International Journal of Information Security and Privacy
Volume 13 • Issue 2 • April-June 2019

Design of Public-Key Algorithms Based on Partial Homomorphic Encryptions

Marwan Majeed Nayyef, Anbar University, Baghdad, Iraq
Ali Makki Sagheer, University of Anbar, Baghdad, Iraq

ABSTRACT

With the rapid development of cloud computing, which has become a key aspect to maintain the security of user information that may be highly confidential and maintained during transport and storage process. The reliance on traditional algorithms that are used to encrypt data are not secure enough because we cannot process the data only after decrypt. In this article is proposed the use of homomorphic encryption to solve this problem because it can deal with encrypted data without the decryption, which can lead to ensuring confidentiality of the data. A number of public-key algorithms are explained, which is based on the concept of homomorphic encryption. In this article an algorithm is proposed based on HE and it is similar to Menesez-EC but with one digit as a secret key according to its advantage, whereby reducing the cost of communication, and storage and provides high processing speed when compared with other algorithms. This algorithm provides enough security for a bank's customer information and then compared with ECC, each of RSA and Piallier algorithms as evaluated.

KEYWORDS

Elliptic Curve Cryptography, Homomorphic Encryption, Partial Homomorphic Encryption, Public Key Cryptography

1. INTRODUCTION

Early on, many researcher studies began on homomorphic encryption more in-depth. Homomorphic encryption was simplified and through advances in research, most of the research appeared to focus their efforts toward homomorphic encryption due to its importance in more aspects spatially in the field of the cloud computing in order to provide information security and maintaining that information from penetrating by the hackers (Chen, Ben, & Huang, 2014). Homomorphic Encryption is an important kind of encryption in computational science, it provides many techniques such as partially, somewhat and fully homomorphic encryption with the purpose of the securely store, transfer and dealing with ciphertext in a way that maintains the confidentiality and integrity of the data (Ogburn, Turner, & Dahal, 2013). Homomorphic encryption can be classified into partially and fully homomorphic encryption, with partial Homomorphic Encryption (PHE) use one operation in ciphertext whereas Fully Homomorphic Encryption (FHE) can use all operations in the ciphertext, and it is one of the most common new topics which make more of the researcher to deal with those concepts because of providing more security for data especially in the cloud environment (Suveetha & Manju, 2016).

There are two main general cryptosystems they are symmetric and asymmetric cryptosystem. AES, DES are symmetric-key algorithm and Elgamal, paillier and RSA are asymmetric cryptosystem, in this paper, we work in the public key encryption algorithms.

DOI: 10.4018/IJISP.2019040105

In section 1, we would explain the concepts, functions and properties of homomorphic encryption. In section 2, Elliptic curve Cryptography is described, in section 3, describe encryption algorithms such as (RSA, Paillier, Elgamal, Goldwasser-Micali and Boneh-Goh-Nissim (BGN)) are based on homomorphic encryption properties. In section 4, we would explain the limitation of PHE. In section 5, we would describe comparison between different algorithms of homomorphic encryption that give a general idea of all the algorithms. In section 6, the proposed algorithm is described, in section 7, we explain the experimental result of the proposed algorithm and compare ECC with other algorithms.

2. BACKGROUND

In 2012 Li Li, Ahmed A.Abd, XiamuNiu proposed new scheme with additive homomorphism property based on ElGamal-Elliptic Curve (ElGamal-EC) for transferring secret images over a channel which is unsecured instead of using ElGamal and RSA scheme. In this paper, the proposed scheme uses a shorter key to better performance than schemes based on ElGamal or RSA. Therefore, decryption of images requires lower processing compared with the method that uses the other additively homomorphic property in ElGamal-EC. Experimental results and analysis show that the proposed method is faster and has superior performance for RSA and ElGamal (Li, Abd El-Latif, & Xiamu, 2012).

In 2015, Kamal Kumar Chauhan; Amit K.S. Sanger, A. Verma, a secure method was developed for keeping data, Data security is an important aspect, especially when data transfer and storage over the internet (cloud computing), therefore various methods of standard encryption algorithm provide security for data in storage and transmission. In the traditional state data to be processed must be decrypted first, but this state makes data understandable to a cloud provider. Standard encryption algorithms are not sufficient to make data more secure. In this paper various schemes are proposed such as (Pillar, RSA, and Boneh-Goh-Nissim (BGN)) based on homomorphic encryption in cloud computing in order to secure data through processing state because of Homomorphic encryption allows the service provider to operate on ciphertext without decryption. The implementation of these schemes helps to provide security for data stored in cloud computing (Chauhan, Sanger, & Verma, 2015).

In 2016, Tannishk Sharma creates a voting system in order to solve the problem of the time consuming, obstruction and disruption which may happen. The development of Information Technology led us to propose an E - voting system to solve all these problems, E-voting system helps us to vote from any place. In this paper, an E-voting system proposed based on Paillier Homomorphic Encryption scheme in order to provide security for those systems through processing and transferring data in ciphertext form. The E – Voting System was executed successfully and contributed to data security which transfers over the internet and also ensures efficiency, privacy, universal verifiability and no vote duplication (Sharma, 2016).

In 2016, Quan Hong, Zhao, Wang, secure environment schemes are proposed to solve the problem of Secure Multiparty Computation (SMC), through the storage and processing of data more secure against attacks and any penetrating, especially in public environments. To protect the privacy of the data, data must be stored and processed in encrypted form in the cloud computing without recovering the original text, which is done through the implementation of one of homomorphic encryption schemes, and this paper, they have proposed the use of Ėlliptic curved Cryptography based on the scheme of homomorphic encryption to solve SMC problem. This scheme has been implemented successfully, and get many benefits, including protection of privacy, the consumption of energy and communication consumption are compared with an algorithm of RSA encryption (Hong, 2016).

In 2017, Wenxiu Ding et al, exhibited another way for processing encrypted data using Homomorphic Encryption that is dealing with the ciphertext. HE limitation only allows person owns a homomorphic secret key to decrypt processed ciphertexts which do not allow for multiple users. In this paper, they propose a Homomorphic Re-Encryption Scheme (HRES) extended to multi-user to access processed ciphertexts. The proposed schemes are implemented to evaluate their performance and security (Wenxiu, Yan, et al., 2017).

3. HISTORY OF HOMOMORPHIC ENCRYPTION

Initially, partial homomorphic encryption uses only one mathematical operation in the ciphertext, in 1978 R. Rivest, L. Adleman and M. Dertouzos is the first researchers who proposed the concept of homomorphic applied to the RSA algorithm that was homomorphic on multiplicative operation. In progress in 1982 Goldwasser and Silvio suggested another algorithm called Goldwasser that is homomorphic on XOR which is applying a homomorphic encryption with safety remarkable level, in 1999, Paillier also suggested a security encryption system that was homomorphically with addition operation, as well as another algorithm such as Elgamal and Elgamal-Elliptic curve algorithms that are homomorphic on multiplication operation (Rivest, Adleman, & Michael, 1987; Benzekki, Fergougui, & El Alaoui, 2016; Yi, Paulet, & Bertino, 2014; Chauhan, Sanger, & Verma, 2015).

The important aspect is proposing a homomorphic encryption scheme depending on the addition and multiplication operation (Two operations). Dan Boneh, EU-Jin Goh and Kobbi Nissim in 2005 built first homomorphic encryption with two operations, which have unlimited number of additions and one multiplication. This way is depending on elliptic curves, the encryption method is adding two messages homomorphically by multiplying the two ciphertexts (Yang & Zhang, 2014; Coron, Naccache, & Tibouchi, 2012).

The first fully homomorphic encryption scheme was created in 2009 and produced by Gentry (2009), Gentry first generates a somewhat homomorphic scheme using ideal lattices which only support a certain number of operations performed on a ciphertext and according to the fact that the ciphertext has a limited number of "noise" that increases when multiplication operation increased which finally effected on the decryption results (Dasgupta & Pal, 2016), then it is bootstrapped to make it fully homomorphic. There are many issues used for reducing the problem of noise that is happening through encryption and decryption of data, so the noise increased with every multiplying and add to the encrypted result (Ogburn, Turner, & Dahal, 2013).

Because of complex computationally, Gentry and Halevi have implemented Gentry's scheme over ideal lattices. In (2010) the authors Van Dijk et al offer fully homomorphic encryption scheme (DGHV) over integer instead of ideal lattices, but it has a key with large size (Dasgupta & Pal, 2016).

Craig Gentry and Shai Halevi at (2011) develop an approach that contained a mixture of (somewhat HE) SWHE and other encryption types called multiplicatively homomorphic encryption (MHE) (Ogburn, Turner, & Dahal, 2013).

In 2013 Gupta and Sharma create a scheme which, depending on symmetric keys with smaller size that based on operations, including matrix computation such as matrix inversion this can lead to making computationally less expensive, in 2014 Sharma also proposed a new scheme that works with one bit; this scheme is generalized by Aggarwal et al. (2016).

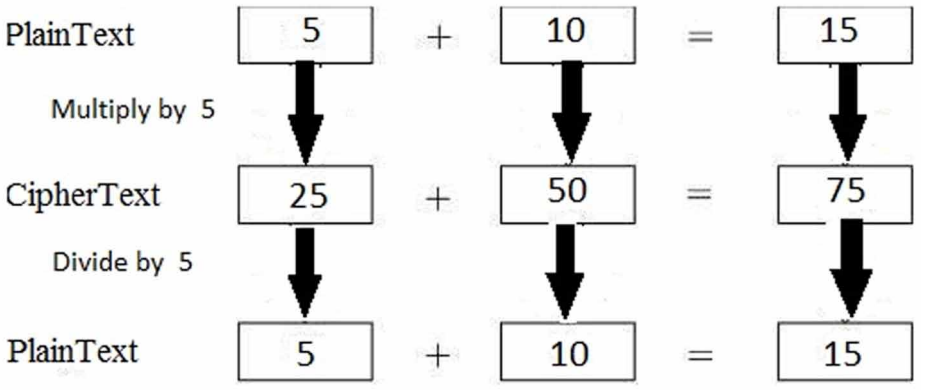
3.1. Concept of Homomorphic Encryption

It allows persons to use a specific mathematical operation applied to the ciphertext to getting results to be the same results if the same operation has been applied to the original text. This concept is shown in the following Figure 1. When the user needs to add two numbers such as 5 and 10, the result is 15, the two numbers are encrypted through multiplied with 5, then the sum of the encrypted number is 75 as a result that is stored on the cloud server, the user download data from cloud and recovered the original text (Chauhan, Sanger, & Verma, 2015).

3.2. Homomorphic Encryption Functions

In our proposal, we are dealing with asymmetric encryption which consists of two keys, one for encryption and another for decryption, so it is necessary to find a procedure used to create these keys according to the nature of algorithm's work, so there are four algorithms or (primitives) of a public key encryption schemes are KeyGen, Enc, and Dec, and an additional Eval:

Figure 1. Homomorphic encryption



1. **Key Generation Function:** It is an algorithm in a client which gets security parameter(k) to generate each of the secret key (sk) and public key(pk), (pk, sk)← KeyGen (k);
2. **Encrypt Function:** Is a random algorithm that produces a ciphertext (c) which came from using plaintext and sk, c← Enc (sk,m);
3. **Evaluation Functions:** The server uses function *f* for evaluating the ciphertext, and it's done by using f and pk, Eval (*f, pk,*c), where c= (c_1, . ., c_t) and t refer to the number of inputs of the circuit (Chen, Ben, & Huang, 2014; Gentry, 2009). Therefore Dec (sk, Eval (*f*, pk, c)) = C (m_1, m_2 ……m_t), Where C is a computation which performs in the client;
4. **Decryption Function:** Is a random algorithm that produces a plaintext (m) which came from ciphertext and sk m← Dec (*c*, sk), and after evaluation, we get the original text as follows Dec (sk, Eval (*f*, pk, c)).

4. HOMOMORPHIC ENCRYPTION PROPERTIES

Suppose that:

$m_1, m_2 \in M$ and c_1 and $c2 \in C$ then $m_1 = Dec(c_1)$ and $m_2 = Dec(c_2)$

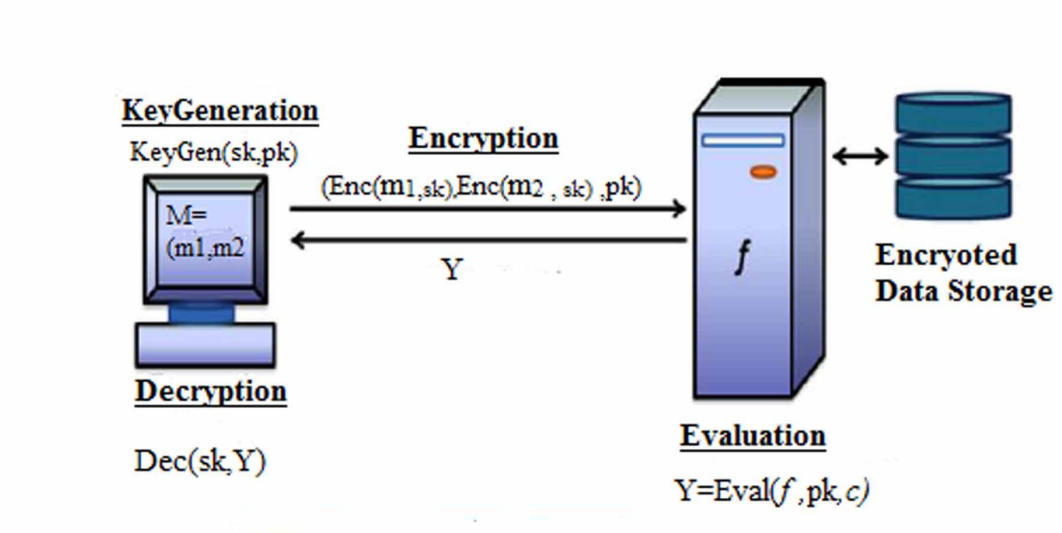
1. Additive Homomorphic Encryption:

$m_1 + m_2 \bmod n = Dec(c_1 + c_2 \bmod n)$

2. Multiplicative Homomorphic Encryption:

$m_1 * m_2 \bmod n = Dec(c_1 * c_2 \bmod n)$ (Filho, Silva, & Miceli, 2016)

Figure 2. Homomorphic encryption functions



4.1. Elliptic Curve Cryptosystem (ECC)

Elliptic curve cryptography is an approach of public-key cryptography, which is based on the structure of algebraic and discrete logarithms of elliptic curve over finite fields F_p. Let us say that q is a prime number, and an elliptic curve EC over a prime field F_p is shown in the following equation EC:

$$y^2 \pmod p = x^3 + ax^2 + b \pmod p$$

where (a and b) $\in F_p$ and satisfies the equation:

$$4a^3 + 27b^2 \neq 0 \pmod p$$

ECC security depends on the Discrete Logarithm Problem (ECDLP) and provides the same security level compared with RSA but with small key size, therefore the use of ECC with Homomorphic encryption is faster and more efficient than the use of RSA, Elgamal etc, (2015).

4.2. Algorithms of Partial Homomorphic Encryption

RSA cryptosystem: R. LRivest, and A. Shamir and L. Adelman (RSA) is the first algorithm that works with the property of HE, which is considered high security in the field of homomorphic encryption (Yi, Paulet, & Bertino, 2014). It is a partial Homomorphic Encryption scheme that operates only with the multiplication operation (Gerasimov, Epishkina, & Kogos, 2017; Sharma, 2016).

5. KEY GENERATION

1. Select a random two prime number which is p and q;
2. Compute N = p * q and φ (N) = (p - 1) * (q - 1);
3. An integer number e is selected which 1 < e < φ(N) if it satisfies the property gcd (e, φ(N)) =1;

4. Compute integer d, which ($1 < d < \varphi(N)$) that is a multiplicative inverse of e if it satisfies $e*d \equiv 1 \pmod{\varphi(N)}$;
5. Finally, public key and private key are generated corresponding to pk= (N, e) and sk = (p, q, d).
   Encryption: The message (M) is converted to ASCII which is an integer number ($M \in Zn$). M is encrypted to get a ciphertext C based on pk(N, e):

$$C = M^e \bmod N$$

Decryption: Recover the message M= ($m_1 \ldots m_n$) from ciphertext C= ($c_1 \ldots\ldots c_n$) by using sk (p, q, d):

$$m = c^d \bmod N$$
$$m = (m^e)^d \bmod N$$
$$m = m^{e\text{-}d=1} \bmod N$$

5.1. Multiplicative Homomorphic Encryption Property of RSA

We said that RSA cryptosystem is homomorphically with multiplicative property:

If $Enc(m_1 * m_2) = Enc(m_1) * Enc(m_2)$

So assume that c_1 and c_2 two ciphertext:

$$c_1 = m_1^e \bmod N$$
$$c_2 = m_2^e \bmod N$$

then:

$c_1 * c_2 = (m_1 * m_2)^e \bmod N$ [6,18]

5.2. ElGamal Cryptosystem

Elgamal and Taher (Elgamal) are another well-known homomorphic encryption scheme whose key exchanges depending on Diffie-Hellman. The effective security of the algorithm based on computing discrete logarithms (Coron, Naccache, & Tibouchi, 2012). According to the Diffie-Hellman algorithm, we can generate two large prime numbers (p, g) which g (generator) is primitive of the root. We suppose that Alice and Bob need to exchange our key for encryption and decryption the message (Dawahdeh, Yaakob, & Sagheer, 2015).

5.3. Key Generation

1. Generate an integer number which represents the secret (private) key of Alice and $1 < a < p - 1$;
2. Compute Alic's public-key (pka) = $g^a \bmod p$;
3. Public key (pk) = {p, g, pk_a}.

Any user wants to send his message to another user, his message must encrypt with the receiver's public key which shows in the following:

1. The massage M represents as an integer number where $0 <= M <= p - 1$;
2. Generate an integer number b which represents the secret (private) key of Bob and $1 < b < p - 1$;
3. Compute Shared Secret-key (SSK) = $pk_a^{\ b} \bmod p$, $pk_a = g^a \rightarrow SSK = (g^a)^{\ b} \bmod p$;
4. Compute $pk_b = g^b \bmod p$, pk_b is a Bob's public-key which sent with cipher text to the Alice for decryption.

Encryption: Message M is encrypted as follows:

$c_2 = SSK * M \bmod p$

Pair (pk_b, c_2) are sent to the receiver as an encrypted message.

Decryption: When Alice wants to decrypt the encrypted message (pk_b, c2) with her secret key a, Alice must compute the following:

1. Compute the $SSK = pk_b^{\ a} \bmod p$, $pk_b = g^b \rightarrow SSK = (g^b)^{\ a} \bmod p$ which SSK must be equal to Bob's SSK (Shared Key);
2. We compute the inverse number of SSK (SSK^{-1});

Finally recovered plaintext from ciphertext as:

$M = c_2 * SSK^{-1}$ (Stallings, 2011; Brakerski & Vaikuntanathan, 2011)

5.4. Multiplicative Homomorphic Encryption Property of ElGamal

When two messages m_1 and m_2 are encrypted to two ciphertext (pk_b, c_2), so the homomorphic property of Elgamal is shown as:

$Enc(m_1 * m_2 \bmod p) = Enc(m_1) * Enc(m_2) \bmod p$
$Enc(m_1, pk) = (g^{b1}, m_1 * pk_a^{\ b1})$ which $pk_a = g^{a1} \bmod p$
$Enc(m_2, pk) = (g^{b2}, m_2 * pk_a^{\ b2})$ and $pk_a = g^{a2} \bmod p$
$Enc(m_1, pk) * Enc(m_2, pk) = ((g^{b1} * g^{b2}), (m_1 * h^{b1} * m_2 * h^{b2}))$
$Enc(m_1, pk) * Enc(m_2, pk) = (g^{b1+b2}), (m_1 * m_2) * h^{b1+b2})$ (Gerasimov, Epishkina, & Kogos, 2017)

Paillier cryptosystem: It is a public key cryptosystem that is represented as partial Homomorphic Encryption which operates only with an addition operation (Chauhan, Sanger, & Verma, 2015). It is suitable for some applications such as a bank, especially in the field of cloud computing.

Key Generation:

1. Generate a two random of prime numbers (p, q) which $gcd(p*q, (p-1)*(q-1)) = 1$;
2. Compute $n = p * q$;
3. $k(n) = LCM(p - 1, q - 1)$, which LCM means (the least common multiplier) and k (n) means Carmichael function;
4. Choose a random generator g which $g \in Zn^2$ and $gcd(g^k \bmod n^2, n) = 1$ (Chauhan, Sanger, & Verma, 2015);
5. Compute a multiplicative inverse (u) which respect to module n, $u = (L(g^k \bmod n^2))^{-1} \bmod n$ where $L(s) = (s-1)/n$ (Kocaba, 2016), So two of the keys are generated.

The Public key is pk (n, g).

The Secret key is sk (k, u) (Sharma, 2016).

Encryption:

1. We need to encrypt the message (m) which m ∈ Zn^2;
2. Select number r randomly;
3. Calculate ciphertext c by using the public key (n, g), so $c = g^m * r^n \bmod n^2$.

Decryption:

1. The message m is recovered from the ciphertext c by using secret key (k, u);
2. $m = L(c^k \bmod n^2) * k(n) \bmod n$.

Addition Homomorphic Encryption property of Paillier:

$Enc(m_1+m_2) = Enc(m_1) * Enc(m_2)$
$Enc(m_1, pk) = g^{m1} * r_1^n \bmod n^2$
$Enc(m_2, pk) = g^{m2} * r^{2n} \bmod n^2$
$Enc(m_1, pk) * Enc(m_2, pk) = (g^{m1} * r_1^n \bmod n^2) * (g^{m2} * r^{2n} \bmod n^2)$
$Enc(m_1, pk) * Enc(m_2, pk) = g^{m1+m2}(r_1 * r_2)^n \pmod{n^2}$
$Enc(m_1, pk) * Enc(m_2, pk) = Enc(m_1 + m_2, pk)$ (Sharma, 2016; Gerasimov, Epishkina, & Kogos, 2017)

6. GOLDWASSER-MICALI CRYPTOSYSTEM

Goldwasser and Micali (GM) are public key encryption and famous homomorphic encryption scheme which was developed by Goldwasser and Micali in 1982.This algorithm is characterized by being defined as the first scheme of public key encryption that is provably secure. However, its cryptosystem is not efficient, and ciphertext takes several hundred times greater than the Plaintext. GM is represented as partial Homomorphic Encryption which operate on XOR. The scheme relies on determining whether the value of a specific h is quadratic residue or not, quadratic residue shows as the follows (Goldwasser & Micali, 1982):

1. $x_p = h \bmod p$, $x_q = h \bmod q$;
2. If {\displaystyle x_{p}^{(p-1)/2}\equiv 1{\pmod {p}}} $x_p^{(p-1)/2} \equiv 1 \bmod p$ and $x_q^{(q-1)/2} \equiv 1 \bmod q$, then h is a quadratic residue mod N;
3. Quadratic non-residue x.

   Key Generation:

   1. Alice randomly and independently generate two large prime integer p and q;
   2. Compute N = p* q;
   3. Public key pk= (h, N). The secret key sk = (p, q).

   Encryption: Assume Bob decided to send a message M to Alice:

   1. Bob first converts the m as a string of bits $M = (m_1, ..., m_n)$;
   2. For each bit of M, Bob randomly generates a value r_i which satisfies the following $gcd(r_i, N) = 1$;
   3. Compute ciphertext C:

$c_i = r_i^2 * h^{mi} \bmod N$
ciphertext outputs $c = (c_1, ..., c_n)$

Decryption: For each value i, by using factorization (p, q), Alice computes if the value c_i is a quadratic residue or not; if true, $m_i = 0$, otherwise $m_i = 1$ message outputs $m = (m_1 ... m_n)$.

XOR Homomorphic Encryption property of GM

Goldwasser scheme has the homomorphic property on XOR:

$Enc(m_1 \oplus m_2) = Enc(m_1). E(m_2)$

where:

$Enc(m_1) = c_1 = r_1^2 h^{m1} \bmod N$, $Enc(m_2) = c_2 = r_2^2 * h^{m2} \bmod N$
$c_1 * c_2 = (r_1^2 * h^{m1}) (r_2^2 * h^{m2}) \bmod N$
$c_1 * c_2 = (r_1 * r_2)^2 (h^{m1 + m2}) \bmod N = Enc(m_1 \oplus m_2)$ (Suveetha & Manju, 2016; Stallings, 2011)

7. BONEH-GOH-NISSIM CRYPTOSYSTEM (BGN)

Dan Boneh, EU-Jin Goh and Kobbi Nissim in 2005 build a public-key cryptosystem with two operations and performs unlimited addition operation with one multiplication operation, this type of PHE is near to FHE, and the encryption method is adding two messages homomorpically by multiplying the two ciphertext (Yi, Paulet, & Bertino, 2014).

Key Generation:

1. Generate a random of two prime $p_1, p_2 \in Z$;
2. Compute $n = p_1 * p_2$;
3. Generate a two generator $g, u \in Z$, $h = u^{p2}$;
4. Public key pk= (n, g, h, e, G, G1), and secret key sk= (p_1).

(G, G1) refers to multiplicative group with order n and e, so $G * G1 \rightarrow G1$ is bilinear map.

Encryption: The message m is encrypted with a public key pk:

$C = g^m * h^r \bmod n$

Decryption: The ciphertext C and secret key are used to re-get the message m, $C^{p1} = (g^{p1})^m$, which m is getting depending on discrete algorithm of C^{p1} to (g^{p1}) as a base.

Addition Homomorphic Encryption property of BGN:

$c_1 = g^{m1} * h^{r1} \bmod n$
$c_2 = g^{m2} * h^{r2} \bmod n$
$c_1 * c_2 = g^{(m1+m2)} * h^{(r1+r2)} \bmod n$ (Chauhan, Sanger, & Verma, 2015)

8. ELGAMAL-ELLIPTIC CURVE CRYPTOSYSTEM (DAWAHDEH, YAAKOB, & SAGHEER, 2015; HONG, WANG, & ZHAO, 2016)

Key generation:

1. Choose publicly prime field F_p and an elliptic curve EC over F_p;
2. Randomly select a base point G ∈ *EC,* which *G* represents as a large subgroup of EC, and n is the order of EC points;
3. Bob randomly chooses secrete key d in range [2, n];
4. Computes $Q_B = d*G$;
5. Make Q_B as public-key and d as secret-key.

Encryption: When Alice wants to send the message M to Bob, she follows the following steps:

1. Mapping the message M as a point on EC;
2. She randomly chooses secrete key e in range [2, n];
3. Compute Q_A=e*G, where Q_A is a public-key of Alice;
4. Compute SSK= $e*Q_B$, (i.e. SSK=e (d*G)), where SSK (Shared Secret key);

Ciphertext is obtained by the following equation:

$$C = P_m + SSK$$

6. Then Transfer ciphertex and Alice public key (C, Q_A).

Decryption: On the other side, Bob receives the encrypted text for the purpose of turning it into the original text as follows:

1. Compute SSK = $d(Q_A) = d(e*G)$;

Subtract SSK from the ciphertext point C:

$$P_m = c_i - SSK$$

9. ADDITIVE HOMOMORPHIC ENCRYPTION OF ELGAMAL-ELLIPTIC CURVE

Let massage $M= \{m_1, m_2, m_3 \ldots\ldots m_n\}$, and P_m is an elliptic curve point derived from *m*. If we say that Elgamal-elliptic curve cryptosystem is homomorphically with additive property, it must satisfy the following (Hong, Wang, & Zhao, 2016):

$$\text{Enc}\,(P_{m1} + P_{m2\ldots\ldots}P_{mn}) \bmod p = \text{Enc}(P_{m1}) + \text{Enc}(P_{m2}) \ldots. \text{Enc}\,(P_{mn}) \bmod p$$

From Equation (10):

$$\text{Enc}\,(P_{m1} + P_{m2\ldots\ldots}P_{mn}) = (P_{m1} + SSK) + (P_{m2} + SSK) \ldots\ldots (P_{mn} + SSK)$$

$$\sum_{i=1}^{n} Enc(Pmi) = \sum_{i=1}^{n} Pmi + \sum_{i=1}^{n} \text{SSK}$$

$$\sum_{i=1}^{n} Enc(Pmi) = \sum_{i=1}^{n} (Pmi + SSK)$$

$$\sum_{i=1}^{n} (Pmi + SSK) = \sum_{i=1}^{n} (Pmi + SSK)$$

10. MENEZES ELLIPTIC CURVE (M-EC) CRYPTOSYSTEM

In this type of encryption, we do not need to convert the message to the points on the elliptic curved, where the encryption process immediately after the message conversion to ASCII code.

Key generation:

1. Choose publicly prime field Fq and an elliptic curve EC over Fq;
2. Randomly select a base point G ∈ *EC, which G* represents as *a large subgroup of EC*, and n is the order of EC points;
3. Bob randomly chooses secret key d in range [2, n];
4. *Computes* $Q_B = d*G$;
5. Make Q_B public-key and d secret-key.

Encryption: When Alice wants to send the message M to Bob, she follows the following steps:

1. The message M is represented as a point of two numbers which $M = (m_1, m_2)$;
2. She randomly chooses secret key e in range [2, n];
3. Compute $Q_A = e*G$, where Q_A is a public-key of Alice;
4. Compute $SSK = e*Q_b$, (i.e. $SSK = e(d*G)$), where (Shared secret key) $SSK = (k_1, k_2)$;

Ciphertext $C = (c_1, c_2)$ is obtained by the following equation:

$c_1 = m_1 * k_1 \bmod p$
$c_2 = m_2 * k_2 \bmod p$

Decryption: On the other side, Bob receives the encrypted text for the purpose of turning it into the original text as follows:

1. Compute $SSK = d(Q_A) = d(e*G)$;

Subtract SSK from the ciphertext point C:

$m_1 = c_1 * k_1^{-1} \bmod p$
$m_2 = c_2 * k_2^{-1} \bmod p$

Multiplicative Homomorphic Encryption of modifying Menezes-Elliptic Curve:

$Enc(m_1*m_2) = Enc(m_1) * Enc(m_2)$

where:

$Enc((m_1, sk), pk) = m_1 * k_1 \bmod p$ and $Enc((m_2, sk), pk) = m_2 . k_2 \bmod p$
$Enc(m_1*m_2) = (m_1*k_1 \bmod p)(m_2*k_2 \bmod p)$
$Enc(m_1*m_2) = m_1*m_2 (k_1*k_2) \bmod p$

11. COMPARISON OF SEVERAL SCHEMES OF HOMOMORPHIC ENCRYPTION

Table 1 presents schemes of partial homomorphic encryption with his homomorphic properties which can operate with either addition or multiplication operations (RSA cryptosystem, Goldwasser cryptosystem, Elgamal cryptosystem, Paillier cryptosystem) except BGN which operates with homomorphic addition and one homomorphic multiplication (Gerasimov, Epishkina, & Kogos, 2017).

12. THE PROPOSED SYSTEM

The author, proposed a secure system to preserves Bank information, so in this paper, we propose an algorithm work on the base of decimal big integer. The secret key of the proposed algorithm is generated based on elliptic curve cryptography, and used to encrypt all personal information in the bank because the sensitivity of the information, since no one is allowed to know its contents or even access to it. Therefore, an algorithm has been proposed to encrypt this information and be dealt with by using Homomorphic Encryption to reach the intended person for the purpose of dealing with private data in it as in Figure 3.

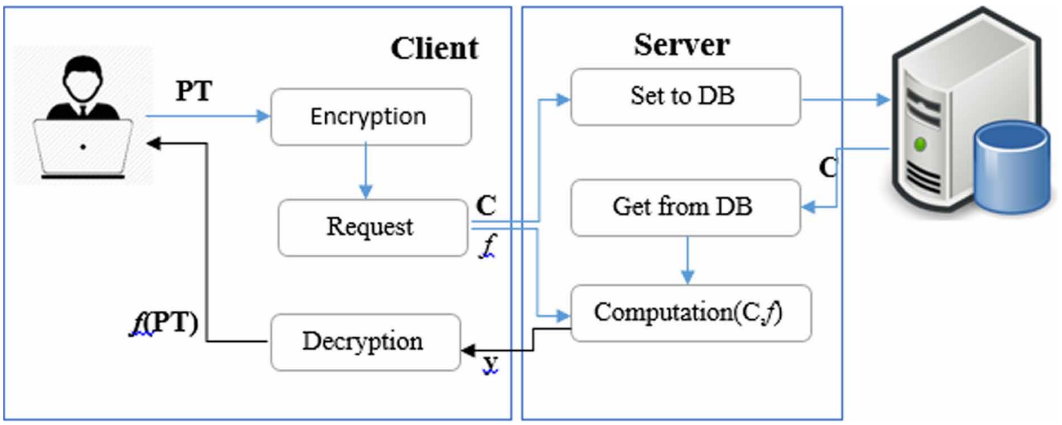
Steps of implement:

1. **Setup:** We depend NIST security parameter (SP) of Elliptic Curve with 160-bit to be used in key generation, so that SP = (B, a, b, p) where B is the Base point of ECC;
2. Key generation:
   a. Select random number d;
   b. Compute K, K=d*B = (k_1, k_2);
   c. Make secret key sk=k_1;

Table 1. (PHE) Partially Homomorphic Encryption

Encryption Scheme	Addition	Multiplication	Application
RSA (Gerasimov, Epishkina, & Kogos, 2017)	No	yes	Banking and secure internet, and credit card transaction
ElGamal (Kocaba, 2016)	No	yes	Hybrid systems
Paillier (Sharma, 2016)	yes	no	Threshold scheme, e-voting system
Goldwasser-Micali	XOR	no	Cloud computing
BGN (Chauhan, Sanger, & Verma, 2015)	yes	1	Security of integer polynomials.
Elgamal-Elliptic Curve (Goldwasser & Micali, 1982)	yes	no	Cloud Computing, E-voting
Menezes Elliptic curve (Sunuwar & Samal, 2015)	No	yes	Cloud Computing, E-voting

Figure 3. System model



3. Encryption:
   a. Convert plaintext PT= $(m_1 \ldots m_n)$ to the corresponding decimal number;
   b. $c_i = m_i * k_1 \bmod p$;
4. Decryption:
   a. Recover the Plaintext from Ciphertext (C);
   b. $m_i = c_i * k_1^{-1} \bmod p$;

Evaluation of Homomorphic Encryption:

$$c_1 * c_2 = Enc(m_1) * Enc(m_2)$$

where:

$$c_1 = m_1 * k_1 \bmod p$$
$$c_2 = m_2 * k_1 \bmod p$$

then:

$$c_1 * c_{2=} (m_1 * k_1) * (m_2 * k_1) \bmod p$$
$$c_1 * c_2 = (m_1 * m_2) * k_1^n \bmod p$$

There are two sides, client and server, the client uses the security parameter (SP) to generate secret and public key with 160-bit, the secret key known only by the client to be used for decrypt the plaint text. The server receives the encrypted data to be stored at the server storage. When there is a specific request from the client, the server retrieves the encrypted data from the server storage then performs computation on the cihpertext until we reach to result(y), the encrypted results are sent to the client for decryption and perform certain processes as in Figure 3.

13. EXPERIMENTAL RESULTS

13.1. The Implementation of Proposed Algorithm

Table 2 shows the encryption/decryption time of the proposed algorithm, ElGamal and RSA algorithm processes in millisecond using 160-bit as a key size. Figures 4 and 5 shows the encryption/decryption results plot of the Table 2.

The evaluation of Homomorphic Encryption represents the correct relationship between the original text and its encryption so that we obtain identical results in the case of evaluation of the original text and encryption text at the same time, Therefore, this implementation represents the time required to reach matching.

13.2. Comparison of Implementing Algorithms

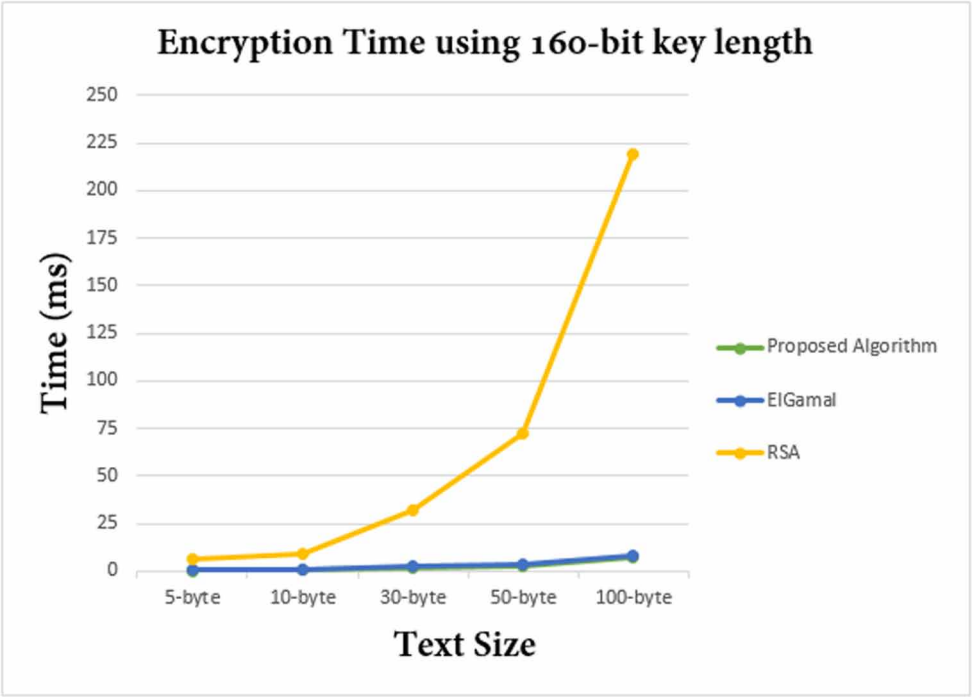
In this paper, several methods are used with a different security level, therefore the security is passed on the mechanism of ECC public-key in terms of storage, space and speed. Computation cost is obtained by the time it takes from encryption process. Communication cost is obtained from the exchange number of bytes on the communication channel.

An experiment is performed among RSA, Paillier and ECC homomorphic encryption algorithms using different message size (32, 64, 128, 256, 512, 1024) bytes for the purpose of computation cost, and the size of the key in that experience is 512-bit, where we notice that the ECC gives the best performance and results when compared with RSA and Paillier in terms of computation and communication cost are shown in Figures 7 and 8.

Table 2. Execution time of encryption and decryption text

Message Byte	Time in ms and 160-bit Key Length					
	Proposed Algorithm		ElGamal		RSA	
	Enc	Dec	Enc	Dec	Enc	Dec
5	0	1	1	3	6	10
10	1	1	1	4	9	21
30	2	4	3	13	32	53
50	3	9	4	29	72	119
100	7	16	8	55	219	388

Figure 4. Encryption time for the text using 160-bit key length



14. CONCLUSION

This paper explains the basic partial homomorphic encryption (PHE) concept and different asymmetric encryption algorithms depending on the properties of homomorphic encryption, and it is considered as an important key for people who wish to conduct their research in a specific cryptographic algorithm based on homomorphic encryption. This paper helps us to propose an algorithm that can be applied to the homomorphic encryption for the purpose of privacy preservation of the bank's customers, Therefore the proposed algorithm may be similar to Menezes-Elliptic Curve scheme, but with on key digit and also is evaluated with another scheme and get a better performance because it works as ECC, where it gives us the best efficiency compared with RSA and Paillier in terms of key size, less storage and speed. This paper also presents an overview of Partial Homomorphic Encryption.

Figure 5. Decryption time for the text using 160-bit key length

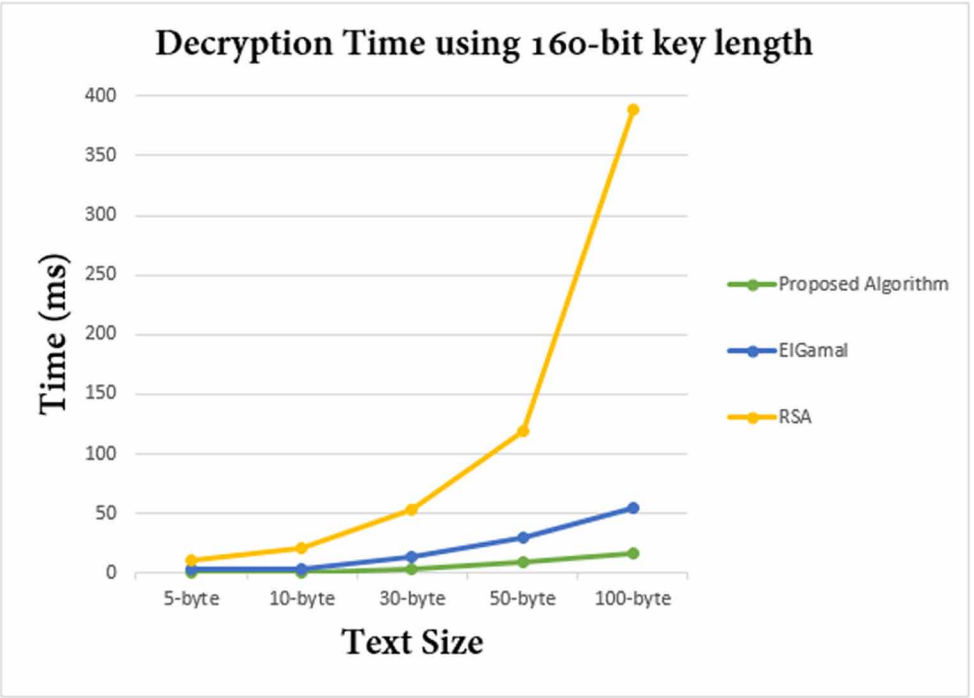


Table 3. Evaluation of proposed algorithm, ElGamal and RSA algorithm with key size 160-bit

No.	# Byte	Proposed Algorithm	ElGamal	RSA
1.	5	0	0	1
2.	10	0	1	2
3.	30	1	2	3
4.	50	1	2	4
5.	100	2	3	8

Figure 6. Evaluation time of homomorphic encryption using 160-bit key size

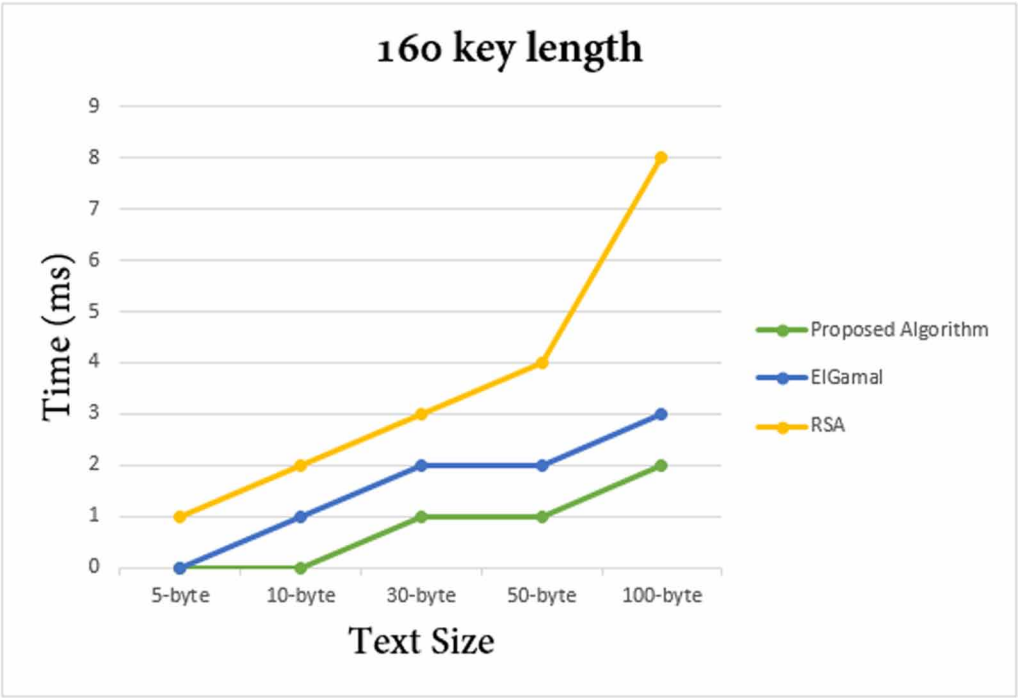


Figure 7. Comparison of ECC, RSA and Paillier in terms of computational cost

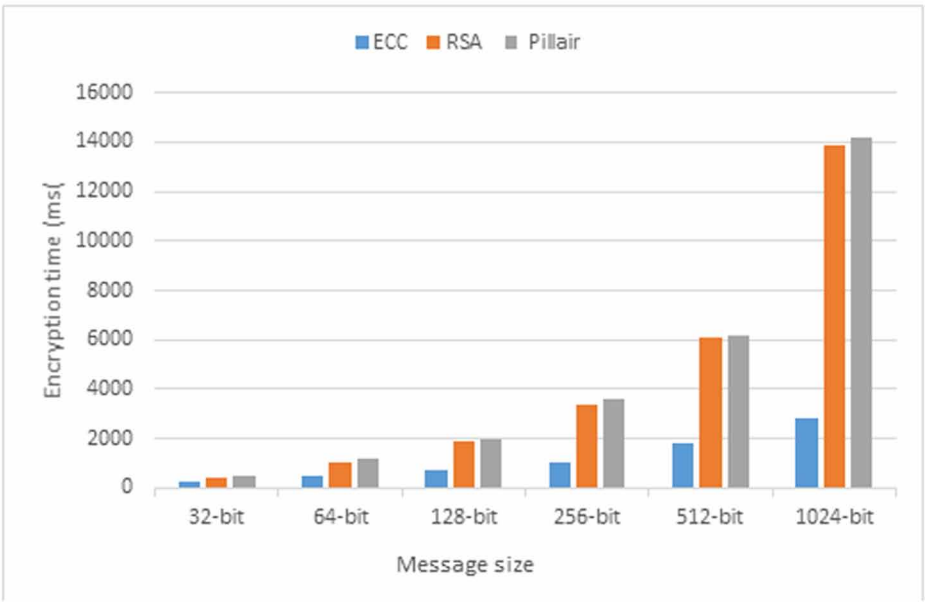
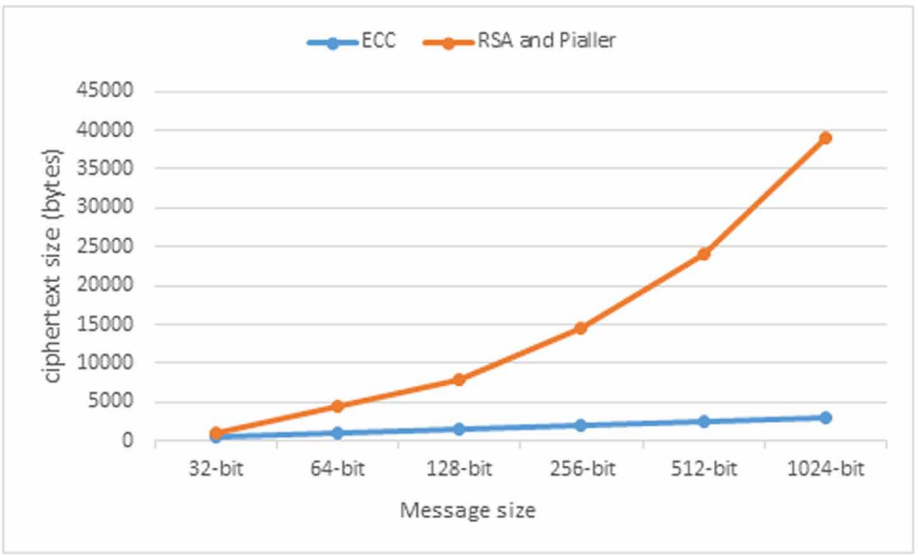


Figure 8. Comparison of communication cost between ECC and (RSA, Paillier)



REFERENCES

Benzekki, K., El Fergougui, A., & El Alaoui, A. E. (2016). A Secure Cloud Computing Architecture Using Homomorphic Encryption. *International Journal of Advanced Computer Science and Applications*, *7*(2). doi:10.14569/IJACSA.2016.070241

Brakerski, Z., & Vaikuntanathan, V. (2011). Efficient Fully Homomorphic Encryption from (Standard). *SIAM Journal on Computing*, *43*(2), 831–871. doi:10.1137/120868669

Chauhan, K. K., Sanger, A. K., & Verma, A. (2015, December). Homomorphic Encryption for Data Security in Cloud Computing. In *2015 International Conference on Information Technology (ICIT)* (pp. 206-209). IEEE.

Chen, L., Ben, B., & Huang, J. (2014). An Encryption Depth Optimization Scheme for Fully Homomorphic Encryption. In *International Conference on Identification* (pp. 137-141). doi:10.1109/IIKI.2014.35

Coron, J. S., Naccache, D., & Tibouchi, M. (2012, April). Public key compression and modulus switching for fully homomorphic encryption over the integers. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques* (pp. 446-464). Springer.

Dasgupta, S., & Pal, S. K. (2016). Design of a polynomial ring based symmetric homomorphic encryption scheme. *Perspectives on Science*, *8*, 692–695. doi:10.1016/j.pisc.2016.06.061

Dawahdeh, Z. E., Yaakob, S. N., & Sagheer, A. M. (2015). Modified ElGamal elliptic curve cryptosystem using hexadecimal representation. *Indian Journal of Science and Technology*, *8*(15).

Filho, G. F., Silva, G. P., & Miceli, C. (2016, July). Public Key Compression Method for Fully Homomorphic Encryption using Genetic Algorithms. In *19th International Conference on Information Fusion*.

Gentry, C. (2009, September). *A fully homomorphic encryption scheme*. Stanford University: The ACM Digital Library.

Gerasimov, A. N., Epishkina, A. V., & Kogos, K. G. (2017). Research of homomorphic encryption algorithms over integers. In *IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)* (pp. 398-403).

Goldwasser, S., & Micali, S. (1982). Probabilistic encryption and how to play mental poker keeping secret all partial information. In *Proceedings of 14th Symposium on Theory of Computing* (pp. 365–377). doi:10.1145/800070.802212

Hong, M. (2016, September). Homomorphic Encryption Scheme Based on Elliptic Curve Cryptography for Privacy Protection of Cloud Computing. In *IEEE International Conference on High Performance and Smart Computing (HPSC)* (pp. 152-157).

Hong, M., Wang, P. Y., & Zhao, W. B. (2016). Homomorphic Encryption Scheme Based on Elliptic Curve Cryptography for Privacy Protection of Cloud Computing. In *IEEE 2nd International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS)*, New York, NY (pp. 152-157). doi:10.1109/BigDataSecurity-HPSC-IDS.2016.51

Hu, Y. (2013). *Improving the Efficiency of Homomorphic Encryption Schemes*. Worcester Polytechnic Institute.

Kocaba, O. (2016). *Design and Analysis of Privacy-Preserving Medical Cloud Computing Systems.*

Li, A., Abd El-Latif, A. A., & Niu, X. (2012). Elliptic curve ElGamal based homomorphic image encryption scheme for sharing secret images. *Signal Processing*, *92*(4), 1069–1078. doi:10.1016/j.sigpro.2011.10.020

Ogburna, M., Turnerb, C., & Dahalc, P. (2013). Homomorphic Encryption. *Procedia Computer Science*, *20*, 502–509. doi:10.1016/j.procs.2013.09.310

Rivest, R.L., Adleman, L., & Michael L.D. (1987). On data banks and privacy homeomorphisms. *Foundations of secure computation*, *4*(11), 169-180.

Sharma, T. (2016, May). E-Voting using Homomorphic Encryption Scheme. *International Journal of Computers and Applications*, *141*(13).

Stallings, W. (2011). Cryptography and network security principles and practice. Prentice Hall.

Sunuwar, R., & Samal, S. K. (2015). *Elgamal Encryption using Elliptic Curve Cryptography*. Lincoln, NB: Cryptography and Computer Security.

Suveetha, K., & Manju, T. (2016). Ensuring Confidentiality of Cloud Data using Homomorphic Encryption. *Indian Journal of Science and Technology*, *9*(8).

Wenxiu, D., Yan, Z., & Robert, H. (2017). Encrypted data processing with Homomorphic Re-Encryption. *Information Sciences*, *409*, 35–55.

Yang, Y., Zhang, S., Yang, J., Li, J., & Li, Z. (2014, October). Targeted Fully Homomorphic Encryption Based on a Double Decryption Algorithm for Polynomials. *Tsinghua Science and Technology*, *19*(5), 478–485. doi:10.1109/TST.2014.6919824

Yi, X., Paulet, R., & Bertino, E. (2014). *Homomorphic Encryption and Applications*. Victoria, Australia: Springer International Publishing. doi:10.1007/978-3-319-12229-8

Marwan Majeed Nayyef completed a bachelor's degree from Anbar University and then completed the fourth-year hosting at the University of Nahrain. Naryyef graduated from Anbar University and got a seat in the graduate studies of the same university.

Ali Makki Sagheer was born in Iraq in 1979. He got on B.Sc. of Information System in Computer Science Department at the University of Technology (2001), Iraq, M.Sc. in Data Security from the University of Technology (2004), Iraq, and a Ph.D. in Computer Science from the University of Technology (2007), Iraq. He is interested in the following fields: cryptology, information security, number theory, multimedia compression, image processing, coding systems and artificial intelligence. He has published many papers in different scientific journals and conferences. Finally, he obtained a Professor scientific degree in cryptography and information security since July 18, 2015.