cryptosystem. The major disadvantage of noise based approach is the growth of noise after each manipulation of the ciphertext (addition and/or multiplication). Indeed, in order to maintain the decryption capacity, it is necessary to control and reduce the noise generated after each treatment. The control of noise in this type of schemes increases their spatial and temporal complexity, which results in a slow calculation (especially during bootstrapping) and a greediness of the memory space required for storing the results (noise amplification). Therefore, this situation influences the application of fully homomorphic encryption to our

daily life. All these causes have encouraged researchers to find other frameworks for designing efficient fully homomorphic encryption.

Among the most eminent attempts to simplify fully homomorphic encryption schemes is the MORE cryptosystem [12]. It is a symmetric cryptosystem based on modular arithmetic whose homomorphy is derived from the usual matrix operations. Multiplication and addition are matrix multiplication and addition. In the MORE encryption scheme, the clear space is the ring $\mathbb{Z}/n\mathbb{Z}$ (ring of residual integers modulo n) where n is a modulo chosen as in the famous RSA algorithm, whereas the ciphertext space is the ring of the modular matrices $K \in \mathbb{M}_2(\mathbb{Z}/n\mathbb{Z})$. The secret key of this cryptosystem is an invertible matrix $K \in \mathbb{M}_2(\mathbb{Z}/n\mathbb{Z})$ chosen randomly by the client and kept confidential with its inverse K^{-1}.

However, the MORE cryptosystem does not support the IND-CPA (Indistinguishability under Chosen Plaintext Attack) and IND-KPA (Indistinguishability under Known Plaintext Attack) attacks. Indeed, if a third party in bad faith has access to a single clear and its ciphertext it will be able to decrypt any encrypted message thereafter without having found the secret key. The cryptosystem MORE has been cryptanalyzed several times [1, 16].

A second attempt, to overcome the security flaws of the MORE encryption scheme and to build a secure fully homomorphic encryption scheme, is recently due to Wang and Li [13]. The two authors retained almost the same conception of MORE except that they proposed to change the ring $\mathbb{Z}/n\mathbb{Z}$ by a non-commutative ring R and they used square matrices of order 3 instead of square matrices of order 2. Despite the use of a non-commutative ring R, clear messages always remain numbers that commute with the elements of R.

Therefore an attack on the Wang and Li scheme is given by Kristian Gjsteen and Martin Strand in [11]. Indeed, according to these authors: to attack the cryptosystem of Wang-Li, we only need to distinguish the encryptions of 0 from a random encryption.

The two authors observed that the diagonal of the ciphertext matrix completely determines the inversibility of the matrix, because an encryption of "0" cannot be inverted. Thus, with a high probability, we can distinguish the non-zero elements of the ring R from the zero elements. If the ring R is divisible, then there are no other non-zero elements than " 0 ". Finally, using a variant of the LU decomposition adapted to the non-commutative rings, we can efficiently calculate the secret key matrix of the scheme.

From what has come before, it can be pointed out that there are two types of fully homomorphic encryption scheme constructions:

A noise-based construction that uses the bootstrapping technique as described in Gentry's framework. The advantage of this construction is its robust security, since the schemes designed so far (based on this approach) are based on mathematical problems arising from the theory of Euclidean lattices, which remains an immune and complex theory. While the major disadvantage of this construction lies in the slowness of its operations (especially the bootstrapping step) and the complexity of its algorithms.

A noise-free construction that uses matrix operations as described in the MORE framework. This construction has the advantage of being very simple, easy to implement and provides very fast operations for any processing on ciphertexts. The main disadvantage of this construction lies in the security of the schemes designed so far. The schemes based on the MORE framework were subject to IND-CPA and IND-KPA attacks.

A first objective of the present encryption scheme is to improve the runtime in fully homomorphic encryption. For that reason we will adopt the MORE framework as the basis of construction instead of the Gentry's one which requires a very slow bootstrapping step. Our second objective is to overcome the dramatic problem of security in previous cryptosystems. We propose a more secure cryptosystem than its predecessors do and resistant to IND-CPA and IND-KPA attacks. Finally, we aim to ensure that our cryptosystem is fully homomorphic, that is to say it allows executing any type of processing on encrypted data. Therefore, the choice of a well-adapted clear space is paramount to concretize the entire homomorphy of our cryptosystem. We intend to use the ring $\mathbb{Z}/N^2\mathbb{Z}$, sanctioned by the two operations $\times$ and+, as clear text space for our encryption scheme. In addition to this, we use a homomorphic transform that converts an integer into a quaternion of Lipschitz. This makes it possible to randomize integers to ensure that the diagonal gives no useful information about the clear (avoid the attack of the cryptosystem of Li-Wang).

Our cryptosystem is resistant to IND-CPA and IND-KPA attacks by the non-commutativity of the ring of the Lipschitz quaternions and by the use of a randomized transform. It inherits its homomorphy, on the one hand from the matrix operations and on the other hand from a new homomorphic transform, between the ring $\mathbb{Z}/N^2\mathbb{Z}$ and the ring of the Lipschitz integers. Its complete homomorphy is obtained by manipulating these Lipschitz integers using a homomorphic transform $intToQuatern$.

4 Homomorphic Transform $intToQuatern$

Any integer $\sigma \in \mathbb{Z}/N^2\mathbb{Z}$ can be encoded into a Lipschitz quaternion according to a homomorphic transform whose operations on the quaternions retain those on the integers. This transform can be given as follows: intToQuatern: $\sigma \in \mathbb{Z}/N^2\mathbb{Z} \longmapsto intToQuatern(\sigma) = m+\alpha Ni+\beta Nj+\gamma Nk \in \mathbb{H}(\mathbb{Z})$ such that $\alpha, \beta, \gamma \in \mathbb{Z}/N\mathbb{Z}$ are randomly chosen integers. The inverse transform that will be named $quaternToInt$ is given by: $quaternToInt(q) = Re(q) mod N^2$.

It is easy to verify the homomorphism of the intToQuatern transform:

- For the addition operation we have clearly: $intToQuatern(\sigma) + intToQuatern(\sigma^{'}) = intToQuatern(\sigma + \sigma^{'} mod N^2)$.

- For the multiplication operation, by passing to the reduced notation of the quaternions, we obtain $intToQuatern(\sigma) \times intToQuatern(\sigma^{'}) = (m, u) \times (m^{'}, v) = (mm^{'} - u.v, mv + m^{'}u + u \wedge v)$,so we can easily verify that $mm^{'} - u.v \equiv (\sigma \times \sigma^{'}) mod N^2$ and $thatmv + m^{'}u + u \wedge v$ can be put on the form $(2L, P, Q)$ such that $P \equiv Q[2]$ and L is an integer. So $intToQuatern(\sigma) \times intToQuatern(\sigma^{'}) = intToQuatern(\sigma \times \sigma^{'} mod N^2)$.

The homomorphic transform *intToQuatern* encode and randomize an input integer. The homomorphic property allows us to preserve operations from integers to Lipschitz quaternions. The non commutativity of multiplication give two results for the same product of two encoded integers (i.e $intToQuatern(\sigma) \times intToQuatern(\sigma^{'}) = intToQuatern(\sigma \times \sigma^{'})$ and $intToQuatern(\sigma^{'}) \times intToQuatern(\sigma) = bitToQuatern(\sigma AND \sigma^{'})$ but $intToQuatern(\sigma^{'}) \times intToQuatern(\sigma) \neq intToQuatern(\sigma) \times intToQuatern(\sigma^{'})$). The inverse transform *quaternToInt* permits to find the encoded integer from a Lipschitz quaternion.

5 An Efficient Fully Homomorphic Encryption Scheme

We place ourselves in a context where Bob wants to store confidential data in a very powerful but non-confident cloud. Bob will later need to execute complex processing on his data, of which he does not have the necessary computing powers to perform it. At this level he thinks for, at first, the encryption of his sensitive data to avoid any fraudulent action. But the ordinary encryption, which he knows, does not allow the cloud to process his calculation requests without having decrypted the data stored beforehand, which impairs their confidentiality. Bob asks if there is a convenient and efficient type of encryption to process his data without revealing it to the cloud. The answer to Bob's question is favorable, in fact since 2009 there exist so-called fully homomorphic encryption, the principle of which is quite simple: doing computations on encrypted data without thinking of any previous decryption.

To be completely homomorphic, it is sufficient for a cryptosystem to perform the two operations of addition and multiplication a multitude of times on ciphertexts. Since their first appearance in 2009, fully homomorphic encryption schemes allow to easily realize the additions whereas the multiplication remains very expensive in term of runtime and exhausting in terms of the noise growth. Actually, on average, an addition doubles the noise of an encrypted message while a multiplication raises it to the square.

In order to profitably benefit from the technological advance of the cloud and to outsource its heavy calculations comfortably, Bob needs a robust highly secure fully homomorphic encryption scheme whose operations of addition and multiplication are done in a judicious time and whose noise generated during a treatment is manageable.

To help Bob take full advantage of the powers of the cloud, we introduce a probabilistic symmetric fully homomorphic encryption scheme without noise. The addition and multiplication operations generate no noise. The multiplication is very fast and it is done in less than a millisecond. The security of our cryptosystem is based on the difficulty of solving a system of multi-varied equations in a non-commutative ring.

5.1 Key Generation

- Bob generates randomly two big prime numbers p and q.

- Then, he calculates $N = p.q$.

- Bob generates randomly an invertible matrix

$$K = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} \in \mathbb{M}_3(\mathbb{H}(\mathbb{Z}/N^2\mathbb{Z}))$$

- Bob calculates the inverse of K, Which will be denoted K^{-1}).

- The secrete key is (K, K^{-1}).

5.2 Encryption

Lets $\sigma \in \mathbb{Z}/N^2\mathbb{Z}$ be a clear text. To encrypt σ Bob proceed as follows:

- Using the transform *intToQuatern* , Bob transforms σ into a quaternion: $m = intToQuatern(\sigma) \in \mathbb{H}(\mathbb{Z}/N^2\mathbb{Z}))$.

- Bob generates a matrix

$$M = \begin{bmatrix} m & r_3 & r_4 \\ 0 & r_1 & r_5 \\ 0 & 0 & r_2 \end{bmatrix} \in \mathbb{M}_3(\mathbb{H}(\mathbb{Z}/N^2\mathbb{Z}))$$

  such that $r_i \in \mathbb{H}(\mathbb{Z}/N\mathbb{Z}) \forall i \in [1, 5]$ are randomly generated with $|r_1| \equiv 0[N]$.

- The ciphertext of σ is $C = Enc(\sigma) = KMK^{-1} \in \mathbb{M}_3(\mathbb{H}(\mathbb{Z}/N^2\mathbb{Z}))$.

5.3 Decryption

Lets $C \in \mathbb{M}_3(\mathbb{H}(\mathbb{Z}/N^2\mathbb{Z}))$ be a ciphertext. To decrypt C Bob proceeds as follows:

- He calculates $M = K^{-1}CK$ using his secrete key (K, K^{-1}).

- Then he takes the first input of the resulting matrix $m = (M)_{1,1}$.
- Finally, he recovers his clear message by calculating $\sigma = quaternToInt(m)$ using the $quaternToInt$ transform.

5.4 Addition and Multiplication

Let σ_1 and σ_2 be two clear texts and $C_1 = Enc(\sigma_1)$ and $C_2 = Enc(\sigma_2)$ be their ciphertexts respectively. It is easy to verify, thanks to the $intToQuatern$ transform, that:

- $C_{add} = C_1 + C_2 = Enc(\sigma_1) + Enc(\sigma_2) = Enc(\sigma_1 + \sigma_2 mod N^2)$.
- $C_{mult} = C_1.C_2 = Enc(\sigma_1).Enc(\sigma_2) = Enc(\sigma_1 \times \sigma_2 mod N^2)$.

6 Comparison with Other Schemes

As it is shown in Table 1, our cryptosystem presents good performances compared to other existing schemes. Its ciphertext and key sizes depend linearly to cleartext space dimension. The other schemes use a small cleartext space which influences the runtime of the algorithm. In our case we are using a large cleartext space which allows us to encrypt big messages and perform computations directly on ciphertexts. We can observe that the complexity of Li-Wang's scheme is smaller than ours, but this scheme uses a smaller cleartext space.

7 Security

Ciphertext indistinguishability is an important security property of many encryption schemes. Intuitively, if a cryptosystem possesses the property of indistinguishability, then an adversary will be unable to distinguish pairs of ciphertexts based on the message they encrypt. It is easy to see that a fully homomorphic encryption scheme cannot be secure against adaptive chosen ciphertext attacks ($IND - CCA2$).

The adversary: We are protecting ourselves from an adversary A, who:

- Is a probabilistic polynomial time Turing machine.
- Has all the algorithms.
- Has full access to communication media.

Chosen Ciphertext Attack: In this model, the attack assumes that the adversary A has access to an encryption oracle and that the adversary can choose an arbitrary number of plaintexts to be encrypted and obtain the corresponding ciphertexts. In addition, the adversary A gains access to a decryption oracle, which decrypts arbitrary ciphertexts at the adversary's request, returning the plaintext.

Startup:

1) The challenger generates a secret key Sk based on some security parameter k (e.g., a key size in bits) and retains it.
2) The adversary A may ask the encryption oracle for any number of encryptions, calls to the decryption oracle based on arbitrary ciphertexts, or other operations.
3) Eventually, the adversary A submits two distinct chosen plaintexts m_0, m_1 to the challenger.

The Challenge:

1) The challenger selects a bit $b \in \{0, 1\}$ uniformly at random, and sends the "challenge" ciphertext $C = Enc(Sk, m_b)$ back to the adversary. The adversary is free to perform any number of additional computations or encryptions.
2) In the non-adaptive case ($IND - CCA$), the adversary may not make further calls to the decryption oracle before guessing.
3) In the adaptive case ($IND - CCA2$), the adversary may make further calls to the decryption oracle, but may not submit the challenge ciphertext C.
4) In the end it will guess the value of b.

The Result:

- Again, the adversary A wins the game if it guesses the bit b.
- A cryptosystem is indistinguishable under chosen ciphertext attack if no adversary can win the above game with probability p greater than $1/2 + \epsilon$ where is a negligible function in the security parameter k.
- If $p > 1/2$ then the difference $p - 1/2$ is the advantage of the given adversary in distinguishing the ciphertext.

In our situation, the adversary A should distinguish an encryption of zero from an encryption of one after asking the encryption oracle of a number of encryptions and the decryption oracle to decrypt arbitrary ciphertexts. The adversary A can do operations on the two given ciphertexts to distinguish zero from one, as he can do operations on the entire ciphertext matrices or just to use some entrees (the diagonal of ciphertexts matrices). In our case, even if the diagonal of M determines completely the invertibility of C, an encryption of an integer $\sigma \in \mathbb{Z}/N^2\mathbb{Z}$ is always non invertible because of the choice of the random $r_1(|r_1| \equiv 0[N])$. Therefore, an adversary cannot then

Table 1: Comparison of the performances of FHE schemes

Algorithm	Cleartext Space	Secret Key	Public Key	Ciphertext
Gentry [8]	$\{0,1\}$	n^7	n^3	$n^{1.5}$
Smart-Vercautern [15]	$\{0,1\}$	$O(n^3)$	n^3	$O(n^{1.5})$
DGHV [17]	$\{0,1\}$	$\tilde{O}(\lambda^{10})$	$\tilde{O}(\lambda^2)$	$\tilde{O}(\lambda^5)$
CMNT [7]	$\{0,1\}$	$\tilde{O}(\lambda^7)$	$\tilde{O}(\lambda^2)$	$\tilde{O}(\lambda^5)$
Batch DGHV [5]	$\{0,1\}^l$	$\tilde{O}(\lambda^7)$	$l.\tilde{O}(\lambda^2)$	$l.\tilde{O}(\lambda^5)$
Li-Wang [13]	$\mathbb{Z}/N\mathbb{Z}$	$O(N)$	NA	$O(N)$
Our scheme	$\mathbb{Z}/N^2\mathbb{Z}$	$O(N^2)$	NA	$O(N^2)$

distinguish encryptions of units from encryptions of non-units. Consequently, the attack proposed on Li-Wang's scheme [13] in [11] do not work for our case. Based on these assumptions, we believe that our fully homomorphic encryption scheme is indistinguishable under chosen ciphertext attacks ($IND-CCA1$).

Concerning the security of the secret key:

Given a random secret key of our encryption scheme:

$$K = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix}$$

and

$$K^{-1} = \begin{bmatrix} a^-_{1,1} & a^-_{1,2} & a^-_{1,3} \\ a^-_{2,1} & a^-_{2,2} & a^-_{2,3} \\ a^-_{3,1} & a^-_{3,2} & a^-_{3,3} \end{bmatrix}$$

and a cleartext $\sigma \in \mathbb{Z}/N^2\mathbb{Z}$.

A ciphertext of $m = intToQuatern(\sigma)$ is determined by:

$$C = KMK^{-1} = \begin{bmatrix} c_{1,1} & c_{1,2} & c_{1,3} \\ c_{2,1} & c_{2,2} & c_{2,3} \\ c_{3,1} & c_{3,2} & c_{3,3} \end{bmatrix}$$

such that

$$M = \begin{bmatrix} m & r_3 & r_4 \\ 0 & r_1 & r_5 \\ 0 & 0 & r_2 \end{bmatrix}$$

Therefore, we obtain the nine following equations:

1) $c_{1,1} = a_{1,1}ma^-_{1,1} + (a_{1,1}r_3 + a_{1,2}r_1)a^-_{2,1} + (a_{1,1}r_4 + a_{1,2}r_5 + a_{1,3}r_2)a^-_{3,1}$

2) $c_{1,2} = a_{1,1}ma^-_{1,2} + (a_{1,1}r_3 + a_{1,2}r_1)a^-_{2,2} + (a_{1,1}r_4 + a_{1,2}r_5 + a_{1,3}r_2)a^-_{3,2}$

3) $c_{1,3} = a_{1,1}ma^-_{1,3} + (a_{1,1}r_3 + a_{1,2}r_1)a^-_{2,3} + (a_{1,1}r_4 + a_{1,2}r_5 + a_{1,3}r_2)a^-_{3,3}$

4) $c_{2,1} = a_{2,1}ma^-_{1,1} + (a_{2,1}r_3 + a_{2,2}r_1)a^-_{2,1} + (a_{2,1}r_4 + a_{2,2}r_5 + a_{2,3}r_2)a^-_{3,1}$

5) $c_{2,2} = a_{2,1}ma^-_{1,2} + (a_{2,1}r_3 + a_{2,2}r_1)a^-_{2,2} + (a_{2,1}r_4 + a_{2,2}r_5 + a_{2,3}r_2)a^-_{3,2}$

6) $c_{2,3} = a_{2,1}ma^-_{1,3} + (a_{2,1}r_3 + a_{2,2}r_1)a^-_{2,3} + (a_{2,1}r_4 + a_{2,2}r_5 + a_{2,3}r_2)a^-_{3,3}$

7) $c_{3,1} = a_{3,1}ma^-_{1,1} + (a_{3,1}r_3 + a_{3,2}r_1)a^-_{2,1} + (a_{3,1}r_4 + a_{3,2}r_5 + a_{3,3}r_2)a^-_{3,1}$

8) $c_{3,2} = a_{3,1}ma^-_{1,2} + (a_{3,1}r_3 + a_{3,2}r_1)a^-_{2,2} + (a_{3,1}r_4 + a_{3,2}r_5 + a_{3,3}r_2)a^-_{3,2}$

9) $c_{3,3} = a_{3,1}ma^-_{1,3} + (a_{3,1}r_3 + a_{3,2}r_1)a^-_{2,3} + (a_{3,1}r_4 + a_{3,2}r_5 + a_{3,3}r_2)a^-_{3,3}$

According to the decryption algorithm, the plaintext m can be obtained by the equation:

(*) $m = (a^-_{1,1}c_{1,1} + a^-_{1,2}c_{2,1} + a^-_{1,3}c_{3,1})a_{1,1} + (a^-_{1,1}c_{1,2} + a^-_{1,2}c_{2,2} + a^-_{1,3}c_{3,2})a_{2,1} + (a^-_{1,1}c_{1,3} + a^-_{1,2}c_2, 3 + a^-_{1,3}c_{3,3})a_{3,1}$

An adversary who possesses the ciphertext C and wants to find the cleartext m or the secret key from the above nine equations should, at least, extract the secret components $a^-_{1,1}, a^-_{1,2}, a^-_{1,3}, a_{1,1}, a_{2,1}$ and $a_{3,1}$ according to the equation (*). Since our fully homomorphic encryption scheme is probabilistic, these nine equations are randomly independent even if the encrypted messages are the same one. Therefore finding the secret key is equivalent to a problem of solving an over-defined system of quadratic multivariate polynomial equations in a non-commutative ring.

8 Implementation and Test

We provide an implementation of our fully homomorphic encryption scheme with the fully homomorphic capability, i.e. we implement the key generation, encryption, decryption, add and mult operations.

The implementation is done using a personal computer with characteristics: bi-cores Intel core i5 CPU running at 2.40 GHz, with 512KB L2 cache and 4GB of Random Access Memory. The present implementation is done under JAVA programming language using the IDE Eclipse platform.

The fundamental results of our tests are summarized in Table 1, for the security parameter n that we used to generate the secret key. In this table we summarize the main parameters of our fully homomorphic encryption scheme.

Table 2: Comparison of the performances of FHE schemes

Security param	Key Gen	Encryption	Decryption	Addition	Multiplication	Secret Key	Ciphertext
256 bit	0.12s	0.02s	0.003 s	$\ll$ 1ms	$\ll$ 1ms	2.25KB	1.125KB
512 bit	0.31s	0.04s	0.004s	$\ll$ 1ms	1ms	4.5KB	2.25KB
1024 bit	1.2s	0.19s	0.01s	$\ll$ 1ms	2ms	9KB	4.5 KB
2048 bit	10.16s	1.76s	0.034s	$\ll$ 1ms	10ms	18KB	9KB
4096 bit	130s	20s	0.1s	$\ll$ 1ms	27ms	36KB	18KB

In one hand, we observe that, even if encryption and decryption operations are approximately the same, the runtime of encryption operation is significantly higher than the runtime of the decryption operation. This excessive difference between the two operations is due to the intToQuatern transform, we note that the most of the encryption time is spent in transforming an integer to a quaternion of Lipschitz. Concerning the evaluation operations, we observe that addition is always done in less than one millisecond and multiplication is done in an optimized time. This is adequate in view of the fact that matrix operations are simples. Therefore, these runtimes are practical in the context of a cloud that has unlimited computation powers.

In the other hand, we note that the secret key size is of the order of some few Kbytes for a given security parameter n. Moreover, the ciphertext size is about half the secret key size. This is because the secret key consists of two matrices but the ciphertext is just one matrix. All ciphertext sizes are fixe owing to the fact that we are using a noise free fully homomorphic encryption scheme.

9 Conclusion

In this article, we presented a new fully homomorphic encryption scheme. It is symmetric, noise free and probabilistic cryptosystem, for which the ciphertext space is a non-commutative ring quaternionic based. We utilize a homomorphic transform to encode an integer into a quaternion before its encryption. Our encryption scheme find its applications in the domain of cloud computing and big data security. It is an efficient and practical scheme whose security is based on the problem of solving an over-defined system of quadratic multivariate polynomial equations in a non-commutative ring. We have provided an implementation and simulation of our algorithm using JAVA programming language and a personal computer Core i5 CPU running at 2.40 GHz, with 512KB L2 cache and 4GB of Random Access Memory. The experimental results justifies the efficiency of our construction.

References

[1] E. Y. Ahmed and M. D. Elkettani, "Cryptanalysis of fully homomorphic encryption schemes," *International Journal of COmputer Science and Information Security*, vol. 14, no. 5, 2016.

[2] E. Y. Ahmed and M. D. Elkettani, "Fully homomorphic encryption: State of art and comparison," *International Journal of COmputer Science and Information Security*, vol. 14, no. 4, 2016.

[3] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) Fully homomorphic encryption without bootstrapping," *ACM Transactions on Computation Theory (TOCT'14)*, vol. 6, no. 3, pp. 13, 2014.

[4] Z. Brakerski and V. Vaikuntanathan, "Efficient fully homomorphic encryption from (standard) LWE," *SIAM Journal on Computing*, vol. 43, no. 2, pp. 831–871, 2014.

[5] J. H. Cheon, J. S. Coron, J. Kim, M. S. Lee, T. Lepoint, M. Tibouchi, and A. Yun, "Batch fully homomorphic encryption over the integers," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 315–335, 2013.

[6] G. Chunsheng, "Fully homomorphic encryption based on approximate matrix GCD," *Aavailable at ePrint*, 2011. (`https://eprint.iacr.org/2011/645.pdf`)

[7] J. S. Coron, A. Mandal, D. Naccache, and M. Tibouchi, "Fully homomorphic encryption over the integers with shorter public keys," in *Annual Cryptology Conference*, pp. 487–504, 2011.

[8] C. Gentry and D. Boneh, *A fully homomorphic encryption scheme*, Doctoral Dissertation, 2009. ISBN: 978-1-109-44450-6.

[9] C. Gentry, S. Halevi, and N. P. Smart, "Homomorphic evaluation of the aes circuit," in *Advances in Cryptology*, pp. 850–867, 2012.

[10] C. Gentry, A. Sahai, and B. Waters, "Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based," in *Advances in Cryptology*, pp. 75–92, 2013.

[11] K. Gjøsteen and M. Strand, "Can there be efficient and natural fhe schemes," *IACR Cryptology ePrint Archive*, vol. 2016, pp. 105, 2016.

[12] A. Kipnis and E. Hibshoosh, "Efficient methods for practical fully homomorphic symmetric-key encrypton, randomization and verification," *IACR Cryptology ePrint Archive*, vol. 2012, pp. 637, 2012.

[13] J. Li and L. Wang, "Noise-free symmetric fully homomorphic encryption based on noncommutative rings," *IACR Cryptology ePrint Archive*, vol. 2015, pp. 641, 2015.

[14] R. L. Rivest, L. Adleman, and M. L. Dertouzos, "On data banks and privacy homomorphisms," *Foundations of secure computation*, vol. 4, no. 11, pp. 169–180, 1978.

[15] N. P. Smart and F. Vercauteren, "Fully homomorphic encryption with relatively small key and ciphertext sizes," in *International Workshop on Public Key Cryptography*, pp. 420–443, 2010.

[16] B. Tsaban and N. Lifshitz, "Cryptanalysis of the more symmetric key fully homomorphic encryption scheme," *Journal of Mathematical Cryptology*, vol. 9, no. 2, pp. 75–78, 2015.

[17] V. Vaikuntanathan, C. Gentry, S. Halevi, and M. V. Dijk , "Fully homomorphic encryption over the integers," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 24–43, 2010.

Biography

Ahmed El-Yahyaoui is a PhD student at Mohammed V University in Rabat, Morocco. His research area is about cryptography and computer science security. In his PhD, he is working on a topical subject which is "Fully homomorphic encryption". He received an engineering degree in telecommunications and information technologies in 2013 from the National Institute of Postes and Telecommunications (INPT) in Morocco.

Mohamed Dafir Ech-Cherif El Kettani is a professor of computer science and information security at ENSIAS in Morocco. His research area is about information security and multicast routing. He obtained an engineering degree from Mohamedia School of Engineering in 1994 and a PhD degree in computer science in 2001 from the same school.