IMPROVED SYMMETRIC BASED ALGORITHM FOR PRESERVING DATA IN PUBLIC CLOUD STORAGE WITH HOMOMORPHIC ENCRYPTION

[1]Ashwini Gulhane, [2]Gaddale Jayabharathi
[1]Assistant Professor, Dept. of Computer Science and Engineering, Kg Reddy college of Engineering and Technology, Moinabad, Hyderabad, India
[2]Assistant Professor, Dept. of Computer Science and Engineering, Kg Reddy college of Engineering and Technology, Moinabad, Hyderabad, India

Abstract: The outstanding research topic on cryptography is called Fully Homomorphic Encryption (FHE), which allows computations to be carried out on encrypted data to the entrust server of the security and privacy concerned related to rising technologies like cloud computing. Various FHE schemes were industrial after the first creation of Craig Gentry in 2009 which security is relying on bootstrapping and censorship. It is based on ideal net and it is more many-sided and huge computational cost and unpractical, but it is enough for hypothetically possible. In this paper, symmetric based frivolous Fully Homomorphism Encryption theme is projected for the somewhat similarity set up that's supported GV system and uses on matrix rather than integers. We have a tendency to scale back the key size considerably by introducing Reduced Approximate GCD drawback. Another half is proving the theme is semantically secure below Approximate GCD. Finally, we have a tendency to propose a replacement algorithmic rule for key generation and fresh for every computation with a stipulated quantity.

Keywords: Cryptography, Homomorphism Encryption, Untreated server, Computation, Bootstrapping, Squashing.

INTRODUCTION

Cloud computing is an emerging technology where we can store and access the resources and infrastructures via Internet with low cost. The Cloud users can outsource the IT products from the Cloud Service Providers (CSPs) for development of various projects for a stipulated time [1]. Simply a cloud computing may be a shopping for a resources like software's, tools , applications, servers, cupboard space and network setup through on-line with low value and high potency on a contract amount. Today's most of IT firms maintain their own cloud for creating the simple and guide primarily based accessed product.

The biggest challenge for researchers is providing security and privacy of cloud users/customers knowledge that's unbroken hold on cloud servers. One in every of the prevailing ways for knowledge security is encryption. In coding| encoding| encryption} the cloud users/customer wills encryption for his or her knowledge by employing a key and algorithmic program. If the secret's same for each coding and secret writing referred to as bilateral wherever as coding secret's public key and secret writing key referred to as non-public key because the system in uneven or public key cryptography. The cloud user unbroken their sensitive knowledge at cloud servers with encrypted format. But any computations like looking, sorting, addition or multiplication and XOR operations on cipher text. The client can decrypt the cipher text which is stored in cloud server by using key or client can give the credentials like key to the CSPs for doing manipulations on server. However consumer cannot trust the CSPs for change the client's confidential information whereas exploiting security levels [2]. Then, to conserving the information secure at server aspect, the consumer itself do the cryptography method to get plaintext and do the any kind of modification, calculations and update the plaintext and later perform the encoding and generate the cipher text to preserve at server. Here, we tend to observe the some security defect is that frequent cryptography and computation could gain likelihood to the attackers to chosen cipher text attack and exploit the information integrity and authentication. To avoid the higher than, in 1978, the Rivert et all proposes the privacy similarity [3] that will the computations on cipher text rather than decrypting it.

HOMOMORPHIC ENCRYPTION

Homomorphic Encryption is a shape of encryption where we do some calculation on already encrypted data without decryption and matches the results of which perform calculations on plain text. Homomorphic cryptography is made technique in cloud computing whereas doing shopper server communiqué with multiple navigations. In cloud computing customers will store their sensitive knowledge on cloud, however to perform any computations like change, looking out and sorting, they alleged to decode the cipher text and perform operations and send to the cloud within the kind of cryptography. But multiple times decoding and change the info, the shoppers ought to depends on the cloud service suppliers (CSPs) [4][25]. In spite the cloud users depends on the CSPs for key distribution. Homomorphic cryptography provides AN atmosphere to the shoppers no ought to share the keys to any cloud service suppliers whereas storing or accessing the cloud. Homomorphic cryptography is that the conversions of information into cipher text that may be analyzed and worked with as if it were still in its original type. Homomorphic cryptography permits advanced mathematical operations to be performed on encrypted knowledge while not compromising the cryptography. In arithmetic, Homomorphic describes the transformation of 1 knowledge set into another whereas protective relationships between components in each sets. The term comes from the Greek words for "same structure." as a result of the info in an exceedingly Homomorphic cryptography theme retains constant structure, identical mathematical operations, whether or not they area unit performed on encrypted or decrypted knowledge can yield equivalent results. Homomorphic cryptography is anticipated to play a vital half in cloud computing, permitting firms to store encrypted knowledge in an exceedingly public cloud and make the most of the cloud provider's analytic services [5].

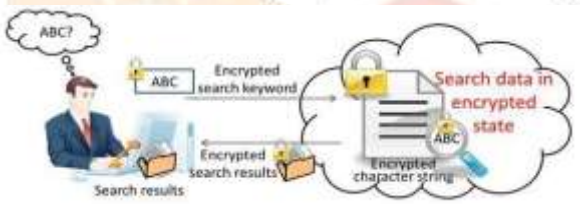

Figure 1: Homomorphic Encryption

The HE is defined as the random computations (i.e. Functions) are perform on encrypted data at server side without meaningful the decryption key. Though when results are decrypted by shopper mistreatment non-public key that matches similar results performed on the plaintext. The subsequent equation shows the fundamental definition of Homomorphic coding. The operations embrace looking out, sorting, filtering, and addition, multiplication, XOR and any operate that are performed on cipher text by the server while not knowing the key. Once the shopper decipher it matches the results of operations to the performed on plaintext.

$$\text{Operation (plain)} = = \text{decrypt (operation}' \text{ (encrypt (plain)))}$$

The figure1 shows that, consumer got to rummage around for knowledge an information known as first principle in cloud server and data in encrypted format known as question that's send to the cloud server for looking out the first principle string on the cloud server. The Homomorphic encoding principle searches the string consistent with the consumer demand and if found the send the results back to the consumer in encrypted format solely. The consumer by victimization their personal key decrypts the encrypted search results. In cryptography, Homomorphic encoding is method of acting similar operations on encoding information (cipher text) while not secret writing of cipher text and sends the results back to the consumer in encrypted format solely. However the result matches once acting same operations on plaintext. Let M (or C) denote the set of the plaintexts (or cipher texts, respectively). AN encoding theme is claimed to be Homomorphic if for any given encoding key k the encoding perform E satisfies [6]

$$m1, m2 \in M, E(m1 \circ m2) = E(m1) \circ E(m2)$$

Informally speaking, Homomorphic cryptosystem may be a cryptosystem with the extra property that there exists AN economical to reason A coding of the total or the merchandise, of 2 messages given the general public key and therefore the encryptions of the messages however not the messages themselves. If M (or C) is AN additive (semi-) cluster then the theme is formula known as Additively Homomorphic and therefore the formula is named addictive Homomorphic coding. Otherwise the theme is named increasingly Homomorphic and therefore the formula is named multiplicative Homomorphic coding. The subsequent table (i)

shows the assorted Homomorphic coding schemes with applications Samples".

Table I: Homomorphic Encryption methods and applications

Crypto system	Addictive	Multiplicative	XOR	Mixed	Application
Paillier	√	X	X	X	e-voting system, threshold scheme
RSA	X	√	X	X	To secure Internet, Banking and credit card secure transaction
ElGamal	X	√	X	X	In Hybrid systems
Goldwasser-Micali	X	X	√	X	Biometric Authentication
Boneh, Goh, and Nissim (BGN).	Many additions	One Multiplications	X	√	Stack Exchange
Brakerski, Gentry and Vaikuntanathan(BGV)	X	X	X	√	For the security of integer polynomials
EHC(Extended Homomorphic Cryptosystem)	X	X	X	√	Efficient Secure Message Transmission in MANETs

The advantages and applications of the Homomorphic Encryption scheme is include No longer usage of private of client by the Cloud service Providers (CSPs), In Medical records, Analyze disease/treatment without disclosing confidential data and able to search for DNA markers without revealing DNA [7], Spam filtering-Blacklisting encrypted mails and third parties can scan you PGP traffic and easy to implement the electronic voting system [8].

Somewhat Homomorphic Encryption (SHE)

- The Somewhat Homomorphic Encryption is defined as the computations to be carried out on cipher text as either additive or multiplicative and quadratic functions etc, but not both combined. The SHE is also called as Partially Homomorphic cryptosystem. The following list of cryptosystem is examples of SHE system with either additive or multiplicative operations.
- The Unpadded RSA [9]. If the RSA public key is modulus m and exponent e, then the encryption of a message x is given by $E(x) = x^e \bmod n$. The Homomorphic property is then $E(x1).E(x2) = x1^e\, x2^e \bmod n = E(x1.x2) \bmod n$.
- ElGamal, In the ElGamal cryptosystem[10], in a cyclic group G of order q with generator g, if the public key is (G,q,g,h), where $h=g^x$, and x is the secret key , then the encryption of a message m is $E(m)=(g^r, m.h^r)$,for some random r$\in${0,1,…,q-1}. The Homomorphic Encryption property is then defined as $E(m1).E(m2)=(g^{r1}, m1.h^{r1})(g^{r2}, m2.h^{r2})=(g^{r1+r2}, (m1.m2)h^{r1+r2})=E(m\ 1.m2)$.
- Goldwasser-Micali, In the Goldewasser-Micali cryptosystem [11], if the public key is the modulus m and quadratic non-residue x, then the encryption of a bit b is $E(b) = x^b r^2 \bmod m$, for some random r $\in$ {0, 1… m-1}. The Homomorphic property is then $E(b1).E(b2) = x^{b1} r1^2 x^{b2} r2^2 \bmod m = x^{b1+b2} (r1r2)^2 \bmod m = E(b1 \oplus b2)$ where $\oplus$ denotes addition modulo 2, (i.e. exclusive –or).
- Benaloh, In the Benaloh cryptosystem[12], if the public key is the modulus m and the base g with a block size of c, then the encryption of a message x is $E(x)=g^x r^c \bmod m$, for the random r $\in${0,…,m-1}. The Homomorphic property is then $E(x1).E(x2) \bmod m = (g^{x1} r1^c)(g^{x2} r2^c) \bmod m = g^{x1+x2}(r1r2)^c = E(x1+x2 \bmod m)$ aillier, In the Paillier cryptosystem [13], if the public key is the modulus m and the base g, then the encryption of a message x is $E(x) = g^x r^m \bmod m^2$, for some random r$\in$ {0,…, m-1}. The Homomorphic property is then $E(x1).E(x2) = (g^x 1 r1^m)\ (g^x 2 r2^m) \bmod m^2 = g^{x1+x2}(r1r2)^m \bmod m^2 = E(x1 + x2)$.

Fully Homomorphic Encryption (FHE)

A cryptosystem that supports discretional computations (i.e. each additive and increasing function on cipher texts is thought as totally homomorphic cryptography (FHE) and is way additional powerful. The existence of associate degree economical and totally homomorphic cryptosystem would have nice sensible implications within the outsourcing of personal computations within the cloud computing. The utility of totally homomorphic cryptography has been long recognized: the matter of constructing

such a theme was 1st planned in 1978, at intervals a year of the event of RSA. An answer proven additional elusive; for quite thirty years, it had been unclear whether or not totally homomorphic cryptography was even doable. Throughout that amount, partial results enclosed the Boneh–Goh–Nissim cryptosystem [14] that supports analysis of an infinite variety of addition operations however at the most one multiplication, and also the Ishai-Paskin cryptosystem [15] that supports analysis of (polynomial-size).

Table II: Existing Homomorphic Encryption scheme and their remarks

Year	Homomorphic Encryption	Remarks
2009	First FHE by Gentry	Gentry & Van Dijk simplify: No more lattices, integers instead
2010	Smart & Vercauteren	Simplify: Reduce key size
2011	Gentry & Halevi	Present working FHE implementation (the one w/2.3GB public key, runs on workstation)
	Coron, Naccache & Tibouchi	Reduces public key size
	Gentry & Brakerski	Shows removal bootstrapping(Leveled FHE)
	Smart & Vercauteren	Implement Single Instruction Multiple Data(SIMD)
2012	Gentry, Smart & Halevi	Elevate SIMD to general circuits.
	Boneh, Gentry & Halevi	Show another PoC using SWHE.
	Brakerski, Gentry & Vaikuntanathan	Improves Leveled FHE
2013	Shai Halevi	Improves speed and reduces noise by modulus switching
2014	GV Method	Reduced size of the public key and time complexity of both encryption and decryption Based on Error Approximate GCD.
2015	Cheon, Jung Hee	The polynomial approximate common divisor problem and its application to the fully homomorphic encryption
2016	Dasgupta Smaranika, and S. K. Pal.	Design of a polynomial ring based symmetric homomorphic encryption scheme

Proposed Fully Homomorphic Encryption

Main Idea: In this section, we have presented our proposed system for fully Homomorphic encryption, which is based on symmetric key. Our scheme consists of sub-modules, which are as below:-

-Key generation step

-Encryption procedure
-Decryption procedure
-Refresh procedure

The complete description and steps involved in our FHE scheme are presented as below:-

Algorithm 1 KeyGen (l, a)-

1. Randomly choose 2a odd number pairs xi and yi, $1 \leq i \leq a$. Selected numbers. Should be pair wise co-prime with each other, consist the size of l-bits.
2. Compute $S=\prod i=1^{a} ni$, where ni=xi * yi
3. Choose an orthogonal matrix K, with its dimension as 4, in ZS. Follow below approach to choose K - Randomly choose a matrix in the search space of ZS. Check if it is following the orthogonality property i.e. $K.K^T = K^T.K = I$ or $K^{-1} = K^T$, then search is completed, otherwise repeat until an orthogonal matrix is found.
4. Compute the transpose of matrix K - $K^T \leftarrow$ transposes (K) mod ZS
   Note: From matrices orthogonality property, $K^{-1} = K^T$
5. Orthogonal matrix K will act as symmetric key in our cryptosystem.

Algorithm 2 Enc_Procedure (M, ni, S, K, K^T)-

1. Consider plaintext as M.
2. Choose an integer R randomly, where R € ZS and R ≠M.
3. Create a matrix Y with its dimension as (a×3), where each row consists of single occurrence of M and rest of two occurrences as R.
4. Employ CRT to obtain solution of simultaneous equations - $\alpha j(1 \leq j \leq 3) \equiv Yij \bmod ni$ where, $1 \leq i \leq a$.
5. Use Coppersmith-Winograd algorithmic procedure [16] for below step of matrix multiplication computation.
6. Obtained cipher text $C = K^T \times d(M, \alpha1, \alpha2, \alpha3) \times K$ where, d (M, α1, α2, α3): denotes the diagonal matrix with diagonal elements as parameters.

Algorithm 3 Dec_Procedure(C, S, K, K^T)-

1: Compute plaintext as, $P = K \times C \times K^T$
2: M ← [P] 11

Algorithm 4 Refresh Procedure(S)-

1: Refresh the symmetric key, which is an orthogonal matrix as-
K ← randortho (t)

Here, randortho () is a randomized function generating new K of dimension t randomly in the space of ZS.

Analysis of Proposed algorithm:

Some observations and analysis of our proposed scheme is presented as below:-
Correctness of Decryption algorithm – we observe that, Dec_Procedure(C, S, K, K^T)=>$K \times C \times K^T$, Since we know the property of an orthogonal matrix as, $K.K^T=K^T.K=I$ or $K^{-1} =K^T$
So, Dec_Procedure(C, S, K, K^T) =>$K \times K^T \times d(M, \alpha1, \alpha2, \alpha3) \times K \times K^T$

- $I \times d(M, \alpha1, \alpha2, \alpha3) \times I$
- $d(M, \alpha1, \alpha2, \alpha3)$
- [p]11
- M

Homomorphic Properties

Our proposed Fully Homomorphic encryption scheme is satisfying both multiplicative as well as additive Homomorphic properties.

- **Multiplicative Homomorphic property**

  Consider C1, C2 are cipher texts corresponding to plaintexts M1, M2.

  $C1=K^T \times d(M1, \alpha1, \alpha2, \alpha3) \times K$

  $C2= K^T \times d(M2,\alpha'1, \alpha'2, \alpha'3) \times K$

  $Now,C1 \times C2=K^T \times d(M1,\alpha'1,\alpha'2,\alpha'3) \times K \times K^T \times d(M2,\alpha'1,\alpha'2,\alpha'3) \times K=K^T \times d(M1,\alpha1,\alpha2,\alpha3) \times I \times d(M2, \alpha'1,\alpha'2,\alpha'3) \times K$

  $= K^T \times d(M1,\alpha1,\alpha2,\alpha3) \times d(M2, \alpha'1,\alpha'2,\alpha'3) \times K = K^T \times d(M1 \times M2), \alpha"1, \alpha"2, \alpha"3) \times K$

  $Now, Dec\_Procedure(C,S,K,K^T) => K \times K^T \times d((M1 \times M2), \alpha"1, \alpha"2, \alpha"3) \times K \times K^T$

  $=> I \times d((M1 \times M2), \alpha"1, \alpha"2, \alpha"3) \times I$

  $=> \times d((M1 \times M2), \alpha"1, \alpha"2, \alpha"3)$

  $=> \times (M1 \times M2)$

- **Additive Homomorphic property**

  Consider, C1, C2 are cipher texts corresponding to plaintexts M1, M2.

  $C1=K^T \times d(M1, \alpha1, \alpha2, \alpha3) \times K$

  $C2= K^T \times d(M2, \alpha'1, \alpha'2, \alpha'3) \times K$

  $Now, C1+C2= (K^T \times d(M1, \alpha1,\alpha2,\alpha3) \times K)+(K^T \times d(M2,\alpha'1,\alpha'2,\alpha'3) \times K)$

  $=K^T \times d((M1+M2), \alpha"1, \alpha"2, \alpha"3) \times K$

  $Now, Dec\_Procedure(C,S,K,K^T) => K \times K^T \times d(M1+M2), \alpha"1, \alpha"2, \alpha"3) \times K \times K^T => I \times d((M1+M2), \alpha"1, \alpha"2, \alpha"3) \times I => d((M1+M2), \alpha"1, \alpha"2, \alpha"3)$

  $=> (M1+M2)$

Example: Algorithm 1: KeyGen (l, a)-

1. Randomly choose a =2 then 2a=2.2=4 odd numbers and two pairs (x1,y1) and (x2,y2), choose (5,7),(11,13)
2. Compute $S = \prod^a$ i=1 ni where ni =xi * yi => n=n1 * n2
   n1=5*11=55, n2=7*13=91. Therefore
   S=55*91=5005 Zs limit is {0, 1….5004}
3. Choose Orthogonal Matrix K, with its dimension as 4, in Zs

$$k= \begin{pmatrix} 0.28 & -0.26 & 0.07 & 0.09 \\ 0.09 & -0.3 & 0.04 & 0.17 \\ -0.01 & 0.05 & 0.9 & -0.07 \\ 0.27 & 0.89 & -0.02 & 0.34 \end{pmatrix}$$

   4 X4 Orthogonal matrix property: $K.K^T=I$, $K^{-1}=K^T$
4. Compute K^T of matrix K
5. Orthogonal matrix K will act as symmetric key in our cryptosystem.

Algorithm 2: Enc_Procedure (M, ni, S, K, K^T)-

1. Consider plaintext as M=257
2. Choose an integer R randomly, R=291 where R € ZS and R ≠M.
3. Create a matrix Y with its dimension as (a×3), where each row consists of single occurrence of M and rest of two occurrences as R.

$$Y= \begin{pmatrix} 291 & 257 & 291 \end{pmatrix}$$

291 291 257 2X3

291 mod 55 257 mod 55 291 mod 55----- row1
291 mod 91 257 mod 91 291 mod 91----- row2

4. Employ Chinese Remainder Theorem (CRT) [24] to obtain solution of simultaneous equations –

αj(1 ≤j≤ 3) ≡ Yij mod ni where, 1 ≤ i ≤a.

291 mod 55=α1 257 mod 55= α2 291 mod 55= α3
291 mod 91=α1 257 mod 91= α2 291 mod 91= α3
Find α1, α2, α3 values, using CRT, we get α1=291, α2=236 α3=312

5. Use Coppersmith-Winograd algorithmic procedure for below step of matrix multiplication computation.

$C = K^T \times d(M, \alpha1, \alpha2, \alpha3) \times K$

$$\begin{pmatrix} K^T X0\ 291 \end{pmatrix} \begin{pmatrix} 257\ 0\ 0\ 0 \\ 0\ 0X \end{pmatrix} \begin{pmatrix} K \end{pmatrix}$$

Constraints:

i) Both m and C must be co primes with each other
ii) (a-1) must be divisible by all prime factors of m
iii) New generated key series should be <m with uniqueness

C=11, m=23 both co-primes (satisfy constraint (i), then choose a=47, (a-1) satisfy constraint (ii) then possible m= {0 ….22}, the generated new key series is unique and equal gap size key and <m (satisfy constraint(iii))

X1= (47X0+11) mod 23

X2= (47X1+11) mod 23

.Xn= (47Xn-1+C) mod 23

0 0 236 0
0 0 0 312

6. Obtained cipher text $C = K^T \times d(M, \alpha1, \alpha2, \alpha3) \times K$ where, d(M, α1, α2, α3): denotes the diagonal matrix with diagonal elements as parameters

Algorithm 3: Dec_Procedure(C, S, K, K^T)-
1. Compute plaintext as, $P = K \times C \times K^T$
2. M ⟸ [P] 11

Algorithm 4: Refresh Procedure(S)-

1: Refresh the symmetric key, which is an orthogonal matrix as-

K' ⟸ randortho(t)

Here, randortho() is a randomized function generating new K of dimension t randomly in the space of ZS.

Xk= (aX (k-1) +C) mod m

Here X is new key, k=1….m, C, m are Constants

Table iii: Execution time of Keygen(), Encry() and Decry()

Size of S(in bits)	Key gen()	Engcry()	Decry()
16-bits	0.332 Sec.	0.173 Sec.	0.096 Sec.
32-bits	28.41Sec	55.20 Sec.	15.0 Sec
50-bits	59.85 Sec.	71.19 Sec	22.40 Sec.
64-bits	1118.42 Sec.	204.91 sec.	102.45 Sec.

Table (iii) the various Homomorphic Encryption methods and their performance and comparisons various items with existing approaches respectively. The following figure (ii) shown the x -axis represents size of s (in bits) of plaintext and Execution time (in seconds) for encryption, decryption and key refresh at y-axis. It shows the size is proportional to the execution time.

RESULTS& DISCUSSION

Proposed FHE theme is light-weight in nature and utilizes matrices machine operations, that ar light-weight in nature as compare to operating with polynomial computations. projected FHE theme is parallelizable. As our projected FHE theme utilizes matrices machine operations whereas encoding also as coding, that offer the advantage of activity outer product matrix vector multiplication that's abundantly parallelizable. Machine quality improvement, we tend to ar utilizing Coppersmith-Winograd recursive procedure [16] for matrix operation computation in encoding step. This methodology offers a considerably improved matrix operation machine quality as O ($n^{2.376}$).

Performance of SHE

The performance of Somewhat Homomorphic Encryption is based on the time taken to generate Key [17] [27]. Estimated time for Encryption, Decryption and degree of polynomials with respect to size of plaintext [18][19][28]. The following Table (iv) represent the comparison of existing methods with newly proposed method with various parameters as shown below.

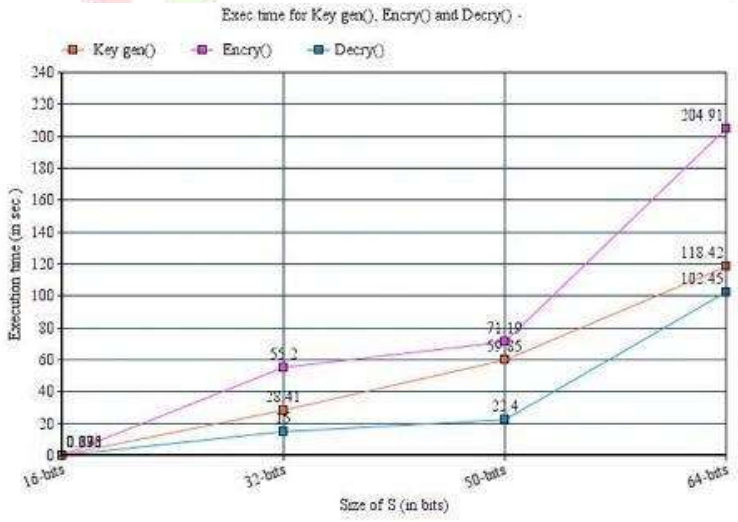

Figure ii: Execution time for encryption, decryption and refreshing a key.

Table iv: Performance: SWHE

Item of Comparison	**DGHV SHE**	**CMNT SHE**	**GV SHE**	**Proposed SHE**
Compactness	No	Yes	Yes	Yes
Dimension	2048 (8000,000-bit integers)	8192 (3,200,000-bit integers)	32768 (13,000,000-bit integers)	33024
KeyGen	1.25Sec	10 Sec	95 Sec	59.85 Sec.
Enc amortized	.060 Sec	.7 Sec	5.3 Sec	71.19 Sec.
Mult/ Dec	.023 Sec	.12 Sec	.6 Sec	22.40 Sec.
Degree	~200	~200	~200	~200
Security base	Approximate GCD	PAGCD(Error-free approximate GCD	Two-element PAGCD	Reduced Approximate GCD

Performance of FHE

The performance of Fully Homomorphic Encryption is based on the time taken to generate Key[20], Estimated time for Encryption, Decryption and degree of polynomials [21][22][23] with respect to size of plaintext. The following table (v) shows the various Homomorphic Encryption methods and their performance.

Table v: comparison of proposed algorithm with existing methods.

Item of Comparison	**DGHV SHE**	**CMNT SHE**	**GV SHE**	**Proposed FHE**
Compactness	No	Yes	Yes	Yes
Dimension	2048	8192	32768	33024
KeyGen	40Sec	8 Min	2 hours	7 min
PK Size	70MByte	285 MByte	2.3GByte	325MBytes
Enc amortized	.060 Sec	0.7 Sec	5.3 Sec	6.5 Sec
Mult/Dec	.023 Sec	.12 Sec	.6 Sec	.8 sec
Recrypt	31 Sec	3 Min	30 Min	20 min
Degree	~200	~200	~200	~200

Security base	Approximate GCD	PAGCD(Error-free approximate GCD	Two-element PAGCD	Reduced Approximate GCD

Performance of Computations:

Table VI: Comparison of various schemes over computations

Item of DGHV Comparison	CMNT SHE	GV SHE	Proposed	
			SHE	FHE
Modulus	**257**	**8209**	**65537**	**65793**
Time for addition(ms)	0.7	0.7	2.9	1.9
Time for multiplication (ms)	39	38	177	42

CONCLUSION

In this paper, associate degree economical, conceptually easy, and semantically secure and risk for sensible applications like Homomorphic cryptography principles at cloud users is projected and reduced the key sizes and time for cryptography and recrypt operations. In future we tend to proposes this technique will implement for uneven crypto system wherever we will use the 2 keys one for cryptography and alternative for coding that may additional reliable and scalable with high security measures.

REFERENCES

[1] W. Liu, "Research on cloud computing security problem and strategy," 2012 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet), Yichang, 2012, pp. 1216-1219.doi: 10.1109/CECNet.2012.6202020

[2] Kevin Hamlen, Latifur Khan, Murat Kantarcioglu, Bhavani Thuraisingham, The University of Texas at Dallas, USA, "Security Issues for cloud computing" International Journal of Information Security and Privacy, 4(2), 39-51, April-June 2010 pp 39-51

[3] Ronald L. Rivest, Len Adleman Michael L. Dertouzos “On Data Banks and Privacy Homomorphisms” Massachusetts Institute of Technology Cambridge, Massachusetts, © 1978 by Academic Press, Inc.

[4] Dishita Dave,Rikkin Thakkar “Homomorphic Encryption In Cloud Computing” 2015 IJIRT | Volume

1 Issue 12 | ISSN: 2349 -6002.

[5] A.P.Nirmala, Dr.R.Sridaran "Cloud Computing Issues at Design and Implementation Levels-A Survey" Int.J.Advanced Networking and Applications, Volume: 03, Issue: 06 Pages 1444-1449(2012) ISSN: 0975-0290.

[6] Frederik Armknecht, Colin Boyd, Christopher Carr, KristianGjøsteen, Angela J ̈aschke, Christian A. Reuter, and MartinStrand "A Guide to Fully Homomorphic Encryption" https://eprint.iacr.org/2015/1192.pdf

[7] K. Lauter, A. L´opez-Alt, and M. Naehrig. Private computation on encrypted genomic data. In Proceedings of Progress in Cryptology - LATINCRYPT 2014, volume 8895, pages 3–27

[8] S. M. Anggriane, S. M. Nasution and F. Azmi, "Advanced e-voting system using Paillier Homomorphic encryption algorithm," 2016 International Conference on Informatics and Computing (ICIC), Mataram, 2016, pp. 338-342.doi: 10.1109/IAC.2016.7905741

[10] Maha TEBAA,Said EL HAJII , V–Agdal, "Secure Cloud Computing through Homomorphic Encryption " International Journal of Advancements in Computing Technology(IJACT) Volume5, umber16, Dec- 2013.

[11] T.ElGamal. A Public –Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. Crypto'84 pp.469-472.

[11] S.Goldwasser, S. Micali (1982). "Probabilistic encryption and how to play mental poker keeping secret all partial information". Proc. 14th Symposium

[12] J.Benaloh.Verifiable secre-ballot elections. Ph.D thesis, Yale University, Dept. of Computer Science, 1988.

[13] P.Pailliar. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. Eurocrypt'99 pp-223-238

[14] D.Benoh,E.Goh, and K.NIssim. Evaluating 2-dnf formulas on ciphertexts, In proceedings of Theory of Cryptography(TCC)'05, LNCS 3378,pages 325-341.

[15] Yuval Ishai and Anat Paskin. 2007. Evaluating branching programs on encrypted data. In Theory of Cryptography. Springer, 575–594

[16] D. Coppersmith and S. Winograd. "Matrix `multiplication via arithmetic progressions", J. Symbolic Computation, 9(3):251-280, (1990).

[17] C. Gentry "Fully Homomorphic encryption using ideal lattices" In STOC pp 169-178 ACM 2009.

[18] Y.Govinda Ramaiah,G.Vijaya kumari "Complete Privacy preserving Auditing for Data Integrity in Cloud Computing" published in 2013 12th IEEE International Conference on Trust,Security and Privacy in Computing and Communications, 2013, DOI: 10.1109/TrustCom.2013.191

[19] N. P. Smart F. Vercauteren "Fully Homomorphic encryption with relatively small key and ciphertext sizes" In Public Key Cryptography-PKC'10 Vol. 6056 of LNCS pp. 420-443 Springer 2010

[20] M. V. Dijk C. Gentry S. Halevi V. Vaikuntanathan "Fully Homomorphic encryption over the integers" Proceedings of Eurocrypt Vol. 6110 of LNCS pp. 24-43 Springer 2010.

[21] Dasgupta Smaranika, and S. K. Pal. "Design of a polynomial ring based symmetric homomorphic encryption scheme", Perspectives in Science 8 (2016), ELSEVIER: pp. 692-695.

[22] Yagisawa, Masahiro. "Fully Homomorphic Public-key Encryption Based on Discrete Logarithm Problem", IACR Cryptology e-Print Archive (2016): 54.

[23] Cheon, Jung Hee. "The polynomial approximate common divisor problem and it's application to the fully homomorphic encryption", Information Sciences-326 (2016): pp.41-58.

[24] Y.H.KuXiaoguangSun, Moore School of Electrical Engineering, University of Pennsylvania, Philadelphia, PA 19104, USA, "The chinese remainder theorem" Journal of the Franklin Institute Volume 329, Issue 1, Jan 1992, Pages 93-97,https://doi.org/10.1016/0016-032(92)900993,http://www.sciencedirect.com/science/a rticle/pii/0016003292900993.

[25] Kristin Lauter, Michael Naehrig and Vinod Vaikuntanathan "an Homomorphic Encryption be Practical?" ACM CCSW 2011 paper. https://eprint.iacr.org/2011/405.pdf