Can Hybrid Homomorphic Encryption Schemes Be Practical ?

Khalid El Makkaoui*
LAVETE laboratory
FST, Univ Hassan 1, B.P. : 577
26000 Settat, Morocco
kh.elmakkaoui@gmail.com

Abderrahim Beni-Hssane
LAROSERI laboratory, Department of Computer Science
Sciences Faculty, Chouaïb Doukkali University
El Jadida, Morocco
abenihssane@yahoo.fr

Abdellah Ezzati
LAVETE laboratory
FST, Univ Hassan 1, B.P. :577
26000 Settat, Morocco
abdezzati@gmail.com

***Abstract*—The ability to perform computations on ciphertexts without knowing any information about the plaintexts makes homomorphic encryption technique useful in a wide variety of confidentiality preserving protocols (e.g., e-voting, e-health, etc.). Unfortunately, the traditional encyption schemes support a limited number of homomorphic operations (usually addition or multiplication). Indeed, in 2009, Gentry proposed a fully homomorphic encryption scheme which supports both multiplicative and additive homomorphic operations. Since then, several fully homomorphic encryption schemes have been proposed. However, the fully homomorphic encryption schemes are still undergoing experimentation and improvement. The hybridization of homomorphic encryption schemes seems to be an effective way to overcome their limitations and to benefit from their resistance against the confidentiality attacks. In this paper, we will study the possibility to hybridize the homomorphic encryption schemes so as to support all homomorphic properties.**

Keywords— *homomorphic encryption; fully homomorphic encryption; hybrid homomorphic encryption; additive homomorphism; multiplicative homomorphism; third party.*

INTRODUCTION

Recently, the majority of companies think to outsource their IT infrastructure into a third party (e,g,. Cloud Provider) so as to reduce data storage and management costs, and to benifit from advantages offred by the selected third party such as: powerful computations, etc. However, the concerns over the sensitive data confidentiality is become a major obstacle to outsourcing data adoption. Regarding, the data storage confidentiality service, data can be encrypted before sending them to the third party, using one of the most efficient secret-key encryption schemes. But, to give a third party the ability to perform any treatments, these data must be decrypted, it is this step that can be considered as a breach of confidentiality. Thus, researchers stressed a useful technique called "Homomorphic Encryption (HE)".

Homomorphic encryption technique is a form of encryption that allows specific types of computations to be performed on encrypted data and generates an encrypted result. The decrypted result of any operation is the same such as working directly on plaintexts [1]. The ability to perform computations on ciphertexts without knowing any information about the plaintexts makes this technique useful in a wide variety confidentiality preserving protocols (e.g., e-voting, e-health, etc.). The homomorphic encryption schemes are numerous such as: the RSA [2], the ElGamal[3], the Paillier[4], etc. Unfortunately, these encryption schemes support a limited number of homomorphic operations. The RSA and ElGamal encyption schemes support only multiplicative homomorphism; whereas, the Paillier encryption scheme support additive homomorphism.

Indeed, in 2009, Gentry proposed for the first time a fully homomorphic encryption scheme (see [5] and [6]) which supports both multiplicative and additive homomorphic properties. Since then, many fully homomorphic encryption schemes have been proposed. However, the traditional encryption schemes with the homomorphic properties have been proven to be most robustness against the confidentiality attacks. The hybridization of these encryption schemes seems to be an effective way to address their limitations and to benefit from their resistance.

In this work, we will study the possibility to hybridize homomorphic encryption schemes in order to support all homomorphic properties as so as fully homomorphic encryption schemes.

The remainder of this paper is organized as follows: In Section II, we will give a formal definition of homomorphic encryption as well as some well-known encryption schemes. In Section II, we will provide a study over the possibility to hybridize the homomorphic encryption schemes. In Section IV, we will present our conclusions and future works.

HOMOMORPHIC ENCRYPTION

Homomorphic encryption technique is a form of encryption which allows specific types of computations to be performed on ciphertexts and generates an encrypted result. The decrypted results match the results of operations performed on raw data [1]. Among homomorphic encryption schemes, we distinguish two categories, depending on the operations performed on encrypted data: partially homomorphic encryption and fully homomorphic encryption. The partially homomorphic encryption schemes support a limited number of homomorphic properties. Whereas, the fully homomorphic encryption support all homomorphic properties. The fully homomorphic

978-1-5090-5146-5/16/$31.00 ©2016 IEEE

encryption category has a subcategory known as somewhat homomorphic encryption (SWHE). The encryption schemes of this subcategory support a limited number of multiplication or/and addition operations.

A. Partially Homomorphic Encryption

Partially homomorphic encryption (also known as homomorphic encryption) can be considered as a group homomorphism [1]. Let the five-tuple (P, C, K, E, D) be an encryption scheme (see [1] and [7]), where P is a finite set of possible plaintexts, C is a finite set of ciphertexts and K is a finite set of possible of keys. For each k ϵ K, there is an encryption rule $e_k \epsilon E$ and a corresponding decryption rule $d_k \epsilon$ D. Let us assume that the plaintexts form a group (P, $\circ$) and that the ciphertexts form a group (C, $\Diamond$), then e_k is a map from the group P to the group C and d_k is a map from the group C to the group P, i.e.,

$$e_k : P \rightarrow C$$

and

$$d_k : C \rightarrow P$$

$\forall$ a, b in $\boldsymbol{P}$, their corresponding ciphertexts c_a ,c_b in $\boldsymbol{C}$, and k in $\boldsymbol{K}$, if:

$$e_k (a \circ b) = e_k (a) \Diamond e_k (b)$$

and

$$d_k (c_a \Diamond c_b) = d_k (c_a) \circ d_k (c_a)$$

So, the encryption scheme is homomorphic.

B. Fully Homomorphic Encryption

Fully homomorphic encryption can be considered as a ring homomorphism [1]. Let the five-tuple (P, C, K, E, D) be an encryption scheme, where P is a finite set of possible plaintexts, C is a finite set of ciphertexts and E, D are the encryption and decryption algorithms. K is the key space (see [1]). For each $k \epsilon K$, there is an encryption rule $e_k \epsilon E$ and a corresponding decryption rule $d_k \epsilon D$. Let us assume that the plaintexts form a ring (P, $\oplus_P$, $\otimes_P$) and the ciphertexts form a ring (C, $\oplus_C$, $\otimes_C$), then e_k is a map from the ring P to the ring C and d_k is a map from the ring C to the ring P, i.e.,

$$e_k : P \rightarrow C$$

and

$$d_k : C \rightarrow P$$

For all a, b in $\boldsymbol{P}$, their corresponding ciphertexts c_a ,c_b in $\boldsymbol{C}$, and k in $\boldsymbol{K}$, if

$$e_k (a \oplus_P b) = e_k (a) \oplus_C e_k (b)$$

$$e_k (a \otimes_P b) = e_k (a) \otimes_C e_k (b)$$

and

$$d_k (c_a \oplus_C c_b) = d_k (c_a) \oplus_P d_k (c_a)$$

$$d_k (c_a \otimes_C c_b) = d_k (c_a) \otimes_P d_k (c_a)$$

Therefore, the encryption scheme is fully homomorphic.

C. Homomorphic encryption schemes

In this part, we will present three homomorphic encyption schemes, two of these schemes support only one homomorphic property; whereas, the last one supports both multiplicative and additive homomorphisms.

1) RSA Encryption Scheme:

In 1977, Ron Rivest, Adi Shamir, and Leonard Adleman proposed for the first time a practical public-key encryption scheme with a homomorphic property, named RSA [2]. The RSA encryption scheme is composed of key generation, encryption, and decryption algorithms as follows:

RSA Key Generation Algorithm

Input: Two large primes p and q randomly and independenly of each other.

- Compute $N = pq$ and $\phi(N) = (p-1)(q-1)$ where $\phi(N)$ is the Euler totient function.
- Choose randomly an integer e such that $1 \leq e < \phi(N)$ and $gcd(e, \phi(N)) = 1$.
- Determine the private exponent d the multiplicative inverse of the public exponent e (mod $\phi(N)$) such that $ed = 1 (mod\ \phi(N))$.

Output: (pk, sk): The public key is $\mathbf{pk} = (N, e)$ and the private key is $\mathbf{sk} = (N, d)$.

RSA Encryption Algorithm

Input: message m, where $m \epsilon Z_N$.

- Compute the ciphertext c as: $c = m^e\ (mod\ N)$

Output: $c = E(pk, m)$

RSA Decryption Algorithm

Input: ciphertext c, where $c \epsilon Z_N$.

- Recover the plaintext message as: $m = c^d\ (mod\ N)$

Output: $m = D(sk, c)$

Figure 1. RSA encryption scheme

2) Paillier Encryption Scheme:

The Paillier encryption scheme [4] invented in 1999 by Pascal Paillier, is a probabilistic public-key encryption scheme. The Paillier encryption scheme has the additive homomorphism and it is composed of key generation, encryption, and decryption algorithms as follows:

Paillier Key Generation Algorithm

Input: Two large primes p and q randomly and independenly of each other.

- Compute $N = pq$ and $\lambda = lcm(p-1, q-1)$
- Choose an interger g where $g \in Z^*_{N^2}$
- Ensure N divides the order of g, that means: $\mu = (L(g^{\lambda} \bmod N^2))^{-1} (\bmod N)$ with $L(u) = (u-1)(N)^{-1}$

Output: (pk, sk): The public key is **pk** = (N, g) and the private key is **sk** = (λ , μ).

Paillier Encryption Algorithm

Input: message m, where $m \in Z_N$.

- Choose $r \in Z^*_N$
- Compute $c = g^m . r^n (\bmod N^2)$

Output: $c = E(pk, m)$

Paillier Decryption Algorithm

Input: ciphertext c, where $c \in Z^*_{N^2}$

- Compute $m = L(c^{\lambda} \bmod N^2) . \mu (\bmod N)$

Output: $m = D(sk, c)$

Figure 2. Paillier encryption scheme

3) DGHV Encryption Scheme:

In 2009, M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan [8] proposed a second fully homomorphic encryption scheme which based on Gentry's scheme [5], but it does not require ideal lattices. The DGHV scheme provides both multiplicative and additive homomorphisms and it is composed of three algorithms as shown in Figure 3.

DGHV Key Generation Algorithm

Input: An odd integer $p \in [2^{\eta-1}, 2^{\eta}]$

Output: (sk): the private key is **sk** = (p).

DGHV Encryption Algorithm

Input: message $m \in \{0,1\}$

- Choose randomly q and r
- Compute $c = pq + 2r + m$

Output: $c = E(sk, m)$

DGHV Decryption Algorithm

Input: ciphertext c

- Compute $m = (c(\bmod p))(\bmod 2)$

Output: $m = D(sk, c)$

Figure 3. DGHV encryption scheme

HYBRIDIZATION OF HOMOMORPHIC ENCRYPTION SCHEMES

This section is divided into two main parties. The first party presents some algebraic structures definitions. The second half of the section consists of a study about the possibility to hybridize the homomorphic encryption schemes.

A. Algebraic Structures Definitions

In this part, we will give the definitions of some algebraic structures that we will need to perform our study.

1) ***Definition1*** [9]**:** In mathematics, a **group** ***G*** is a set of elements that equipped with one binary operation ($\circ$). (***G***, $\circ$) must satisfy the following properties:

1. $\forall$ a,b in ***G***, the result of the operation $a \circ b$ is also in ***G***.
2. $\forall$ a,b and c in ***G***, $(a \circ b) \circ c = a \circ (b \circ c)$ (i.e., $\circ$ is associative).
3. $\exists$ $e \in$ ***G*** (called the **identity element**), such that $\forall$ $a \in$ ***G***, $a \circ e = e \circ a = a$.
4. $\forall$ $a \in$ ***G***, $\exists$ $b \in$ ***G*** (called the **inverse** of a) such that $a \circ b = b \circ a = e$.

For all a and b in ***G***, if $a \circ b = b \circ a$ (i.e., $\circ$ is commutative), we say that (***G***, $\circ$) is commutative or abelian.

2) ***Definition2*** [9]***:*** A group homomorphism is a function between groups which preserves the algebraic structure. Let us give two groups, ***G*** together with an operation ($\circ$), and ***H*** together with an operation ($\lozenge$). A group homomorphism from (***G***, $\circ$) to (***H***, $\lozenge$) is a function

$$f : \boldsymbol{G} \rightarrow \boldsymbol{H}$$

such that

$$f(a \circ b) = f(a) \lozenge f(b)$$

for all a, b in ***G***.

3) ***Definition3*** [9]***:*** A ring ***R*** is a set of elements equipped with two binary operations, addition (+) and multiplication ($\times$), that satisfy the following properties (The symbols a, b, and c represent any elements from ***R***):

1. (***R***,+) is an abelian group.
2. $(a \times b) \times c = a \times (b \times c)$, ($\times$ is associative)
3. $a \times (b + c) = (a \times b) + (a \times c)$, $(b + c) \times a = (b \times a) + (c \times a)$ (i,e,. multiplication distributes over addition on the right and the left).

4) ***Definition4*** [9]***:*** A ring homomorphism is a function between two rings which preserves the algebraic structure. Let ***R*** and ***S*** be two rings, a ring homomorphism is a function

$$f : \boldsymbol{R} \rightarrow \boldsymbol{S}$$

such that

$$f(a + b) = f(a) + f(b) \text{ and } f(a \times b) = f(a) \times f(b)$$

for all a, b in ***R***.

B. *Hybridization of Homomorphic Encryption Schemes*

In this part, we will try to answer to the question of whether hybrid homomorphic encryption schemes be practical, in other words, the possibility to build a new encryption scheme which supports all homomorphic operations from the encryption schemes that supports a limited number of homomorphic operations (addition or multiplication).

Recall that the homomorphic encryption can be considered as a group homomorphism (see more the definition of group homomorphism in section III). And the fully homomorphic encryption can be considered as a ring homomorphism (see the definition of ring homomorphism in section III). The homomorphic encryption schemes can support one homomorphic property. Whereas, the fully homomorphic encryption schemes can support all homomorphic properties.

To develop a new homomorphic encryption scheme which supports all homomorphic operations from two homomorphic encryption schemes one supports only addition and other supports only multiplication operations. The hybrid homomorphic encryption scheme must preserve the algebraic structure.

Let the five-tuple (P, C_1, K_1, E_1, D_1) be an homomorphic encryption scheme which supports only multiplication (e.g., RSA encryption scheme [2]) and (P, C_2, K_2, E_2, D_2) be an other homomorphic encryption scheme which supports only addition (e.g., Paillier encryption scheme [4]), where P is a finite set of possible plaintexts, C_1 and C_2 are the finite set of ciphertexts, E_1 and E_2 are the encryption algorithms, D_1 and D_2 are the decryption algorithms, and K_1 and K_2 are the key spaces. Let us assume that the plaintexts form two groups $(P, \otimes_p)$ and $(P, \oplus_p)$, and the ciphertexts form also two groups $(C_1, \otimes_c)$ and $(C_2, \oplus_c)$. As the two encryption schemes are homomorphic; therefore, the functions follow form group homomorphisms:

$$E_1: (P, \otimes_p) \rightarrow (C_1, \otimes_c)$$
$$D_1: (C_1, \otimes_c) \rightarrow (P, \otimes_p)$$
$$E_2: (P, \oplus_p) \rightarrow (C_2, \oplus_c)$$
$$D_2: (C_2, \oplus_c) \rightarrow (P, \oplus_p)$$

From these two homomorphic encryption schemes, we can build an hybrid homomorphic encryption scheme that support all homomorphic properties, if and only if we can create two rings $(P, \oplus_p, \otimes_p)$ and $(C, \oplus_c, \otimes_c)$ from the groups $(P, \otimes_p)$, $(P, \oplus_p)$, $(C_1, \otimes_c)$ and $(C_2, \oplus_c)$, with the functions bellow form ring homomorphisms :

$$Hyb.E: (P, \oplus_p, \otimes_p) \rightarrow (C, \oplus_c, \otimes_c)$$
$$Hyb.D: (C, \oplus_c, \otimes_c) \rightarrow (P, \oplus_p, \otimes_p)$$

Where, *Hyb.E* and *Hyb.D* are the hybrid encryption and decryption algorithms.at

The structure of hybrid homomorphic encryption schemes will be composed of many algorithms as shown in Figure 4 (for instance). These algorithms are dependent on the basis homomorphic encryption schemes.

Hybrid Generation Algorithms

Input: E_1, E_2, D_1, D_2, K_1 and K_2

Output:

- Hybrid key $Hyb.K = f(K_1, K_2) = (pk_{hyb}, sk_{hyb})$
- Hybrid encryption algorithm : $Hyb.E = f(E_1, E_2)$
- Hybrid decryption algorithm : $Hyb.D = f(D_1, D_2)$

Hybrid Encryption Algorithm

Input: message m

- Compute $c = Hyb.E(pk_{hyb}, m)$

Output: c

Hybrid Decryption Algorithm

Input: ciphertext c

- Compute $c = Hyb.D(sk_{hyb}, c)$

Output: message m

Figure 4. hybrid homomorphic encryption scheme

Finally, an hybrid homomorphic encryption scheme that supports all homomorphic properties can be considered as a ring homomorphism; therefore, to develop a new hybrid homomorphic encryption scheme their algorithms (hybrid key generation, hybrid encryption and hybrid decryption algorithms) must preserve the algebraic structure of a ring homomorphism.

CONCLUSIONS AND FUTURE WORKS

In this paper, we have presented a formal definition of homomorphic and fully homomorphic encryption techniques as well as some well-known homomorphic encryption schemes. In answer to the question of whether hybrid homomorphic encryption schemes be practical, we have given the algebraic structure of an hybrid homomorphic encryption scheme.

In our future works, we will try to take RSA and Paillier encryption schemes as a basis to propose a new homomorphic encryption scheme that supports all homomorphic properties.

REFERENCES

[1] Xun Yi, Russell Paulet and Elisa Bertino, 'Homomorphic Encryption and Applications', 1st edn. Springer Briefs in Computer Science, Springer International Publishing, 2014.

[2] R. Rivest, A. Shamir, L. Adleman, ‘A method for obtaining digital signatures and public-key cryptosystems’. Commun. ACM 21 (2), 120–126, 1978.
[3] ElGamal, T. ‘A public key cryptosystem and a signature scheme based on discrete logarithms’, IEEE Transactions on Information Theory, pp.469–472, 1985.
[4] P. Paillier, ‘Public-key cryptosystems based on composite degree residuosity classes’, Advances in Cryptology EUROCRYPT’99, pp. 223–238, 1999.
[5] C. Gentry, ‘Fully homomorphic encryption using ideal lattices’, Proceedings of STOC’09, pp. 169-178, 2009.
[6] C. Gentry,Fully Homomorphic Encryption Using Ideal Lattices. PhD thesis, 2009.
[7] D. R. Stinson, Cryptography: Theory and Practice, 3rd edn. Discrete Mathematics and Its Applications, Chapman Hall, 2006.
[8] M. van Dijk, C. Gentry, S. Halevi, V. Vaikuntanathan, ‘Fully homomorphic encryption over the integers’, Proceedings of Advances in Cryptology, EUROCRYPT’10, pp. 24–43, 2010.
[9] Marlow Anderso and Todd Feil, ‘A First Course in Abstract Algebra: Rings, Groups, and Fields’, 3rd edn. CRC Press, Taylor & Francis Group, 2014.