2017 International Conference on Intelligent Computing and Control (I2C2)

Homomorphic Encryption-State of the Art

Maya Mohan[1]
Department of CSE
NSSCE, Palakkad, India
mayajeevan@gmail.com

M K Kavitha Devi[2]
Department of CSE
TCE, Madurai
India

Jeevan Prakash V[3]
Department of Mathematics
NSSCE, Palakkad
India

***Abstract*— Homomorphic Encryption has become a viable method for privacy preservation. It basically deals with the computations of encoded data. Homomorphic encryption scheme allows operations on encrypted data which, when decrypted will provide the same results as we performed directly on the raw data. Certain cryptographic algorithms having the homomorphic property. A detailed survey is carried out on the various homomorphic encryption scheme based on the parameter involved, the encryption decryption mechanisms, their hammer-phic properties, security considerations etc. The encryption methods with homomorphic property are much suitable for applications require data privacy and security.**

***Index Terms*— Security, Privacy Preservation, Cryptography, Partially Homomorphic Encryption, Somewhat Homomorphic Encryption, Fully Homomorphic Encryption.**

I. INTRODUCTION

In homomorphic cryptosystem, the encryption function preserves homomorphism by holding the group operations applied on the ciphertext. Homomorphic encryption [1] is executed on ciphertexts for ensuring confidentiality and privacy. The computations executed in the cryptographic domain produce new encrypted data which, when decrypted exactly resembles the result of the computations performed on the plaintext. In normal scenario, in order to perform a computation on the encrypted data, the data need to be decrypted. For each computation, the paired operations, encryption and decryption are taking place. This can be avoided by using Homomorphic Encryption (HE). In HE, the operations are directly performed on the encrypted data. The resultant data when decrypted will result in a plaintext which matches the plaintext that is obtained by performing the same operations on the original plaintext.

HE allows the collaboration of different services without exposing the data to each of these services. Conventional encryption methods protect data while in the store and in transit but not during computation. HE preserves the security of the data during computation too. The scenario is shown in Fig. I. Operation Preserving (OP) [2] mapping is performed in HE. In order to achieve a reliable HE the mapping between the operations on the plaintext and the ciphertext should be the same or should produce same results.

Remark 1: Consider the data a, b ϵ $\{0, 1\}^k$ where $k \geq 1$ then f (a, b) = Dec [f (Enc (a)), Enc (b))] where f can be +, - or $\oplus$ operations.
Enc← Encryption Dec ← Decryption f ← function applied to the data

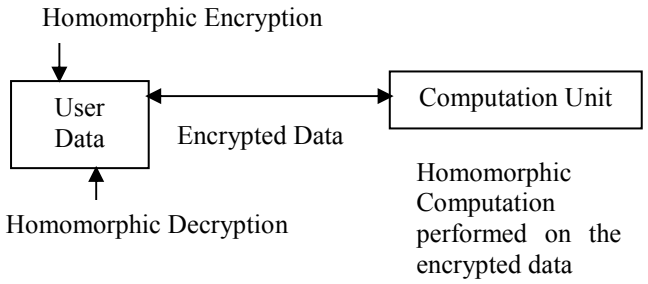
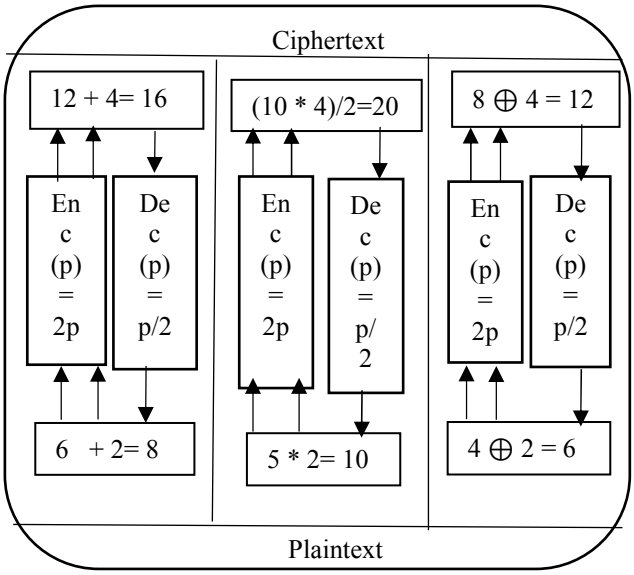


Fig. I Homomorphic Encryption

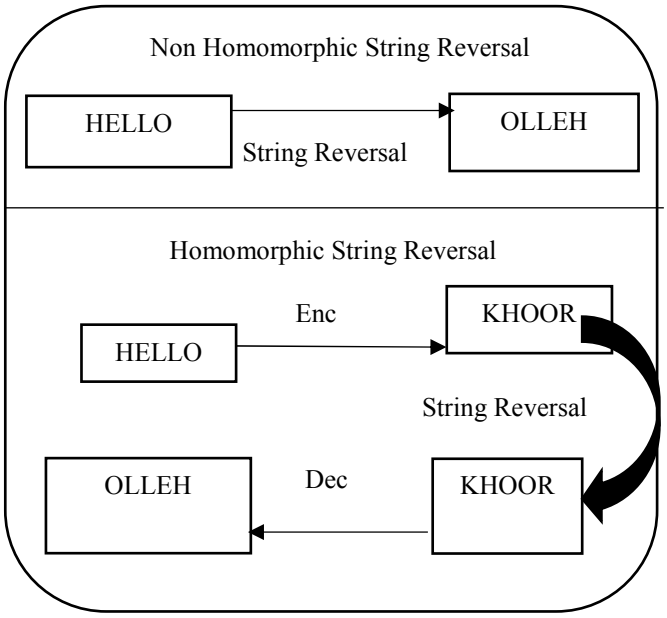
The examples given in Fig. II explains the homomorphic mechanism applied to the plaintext. In example (a) three operations are illustrated with three set of data. The plaintext (p) and the ciphertext (c) belongs to the set of integers. The operations addition, multiplication and XOR are applied to a collection of integers. The encryption and decryption functions considered are doubling and halving a number. In case of addition 6, 2 are the plaintexts and their sum is 8.While performing encryption of the data 6, 2 gives 12, 4 and their sum is 16.On performing decryption on 16, obtains 8 which is equivalent to the calculation performed on the raw data. In the similar way multiplication and XOR operations are performed. But this is not ensuring valid result for all sorts of data. In example (b) String reversal is performed using homomorphic encryption. The point to be noted is that the operation performed on the input ciphertext will produce another ciphertext which, when decrypted will produce a plaintext corresponding to the chosen operation on the input plaintext.

A. Significance of HE

HE can be widely used in the financial and medical industries for the protection of data [3]. Sony's play station network got hacked in 2011 and billions of personal information got disclosed due to unencrypted data storage [4].Similar incidents are happening till date because of improper security measures. Nowadays there are lots of data storing in the cloud by different organizations, business icons as well as individuals. These data may outsource to different locations for performing various computations. HE is the best choice for manipulation of data residing in the cloud [5] in a secure way. By design homomorphic encryption schemes are malleable [6]. Various cryptosystems existing today holds the homomorphic property which can be utilized for performing homomorphic encryption [7].



(a)



(b)

Fig.2 Homomorphic Encryption Examples

B. Classification of HE Schemes

A Homomorphic encryption scheme is said to be efficient if the ciphertext size is polynomially bounded with respect to the security parameter over repeated computation. Homomorphic Encryption scheme belongs to any of the main three categories like Fully Homomorphic Encryption (FHE), Partially Homomorphic Encryption (PHE) and Somewhat Homomorphic Encryption (SHE). Apart from these main classifications, there are lots of other categories based on the type of computations performed on the encrypted data, the keys used that are discussed below.

- Partially Homomorphic Encryption- In PHE only some operations can be performed on the encrypted data. For example addition and multiplication are the two given operations, then any one of them can be performed on the encrypted data but not both. PHE schemes are useful for certain applications which requires specific computations. Helios Voting scheme [8] make use of PHE. By this approach the encrypted voted ballots can be publicly stored in the cloud and the public can count the votes as well as verify their votes for each candidate without producing any proofs to the third party.
- Somewhat Homomorphic Encryption-In SHE scheme more than one operation can be performed on the encrypted data. But the limitation of SHE is that no all operations apply to all types of data. Even though it can support multiple operations, but only in limited numbers. SHE is suitable for a variety of real time applications such as financial, medical and recommender systems. Since SHE supports a limited number of operations it will be much faster than fully homomorphic schemes.
- Fully Homomorphic Encryption - FHE scheme supports any number of operations on any encrypted data. The circuit designed for FHE is homomorphically evaluated. This will be suitable for any sort of application working with encrypted data. Compared to PHE and SHE, FHE is little bit less efficient due to the computational overhead. As of today, for a particular application PHE and SHE is showing better efficiency compared to FHE scheme.
- Additive Homomorphic Encryption-Additive Homomorphism deals with addition of encrypted data. Let us consider two data *'a'* and *'b'* and is encrypted to *Enc*(*a*)and *Enc*(*b*).On applying additive homomorphic encryption, we get as given in (1),

$$Enc\,(a) + Enc\,(b) = Enc\,(a + b) \quad (1)$$

According to additive homomorphism, the sum of the encrypted messages is equivalent to the encrypted sum of those messages.

- Multiplicative Homomorphic Encryption-Multiplicative homomorphism talks about multiplication of the encrypted data. Let us assume two data *'a'* and *'b'* and is encrypted to *Enc(a)*and *Enc(b).*While applying multiplicative homomorphic encryption leads to (2),

$$Enc\,(a) \times Enc\,(b) = Enc\,(a \times b) \quad (2)$$

- Algebraic Homomorphic Encryption-If *'f'* is a function from $A \rightarrow B$ and k ϵ K and *a, b* ϵA then the equations (3), (4), (5) form Algebraic Homomorphism.

$$f(k \times a) = k \times f(a) \quad (3)$$
$$f(a + b) = f(a) + f(b) \quad (4)$$
$$f(a \times b) = f(a) \times f(b) \quad (5)$$

According to algebraic homomorphy developing algebraic homomorphic encryption scheme is really a challenging one. Lots of proposals are evolved, but none of them given a satisfactory solution [9].

- Somewhat Homomorphic Encryption- uses secret key under the approximate GCD assumption

Step 1. Select the secret key p (a large odd number) choose the security parameter, k

Step 2. Encrypt the bit 'm'. Randomly choose a large multiple of p. Let us assume q×p (q ≈ K^5 bits).Randomly pick small even number 2 × r (r ≈ k bits). The ciphertext c = q × p + 2 ×r + m

Step 3. Decrypt the bit c, c mod p = 2×r + m (mod p) where 2r is the noise accumulated. Consider only the least significant bit as the plaintext.

- Addition and multiplication of encrypted data using secret key homomorphic encryption

Consider two ciphertexts c_1 and c_2 , where

$$c_1 = q_1p + (2r_1 + m_1)$$
$$c_2 = q_2p + (2r_2 + m_2)$$

Addition and multiplication of ciphertext c_1 and c_2 are as Follows,

$$c_1 + c_2 = q_1p + (2r_1 + m_1) + q_2p + (2r_2 + m_2)$$
$$\text{LSB} = m_1 \oplus m_2$$
$$c_1 \_ c_2 = q_1p + (2r_1 + m_1) \times q_2p + (2r_2 + m_2)$$
$$\text{LSB} = m_1 \text{ and } m_2$$

When the number of operations increase, ciphertext size also increases. Because of this drawback not apt for majority of the applications. The noise level increases once the operations are increasing.

- Somewhat Homomorphic Encryption Based on public keys

Step1. Select the secret key p, an odd number consist of n^2 bits

Step2. Calculate the public key, pk such that

$$pk = [q_0p + 2r_0, q_1p + 2r_1 \ldots . q_tp + 2r_t]$$
$$= (x_0, x_1 \ldots . , x_t)$$

Step3. Encryption of the bit m,

$$c = \Sigma_{i \epsilon s} x_i + 2r + m \ (mod\ x_0)$$

Where s is a randomly chosen subset such that s ϵ (1.. t)

Step4. Decryption of c,

$$C \bmod p = 2r + b \ (mod\ p) = 2r + b$$

The least significant bit is taken into consideration.

But limitation with this method is that it can perform only a limited number of operations. Noise gets accumulated when the operations increases.

- Fully Homomorphic Encryption with Public Key -FHE using public key can be effectively implemented using bootstrapping method. In bootstrapping method, whenever the noise level increase beyond the threshold level, re-encryption is performed to reset the noise level. Once advantage with approach is that the noise is level is under control.

The paper aiming to have an understanding about homomorphic encryption, its classifications, significance, limitations and applications. In section II various homomorphic encryption schemes are explained, section III deals the various applications and section IV, the conclusion.

II. Various Homomorphic Encryption Schemes

Homomorphic Encryption basically consists of four tuples representing the functions Key Generation, Encryption, Decryption, Evaluation and is represented as, $H = (KG, Enc, Dec, Eval)$

- Key Generation: Generating the private-public key pair for performing encryption and decryption by considering the security parameter as the input.
- Encryption: The encryption algorithm takes the plaintext and the public key as the input and generate the ciphertext as the output
- Decryption: The decryption algorithm takes the ciphertext and the private key as the input and produces the plaintext as the output.
- Evaluation: The evaluation function performs the homomorphic evaluation of the ciphertext. Evaluation function takes the public key, a function and the ciphertext as inputs and return another ciphertext as output.

$C^* = Eval_{pk}(f, c)$

$c \leftarrow$ Ciphertext generated using for homomorphic encryption

$c^* \leftarrow$ Ciphertext after performing homomorphic encryption

$f \leftarrow$ Function used for the computation of the ciphertext

$pk \leftarrow$ *Public* key

Remark 1. An encryption scheme is said to be homomorphic, if for all messages m_1, m_2, m_n and the function belongs to the set of all functions c then

$$(PU_k, PR_k) \leftarrow Gen(1^k) \text{ where } k \geq 0 \text{ then the ciphertexts}$$

$$c_1 \leftarrow Enc(PU_k, m_1); c_2 \leftarrow Enc(PU_k, m_2); c_n \leftarrow Enc(PU_k, m_n)$$

$$C^* \leftarrow Eval(PU_k, f, c_1, c_2 \ldots c_n) \text{ and } m \leftarrow Dec(PR_k, C^*)$$

If c is a class of all functions or the evaluation satisfies for any arbitrary function for all sort of data then the scheme is said to be fully homomorphic.

A. RSA Encryption Scheme

The RSA encryption scheme [10] exhibits multiplicative homomorphism. The Algorithm with its homomorphic properties are depicted in Table I. The highest security level for any homomorphic encryption scheme is IND-CPA.

Unfortunately the deterministic property of RSA fails in achieving IND-CPA. If any restructuring like OAEP (Optimal Asymmetric Encryption Padding) applying on RSA lifts the homomorphic property. Constructing a semantically secure homomorphic encryption based on RSA is still an open issue.

TABLE I. RSA CRYPTOSYTEM

Key Generation: 1:Select two large prime numbers p and q such that $n = p \times q$ 2.Calculate $\phi(n) = (p-1)(q-1)$ 3.Select a random key e, where $1 < e < \phi(n)$ and $gcd\ (e; \phi(n)) = 1$ 4. Calculate the private key d such that $d = e^{-1} mod\ \phi(n)$ The public key=[$e; n$] The private key=[$d; n$]
Encryption: Encryption of message m using public key produces the ciphertext c $c = m^e\ mod\ n$
Decryption: Decrypting the ciphertext c using the private key gives the plaintext m $m = c^d mod\ n$
Homomorphic Multiplication: $c_1 = m_1^e\ mod\ n$ $c_2 = m_2^e\ mod\ n$ $c_1 \times c_2 = m_1^e\ mod\ n \times m_2^e\ mod\ n == (m_1 \times m_2)^e\ mod\ n$ i.e $Enc(m_1) \times Enc(m_2) = Enc(m_1 \times m_2)$

B. ElGamal Encryption Scheme

ElGamal is a public key based cryptosystem introduced in the year 1985 by ElGamal, manifests multiplicative homomorphism. ElGamal construction works for any sort of group for which suitable discrete logarithm is found to be ungovernable. ElGamal and its siblings are the right candidates for the homomorphic encryption scheme. The homomorphic properties of the scheme are listed in Table II.

C. Goldwasser-Micali Cryptosystem

Goldwasser- Micali scheme (GM) is a public key based cryptosystem introduced in the year 1982 by Goldwasser and Micali shows additive homomorphism. It is the first probabilistic public key encryption scheme considered to be provably secure according to the cryptographic assumptions. But the method is impractical due to the large size of the generated ciphertext for a given plaintext.GM is a combination of three algorithms (K, E, D), where K is the key generation algorithm, E is the encryption algorithm and D is the decryption algorithm. The algorithms K and E are probabilistic whereas D is purely deterministic in nature. The algorithm with its homomorphic property is given in Table III. In order to achieve semantic security, the size of the ciphertext should be larger than the plaintext in GM encryption scheme, and the size of the generated ciphertext is 1024 times more than the size of the plaintext.

D. Benaloh Homomorphic Encryption scheme

Benaloh scheme is an augmented form of GM scheme introduced in 1994 provides additive homomorphism. Benaloh scheme designed for the encryption of blocks of data, whereas GM encrypts bit by bit. The algorithm and its homomorphic property are discussed in Table IV.

The Decryption function is a bit complex but the encryption cost is not too high and easy to perform. Better efficiency can be achieved by choosing small block sizes.

E. Paillier Encryption scheme

Paillier cryptosystem developed in 1999 by Pascal Paillier comes under the category of public key cryptography. The scheme exhibits additive homomorphic property, which appreciate like design of voting protocols, water marking, secret sharing schemes etc. Self-blinding property of Paillier cryptosystem has potential applications in the field of cryptography. The scheme is illustrated in Table V. The Paillier cryptosystem is based on decisional composite residuosity assumption. The cryptosystem is IND-CPA secure. The system is considered to be malleable, hence not IND-CCA2 secure. In Paillier cryptosystem the plaintext-ciphertext space ratio is ½.

TABLE II. ELGAMAL CRYPTOSYSTEM

Key Generation: 1.Select a large prime number p 2.choose a primitive root,α of p such that $\alpha < p$ 3.Select a random number k as the secret key, where $k < p$ 4.Calculate the public key Y such that $Y = \alpha^k\ mod\ p$
Encryption: 1.Select a random number r 2. Calculate $c_1 = \alpha^r$ mod p 3. Calculate the shared key K ,$K = Y^r$ modp 2. Calculate $c_2 = m \times K$ mod p The encrypted message is (c_1, c_2)
Decryption: 1. Calculate K = c1k mod p 2. m = c2 × K-1modp
Homomorphic Multiplication: $c_1 = (\alpha^{r1}, m_1 \times Y^{r1})$ $c_2 = (\alpha^{r2}, m_2 \times Y^{r2})$ $c_1 \times c_2 = (\alpha^{(r1+r2)}, m_1 \times m_2 \times Y^{(r1+r2)})$ ie $Enc(m_1) \times Enc(m_2) = (\alpha^{(r1+r2)}; m_1 \times m_2 \times Y^{(r1+r2)})$ $= Enc(m_1 \times m_2)$

TABLE III. GOLDWASSER- MICALI CRYPTOSYSTEM

Key Generation: 1.Select two random prime numbers p and q 2.Calculate $n = p \times q$ 3.Select a random number r such that $[\ r/p\ ] = [\ r/q\ ] = -1$ where r is not a quadratic residue(QR) but has Jacobi symbol 1 4.Calculate the public key Y such that Y = (n, r) The private key=(p, q)
Encryption: Encryption is performed bit by bit of the plaintext. To encrypt a bit b, 1.Select a random number k such that $k \in Z_n^*$ 2. Calculate $c = k^2 \times r^b$ mod p
Decryption: For each ciphertext using prime factorization(p, q) determine whether the ciphertext is a QR if yes b = 0 else b = 1
Homomorphic Multiplication: $c_1 = (k_1^2 \times r^{b1})$ $c_2 = (k_2^2 \times r^{b2})$ $c_1 \times c_2 = (k_1 \times k_2)^2 \times r^{(b1+b2)}$ ie $Enc(b_1) \times Enc(b_2) = Enc(b_1 \oplus b_2)$

F. Some more Innovations

Naccache-Stern is an extended version of the Benaloh Cryptosystem. The security lies in the higher residuosity assumption. Apart from Benaloh's scheme Naccache -Stern

scheme, consider a set of prime numbers denoted as 'k' that leads to small expansion. The encryption is almost same as the Benaloh scheme, but decryption varies. According to the authors it is advisable to choose the set k as to get an expansion equal to 4. By changing the base group G Okamoto-Uchiyama scheme brings some improvements to Naccache-Stern and Benaloh schemes. This could able to achieve an expansion of 3.Semantic security is unachievable if the ciphertext space and the plaintext space are of same size. A generalized scheme of Paillier cryptosystem proposed by Damgard and Jurik in 2001.The scheme deals with the integers belong to the group of the form Z_n^{s+1} where s > 0.When the s value increases the expansion decreases. If an adversary can break the system for a particular value of s then it is possible to break the system with value s-1.The scheme could able to achieve same level of semantic security as that of Paillier cryptosystem. The expansion can be equal to 1 + 1/ s. For large values of s the expansion value approaches to 1.

TABLE IV. BENALOH HOMOMORPHIC CRYPTOSYSTEM

Key Generation: For a given block of plaintext of size ′ b′ , 1.Select two random prime numbers p and q such that b divides (p - 1) and n = p ×q gcd(b, (p-1)/b) = gcd(b,(q - 1)) = 1 and $\phi(n) = (p - 1)\ (q - 1)$ 2.Select a random number $Y \in Z_n^*$ such that $Y^{\phi(n)/b} \neq 1$ 3.Calculate $x= Y^{\phi(n)/b}$ The public key (Y, n) and the private key is ($\phi(n)$, x))
Encryption: Encryption is performed on the plaintext $m \in Z_b$, 1.Select a random number k such that $k \in Z_n$ 2. Calculate $E_b(m) = Y^m \times k^b \bmod n$
Decryption: For each ciphertext $c \in Z_n^*$ 1.Determine $a=C^{\phi(n)/b}$ 2.The plaintext m is calculated as, $m = \log_x(a)$ or $x^m = a \bmod n$
Homomorphic Multiplication: $c_1 = (Y^{m1} \times k_1^b) \bmod n$ $c_2 = (Y^{m2} \times k_2^b) \bmod n$ $c_1 \times c_2 = Y^{m1+m2} \times (k1k2)^b \bmod n$ ie $Enc(m_1) \times Enc(m_2) = Enc((m_1 + m_2) \bmod n)$

In the family of partially homomorphic scheme the last one to discuss is the ElGamal-Paillier Amalgam scheme which combines the Paillier cryptosystem and the ElGamal variant. The merged scheme looked forward to preserve the advantages of both scheme and suppressing the drawbacks. According to the scheme to encrypt the message 'm' where $m \in Z_n$, the sender chooses a random integer k and calculates the ciphertexts as,

$$(c_1, c_2) = (g^k \bmod n, (1 + n)^m (Y_A^k \bmod n)^n \bmod n^2)$$

Where n is the modulus value chosen to be large prime, k is the selected random number, g is the primitive root of n and Y_A is the public key. The combined scheme successfully decrypts the entire message m with the only initial expense of key generation. But the scheme faces lots of overhead due to the huge ciphertext generated compared to the combined scheme of ElGamal and Elliptic curve. Manipulation of the encrypted data in a meaningful way using any arbitrary function is achieved by using fully homomorphic encryption schemes. Arbitrary function in the sense it should go hand in hand with the chosen encrypted scheme .For a FHE scheme to be practical ,it is trivial to build up an encryption scheme that can manage all the functions.

TABLE V. PAILLIER CRYPTOSYSTEM

Key Generation: 1.Select two large random prime numbers p and q of equal length such that gcd (pq, (p - 1)(q - 1)) = 1 2. Calculate n = p × q and $\lambda(n) = lcm(p - 1, q - 1)$ (λ(n) -Carmichael's function) 3.Select a random number g such that $g \in Z_{n2}^*$ 4.Check whether n divides the order of g Check for the modular multiplicative exist for $(L(g^{\lambda} \bmod n^2)) \bmod n$ ie, $\mu = (L(g^{\lambda} \bmod n^2))^{-1} \bmod n$ The function L(U) = U-1/n The public key (n, g) and the private key is (λ,μ)
Encryption: Encryption is performed on the plaintext $m \in Z_n$, 1.Select a random number r such that $r \in Z_n^*$ 2. Calculate $c = g^m \times r^n \bmod n^2$
Decryption: For each ciphertext $c \in Z_n^2$ 2.The plaintext m is calculated as, $m = L(c^{\lambda} \bmod n^2) \times \mu \bmod n$
Homomorphic Multiplication: $c_1 = (g^{m1} \times r_1^n) \bmod n^2$ $c_2 = (g^{m2} \times r_2^n) \bmod n^2$ $c_1 \times c_2 = Y^{m1+m2} \_ k_1k_2^b) \bmod n$ ie $Enc(m_1) \times Enc(m_2) = (g^{m1+m2} \times r_1 + r_2^n) \bmod n^2$ $= Enc(m_1 + m_2 \bmod n^2)$

The development of FHE has great practical implications in the context of cloud computing. It is desirable to have non-determinism, semantic security (one wayness) and IND-CPA secure properties for the designed FHE. Each time an operation is performed on the encrypted data results in some noise. While performing the operations such as multiplication, the noise level exponentially grows. This may lead to unsuccessful decryption. The size of the key in FHE is quite large compared to the normal encryption scheme. This can be improved by developing the FHE scheme based on learning with errors (LWE) with a better key size, introduced by Regev .

G. Hentry's Fully Homomorphic Encryption

The first FHE scheme is introduced in the year 2009 by Craig Hentry. He developed a bootstrapped based re-encryption scheme to achieve complete homomorphism. While performing homomorphic encryption the noise level get increased. A threshold level is set in the algorithm and once the noise exceeds the threshold level by re-encrypting the encrypted data, it is get adjusted. Consider the data to perform homomorphic encryption is 'a'. On applying encryption obtains Enc (a) which is having a noise level $N^* > N$, where N is the threshold value. After applying recryption, resulting in Enc (a) with noise level Sqrt (N). Hentry's scheme make use of SHE on ideal lattices. By applying bootstrapping using recryption on SHE obtains FHE. In lattice based encryption the circuit complexity of the decryption algorithm is very low compared to exponential based algorithms such as RSA and

ElGamal. Since ideal lattices are ideals in the polynomial ring, inherits the operations addition and multiplication. Integers are replaced by ideal lattices because of good and bad bases. Good bases are used as secret keys and bad bases are used as public keys. sThe key generation of Hentry's scheme make use of Smart-Vercauteren method. The public key for the FHE is the Hermite normal form of V and the private key is the single odd coefficient of $\vec{W}$ .Encryption, decryption and the homomorphic properties are given in Table VI.

TABLE VI. HENTRY'S SCHEME

Key Generation: Select an odd number P such that P > 2N, where N is the threshold value.
Encryption: For encrypting the bit 'b' ,the corresponding ciphertext 'c' is, c = b + 2X + KP where X = [-n /2 , n /2] and K any random integer
Decryption: For each ciphertext c ,the plaintext b is calculated as b=(c mod P)mod 2 where c mod P is the noise in the range[-n,n] Decryption is successful if ,b + 2X ϵ [N,N] ⊆ [-p/2 ,p/2]
Homomorphic Multiplication: c1 + c2 = b1 ⊕ b2 + 2X + KP c1 ×c2 = b1 × b2 + 2X + KP

H. Zaryab Khan's Scheme

Zaryab Khan introduced a fully homomorphic scheme based on perfectly colorblind function. Unlike from other FHE schemes this scheme is not generating any noise in the ciphertext. This is considered to be the fastest public key encryption scheme available today. In this scheme if the public key is given, it is extremely difficult to find the secret key. For a given ciphertext, it is mathematically hard to obtain the plaintext. The scheme is quite adoptable unlike Gentry's scheme which make use of ideal lattices which are computationally very expensive. The key generation of the scheme is a bit complex compared to other schemes. The homomorphic property is explained in Table VII.

III. APPLICATION OF HOMOMORPHIC ENCRYPTION

A survey has been carried on various applications of homomorphic encryption schemes are shown in Fig. III. It is observed that in the recent years plenty of applications including cloud computing use homomorphic encryption. The application areas include cloud computing, voting schemes, statistical analysis, multiparty computation, recommender systems etc.

IV. CONCLUSION

The paper gives a glimpse on various Homomorphic encryption standards. It addresses the parameters of various schemes and their homomorphic properties. Discussions are carried out regarding the security issues related to each scheme. A detail analysis is executed, which helps in identifying the suitable applications for each scheme. It will be very useful to the research community to work in Homomorphic encryption. This survey will be adequate to understand the need for homomorphic encryption for privacy preservation.

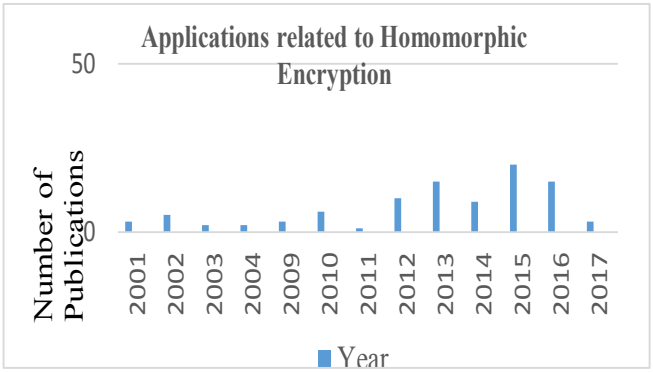


Fig .III Survey on Homomorphic Encryption Applications

TABLE VII. ARYAB KHAN'S SCHEME

Encryption: 1.Choose a random number l lies in (0, P^{ms}), 2.choose a random integer i lies in β- 3 < i < β 3.Toss a binary coin (T or F) If T, choose a numerical teichmuller character N ϵ ringY $E(m) = (e_{k1})^m \times (e_{k2})^{rnd} \times (l)^{pi} \times N \bmod (q(Y)/p^{\beta})$ If F , $E(m) = (e_{k1})^m \times (ek2)^{rnd} \times (l)^{pi} \bmod (q(Y)/p^{\beta})$ where rnd represents any random number
Decryption: 1.c' = $\psi^{-1}(c)$ 2. c'' = (log(c')mod(p(x)/p^{β}) 3. m = [(a_0^{-1})×(m') mod (P^{ms})]/p
Homomorphic Addition: $E(m_1) \times E(m_2) = E(m_1 + m_2)$ **Homomorphic Multiplication:** $E(m_1)^{m2} = E(m_1 \times m_2)$

REFERENCES

[1] Monique Ogburn , Claude Turner, Pushkar Dahal , "Homomorphic Encryption" ,Procedia Computer Science, vol. 20, pp. 502-509, 2013

[2] Popa R. A., Red_eld C. M. S.,Zeldovich N. and Balakrishnan H., "Protecting Con_dentiality with Encrypted Query Processing" ,Proceedings of the 23rd ACM Symposium on Operating Systems Principles (SOSP 2011), Cascais, Portuga,2011

[3] Diffie,Whit_eld and Martin E Hellman,"New directions in Cryptography" ,IEEE Transactions on Information Theory,22(6),pp 644-654,1976.

[4] K.Thomas,"Sony Makes it Official: PlayStation Network Hacked", PC Computing, pp. 12,2011

[5] Maha TEBAA,Said EL HAJJI, Adellatif EL GHAZI," Homomorphic encryption applied to the cloud computing security" ,Proceedings of the world congress on Engineering,vol.1,2012

[6] D. Dolev, C. Dwork, and M. Naor, " Nonmalleable cryptography" SIAM J. Comput., 30(2), pp. 391437,2000.

[7] R.Rivest,L.Adleman and M.Dertouzos, "On data banks and privacy homomorphisms ",In foundation of Secure Computation,Academic Press, pp-169-177,1978

[8] Ralf Kusters, Tomasz Truderung and Andreas Vogt ," Clash Attacks on the Veri_ability of E-Voting Systems", IEEE Symposium on Security and Privacy,pp-395-409,2012

[9] Jean Sebastien Coron, Tancrede Lepoint and Mehdi Tibouchi,Batch fully homomorphic encryption over the integers, Cryptology ePrintArchive, Report 2013/036,2013

[10] Rivest, R., Shamir, A.,and Adleman, L. ,"A Method for Obtaining Digital Signatures and Public-Key Cryptosystems,Communications of the ACM, 21(2), pp. 120-126,1978.