Advances in Computational Sciences and Technology
ISSN 0973-6107 Volume 10, Number 5 (2017) pp. 825-837

http://www.ripublication.com

# A New Hybrid Homomorphic Encryption Scheme for Cloud Data Security

**D.Chandravathi[†] and Dr. P.V.Lakshmi[††]**

[†]*GVP College for Degree and PG Courses, Rushikonda, Viakhapatnam-45, India.*
*Chandravathi.d@gmail.com*
[††]*GITAM University, Rushikonda, Viakhapatnam-45, India.*

**Abstract**

A new emerging trend in the field in computer technology is cloud computing. Traditionally the data was maintained and controlled in one own data centres. Today, Cloud computing provides delivery of computing as a service. Depending on the service, the huge information is shared over the network and this huge amount of data is stored in the cloud provider centre. Hence, security is a major concern for the data in the cloud. Various encryption algorithms were applied for securing the data in the cloud. But still, the data is not secure in the cloud due to various attacks on the data .Data can be stored in the cloud in an encrypted format. Hence, a new technique called Homomorphic encryption is introduced which is a good idea that allows specific operations to be performed on the encrypted data. In this paper, we propose hybrid homomorphic encryption algorithm for providing enhanced security and confidentiality of the data that is stored in the cloud. The Encryption process is carried by using RSA algorithm, which generates the cipher and this cipher is the input for Elgamal encryption process, which generates cipher and this resultant cipher is sent to the cloud provider where Homomorphic multiplicative operations are applied to the encrypted cipher .Hence, the encrypted cipher is stored in the cloud which can be accessed at any time. Since the cloud provider has the encrypted cipher no other person can be able to know which operation has been performed. Hence security and authentication is enhanced.

**Keywords:** Technology, Cloud computing, cloud provider, service, encrypted data, Homomorphic encryption, RSA, ElGamal, Cloud security, hybrid, cipher, multiplicative operations, security, authentication.

## INTRODUCTION

Most of the business applications need new ideas for implementation. Since the data centres occupy lots of space maintenance cost is a big issue. Cloud computing is a new way for storing large data and to run applications [4]. It uses the concept of multi-tenancy in which one application and many users can be customized and use the same applications along with building the application. The cloud service provider looks after the upgradations and maintenance. Cloud Computing is an Internet-based technology[1]. The utilization of cloud is growing fast with respect to increase in network bandwidth as it is reliable. Security and privacy are always major issues which are to be considered in cloud computing environment. Some of the security issues are Data integrity, Data Availability, Privacy and Confidentiality, Data location and Relocation, Storage, Backup and Recovery [5].

The cryptography goal is to provide bundle of security function that can assure the secrecy of the system. These goals can be listed under the following five categories [9]:

- **Authentication:** The identity of the sender and the receiver should be verified before sending the message.
- **Secrecy or confidentiality:** Only authenticated users can interpret the message and no one else can use it.
- **Integrity:** The content of the communicated data is assured to be free from any type of modification.
- **Service Reliability and Availability:** Since intruders affect their availability and type of service to their users. System should provide a way to grant their users the quality of service they expect [14].
- **Non-Repudiation:** This function implies that neither the sender nor the receiver can't deny that have sent a certain message.

### Types of Cryptographic Algorithms:

There are several ways of classifying cryptographic algorithms. They are:

- **Secret Key Cryptography (SKC):** This type of cryptography uses a single key for both encryption and decryption. It is also called symmetric encryption which is primarily used for privacy and confidentiality.
- **Public Key Cryptography (PKC):** This type of cryptography uses one key for encryption and another for decryption. It is also called asymmetric encryption which is primarily used for authentication, non-repudiation, and key exchange.
- **Hash Functions**: This type of cryptography uses a mathematical transformation to irreversibly "encrypt" information, providing a digital fingerprint. It is primarily used for message integrity.

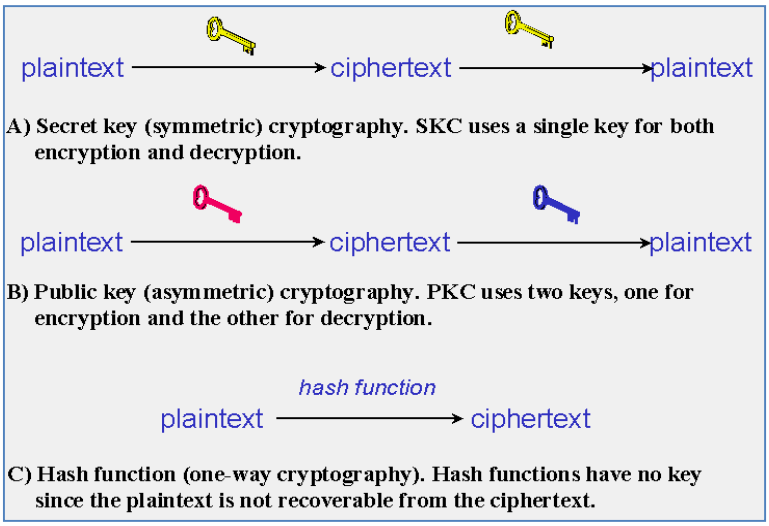


**Fig 1**

**RSA Scheme:**

The RSA is an asymmetric public key cryptosystem. We can define is any number of pairs of algorithms (E, D) on the same set of values[11][12]. Here, E is the public encryption algorithm and D is the private decryption algorithm.

The RSA cryptosystem uses modular exponentiation. The modulus 'N' is the product of two large prime's P and Q, Public key and private key are generated as:

$$E = D^{-1} \bmod Ø(N)$$

The encryption operation is performed using the public key 'N' and 'E' as follows:

$$C = M^E \pmod N$$

where M is the plaintext such that 0<M<N and C is the ciphertext which can be decrypted using the private key 'N' and ' D' as follows:

$$M = C^D \pmod N$$

**Public Key Encryption Scheme:**

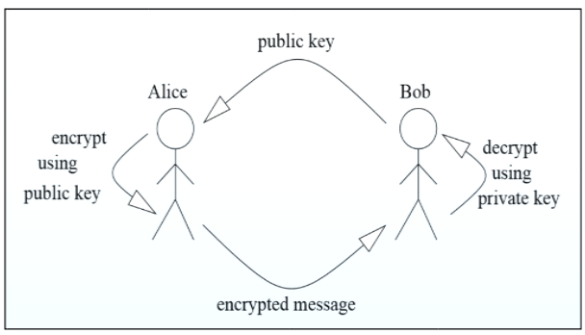


**Fig 2**

**Elgamal Scheme:**

The ElGamal encryption system is an asymmetric key encryption algorithm for public-key cryptography which is based on the Diffie–Hellman key exchange[2]. It was developed by Taher Elgamal in 1984. ElGamal is randomized, the trivial attack does not work, which is one of its advantages over RSA. Also, ElGamal uses a smaller key length when compared to RSA.

**Key Generation**

Participant 'A' generates the public/private key pair

1. Generate large prime 'P' and generator 'g' of the multiplicative Group Z∗p of the integers modulo P.
2. Select a random integer 'A', $1 \leq A \leq P - 2$, and compute gA mod P.
3. A's Public key is (P, g, gA); A's Private key is A.

**ElGamal Cryptosystem - Encryption Procedure**

Participant B encrypts a message M to A

1. Obtain A's authentic public key (P, g, gA).
2. Represent the message as integers M in the range {0, 1, . . . , P − 1}.
3. Select a random integer K, $1 \leq K \leq P - 2$.
4. Compute t1 = gK mod P and t2 = M ∗ (gA)K.
5. Send ciphertext C= (t1,t2) to A.

**ElGamal Cryptosystem - Decryption Procedure**

Participant A receives encrypted message 'M' from B

1. Use private key 'A' to compute (t1P−1−A) mod P. Note: t1P−1−A = t1−A = A−AK.
2. Recover m by computing (t1−A) ∗ t2 mod P.

**Cloud Computing:**

Cloud is a place where one can access applications and services, and where the data can be stored securely [1]**.** The reasons for using cloud are:

1. Managing or maintaining the data is easier.
2. The size is infinite.
3. With an Internet connection accessing the cloud-based applications and services from anywhere with a device can be done.

The information technology model for computing is composed of the components like hardware, software, networking, and services. Hence, it is necessary to enable the development and delivery of cloud services via the Internet or a private network. There are users in the cloud. Cloud Provider and Cloud User are the prominent actors in Cloud Computing. Cloud Provider provides cloud services. Organizations, educational institutes, individuals who utilize the cloud services are the cloud users. Hence, security, confidentiality and visibility with respect to the cloud providers is much essential. The main aspect is to protect the data from hacking.

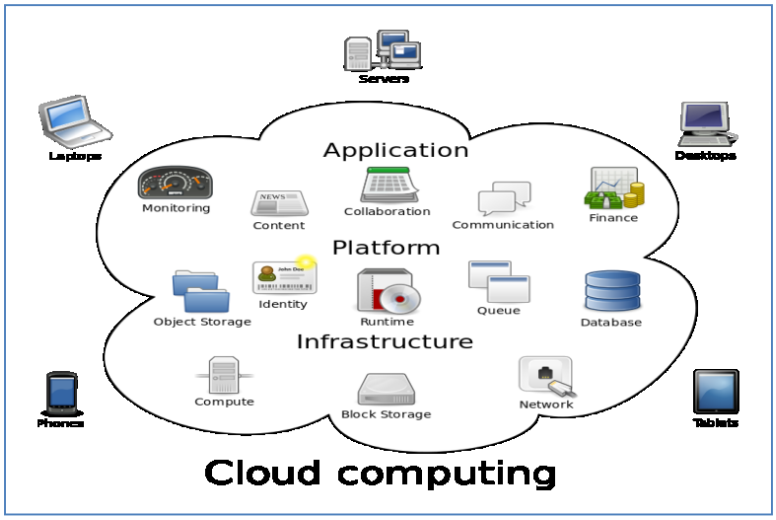


**Fig 3**

**Homomorphic Encryption**

Homomorphic encryption is one of the emerging techniques that are in use in cloud data security. The process is carried in the cloud service provider. The plain text is encrypted using either of the encryption algorithms. The cipher is generated, which is sent to the cloud for storing. This cipher can be hacked by the intruders. Hence it is not safe. Homomorphism encryption performs encryption on the encrypted data which is in the cloud. It performs operations on the encrypted data and the resultant is stored in the cloud so that none of them knows what exactly the data is.

Homomorphic encryptions are of either semi/partitially homomorphic or fully homomorphic. Partially homomorphic encryption uses multiplicative/additive operations on the encrypted data. Fully homomorphic encryption uses both multiplicative and additive operations on the encrypted data. RSA and Elgamal are both asymmetric cryptographic algorithms. Both the algorithms satisfy only multiplicative homomorphic property .Hence they are semi/partially homomorphic.

**Partially homomorphic encryption:**

The idea of performing simple computations on encrypted messages was first introduced by Rivest, Adleman, and Dertouzous . The original motivation for these homomorphisms was to allow for an encrypted database to be stored by a third party and to allow the owners and other authorized people to perform calculations with the data without decrypting it. A multiplicative homomorphic cryptosystem has an encryption function E that satisfies the following property: E(M1) * E(M2) = E(M1*M2) where M1 and M2 are plain text messages Some of the applications of homomorphic encryption are: Cloud computation, Electronic voting, Data mining, Financial transactions, Electronic cash, Medical records.

If the RSA public key is modulus $m$and exponent $e$, then the encryption of a message $x$is given by $\mathcal{E}(x) = x^e \bmod m$. The homomorphic property is then

$$\mathcal{E}(x_1) \cdot \mathcal{E}(x_2) = x_1^e x_2^e \bmod m = (x_1 x_2)^e \bmod m = \mathcal{E}(x_1 \cdot x_2)$$

In the ElGamal cryptosystem, 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 $\mathcal{E}(m) = (g^r, m \cdot h^r)$, for some random $r \in \{0, \ldots, q-1\}$. The homomorphic property is then

$$\mathcal{E}(x_1) \cdot \mathcal{E}(x_2) = (g^{r_1}, x_1 \cdot h^{r_1})(g^{r_2}, x_2 \cdot h^{r_2})$$
$$= (g^{r_1+r_2}, (x_1 \cdot x_2)h^{r_1+r_2}) = \mathcal{E}(x_1 \cdot x_2).$$

**Proposed Method:**

In our proposed work, Hybrid encryption process is used for encryption of data in the cloud. The process is as follows: The plain text is encrypted using RSA algorithm and cipher C1 is generated. This cipher C1 is treated as the plaintext for the Elgamal algorithm where reencryption is carried to generate the cipher C2.This encrypted cipher C2 is the encrypted data that is to be stored in the cloud. On this encrypted cipher C2 homomorphic Elgamal multiplicative operations are carried to get the resultant cipher C, which is stored in the cloud.The client uses hommorphic operations to verify C in the cloud. Since hybrid encryption is used high security and authentication is provided .Thought time taken for key generation and encryption is increased as it has to do many computations on RSA and Elgamal followed by Homomorphic encryption, Security is enhanced.

**Overview of the scheme:**

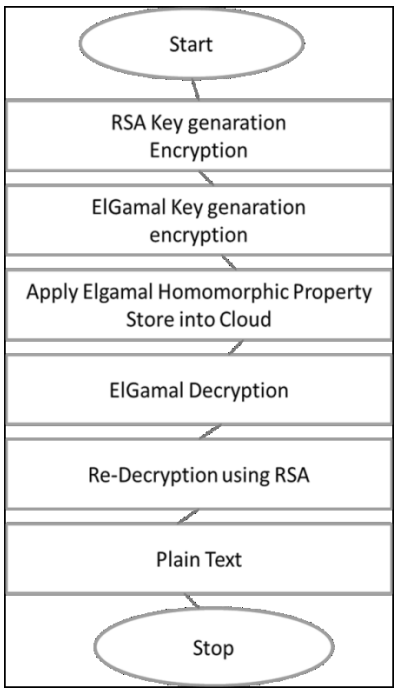


**Fig 4**

**Algorithms:**

**RSA ALGORITHM**

The RSA algorithm includes three phases**:**

- Key generation phase
- Encryption phase and
- Decryption phase

Key generation RSA involves a public key and a private key. The public key is known to everyone and is used for encrypting messages. Messages encrypted with the public key can be decrypted using the private key in a reasonable amount of time. The keys for the RSA algorithm are generated in the following way:

**Step 1:** Firstly two prime number 'P' and 'Q' are chosen. For security purposes, the integers 'P' and 'Q' should be chosen at random.

**Step 2:** Then value of 'N' is computed by using N= P *Q. 'N' is used as the modulus for both the public and private keys. Its length, usually expressed in bits, is the key length.

**Step 3**: Compute $\varphi(N) = \varphi(P)*\varphi(Q) = (P - 1)(Q - 1)$, where 'φ' is Euler's totient function.

**Step 4**: Choose an integer 'E' such that $1 < E < \varphi(N)$ and $\gcd(E, \varphi(N)) = 1$; i.e. 'E' and $\varphi(N)$ are co primes. 'E' is released as the public key exponent.

**Step 5**: Determine 'D' as $D-1 \equiv E \pmod{\varphi(N)}$, i.e., 'D' is the multiplicative inverse of E (modulo $\varphi(N)$). This is more clearly stated as solve 'D' which is calculated as

$$D*E \equiv 1 \pmod{\varphi(N)}$$

where 'D' is the private key exponent.

**Step 6**: By construction, $D*E \equiv 1 \pmod{\varphi(N)}$. The public key consists of the modulus 'N' and the public (or encryption) exponent 'E'. The private key consists of the modulus 'N' and the private (or decryption) exponent 'D', which must be kept secret. P, Q and $\varphi(N)$ must also be kept secret because they can be used to calculate D.

**RSA Encryption**

Receiver transmits his public key (N, E) to Sender and keeps the private key secret. Sender then wishes to send message M to Receiver. He then computes the ciphertext 'C' corresponding to

$$C \equiv M\text{^}E \pmod{N}$$

Sender then transmits 'C' to Receiver.

**Re-Encryption Using ElGamal**

**Key Generation:**

1. Input the value of 'P' , which is used in RSA Algorithm.
2. Choose a generator number A .

Test A for generator :

- It must be between 1 and P-1.
- Calculate Ø = P-1
- Find the all factors of Ø {f1,f2,….,fn} – { 1 }
- Find {Q1,Q2,…..,Qn} where
  - Qi = fi for the redundant factors
  - $Qi = fi^{freq}$
- Generate the number if and only if
  - $wi = A^{Ø/Qi} \bmod E$ where P < >1 , for all Qi

3. Choose an integer (X) < (P-2) as secret number.
4. Compute (D) where $D = A^X \bmod P$.
5. Determine the public key (P, A, D) and the private key (X).

**Elgamal Encryption**

1. Collect the public key (P , A , D ) from the receiver .
2. Select an integer 'k' such that : $1 < k < P-2$

3. Represent the plaintext as an integer 'M' where $0 < M < P-1$
4. compute (y) as follows :

$$y = A^K \bmod P$$

5. Compute z as follows :
   $z = (D^K * M) \bmod P$
6. Generate the ciphertext (C) as follows :
   $C= ( y , z )$.

**Homomorphic Property:**

ElGamal encryption scheme has a homomorphic property. Given two encryptions as

$(c_{11}, c_{12}) = (g\text{^}r_1 , M_1*y\text{^}r_1)$

$(c_{21}, c_{22}) = (g\text{^}r_2 , M_2*y\text{^}r_2 )$

where r1 , r2 are randomly chosen from {1, 2,...., Q -1 } and $M_1$, $M_2$ one can compute

$$(c_{11}, c_{12})\ (c_{21}, c_{22}) = (c_{11}*c_{21}, c_{12}*c_{22})$$
$$= (g^{r1}* g^{r2}, (M_1*y^{r1})*( M_2*y^{r2}))$$
$$= (g^{r1+r2} ,( M_1 M_2)* y^{r1+r2})$$

The resulted cipher text is an encryption of $M_1$ $M_2$.

**RESULTS**

**a) Key Generation:**

**Table 1**

| File size ( Bytes) | Key Generation Time (msec) |
|---|---|
| 2356 | 8 |
| 6344 | 8 |
| 365 | 8 |
| 4230 | 8 |
| 19314 | 8 |
| 21850 | 8 |
| 34586 | 8 |
| 47253 | 8 |

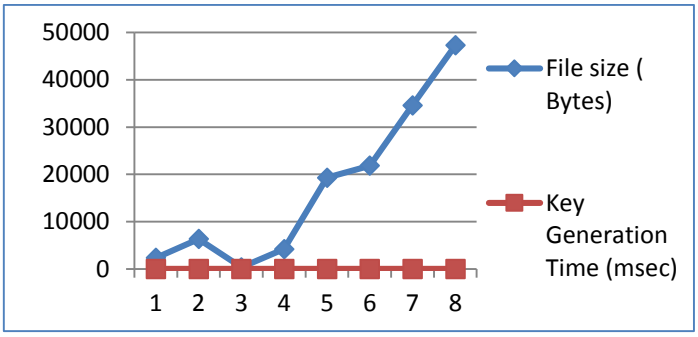


**Fig 5**

**b) Encryption Time:**

**Table 2**

| File size ( Bytes) | Encryption time(msec) |
|---|---|
| 2356 | 56 |
| 6344 | 267 |
| 365 | 5 |
| 4230 | 145 |
| 19314 | 18694 |
| 21850 | 25193 |
| 34586 | 74694 |
| 47253 | 154768 |

**Time Analysis for Encryption**:

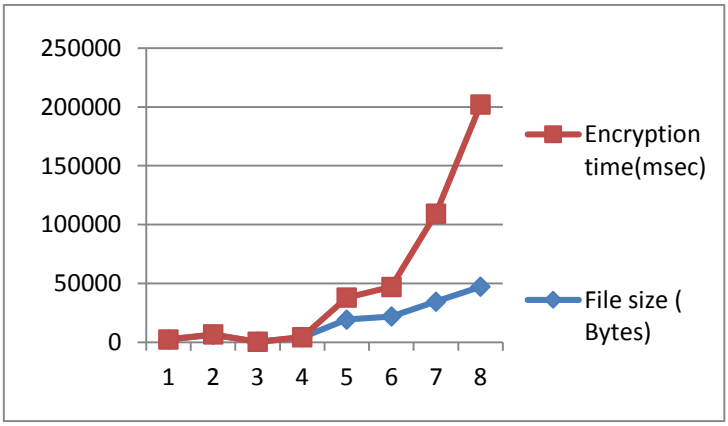


**Fig 6**

**c) Decryption Time:**

**Table 3**

| File size ( Bytes) | Decryption time (msec) |
|---|---|
| 2356 | 33 |
| 6344 | 166 |
| 365 | 4 |
| 4230 | 73 |
| 19314 | 178 |
| 21850 | 1354 |
| 34586 | 804 |
| 47253 | 3659 |

**Time Analysis for Decryption:**

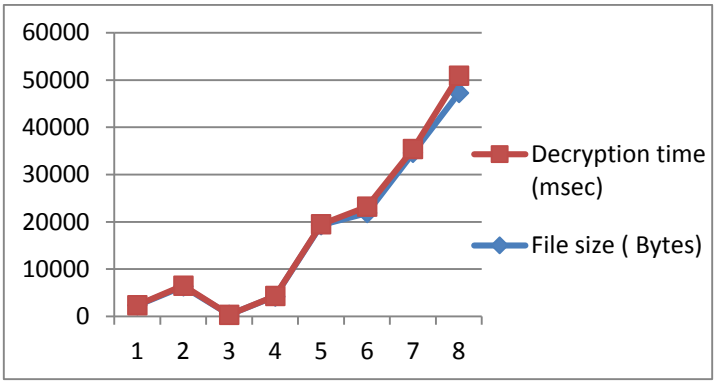


**Fig 7**

**Total Execution Time:**

**Table 4**

| File size ( Bytes) | Total time (msec) |
|---|---|
| 2356 | 97 |
| 6344 | 441 |
| 365 | 17 |
| 4230 | 226 |
| 19314 | 18880 |
| 21850 | 26555 |
| 34586 | 75506 |
| 47253 | 158435 |

**Total Execution Time Analysis:**

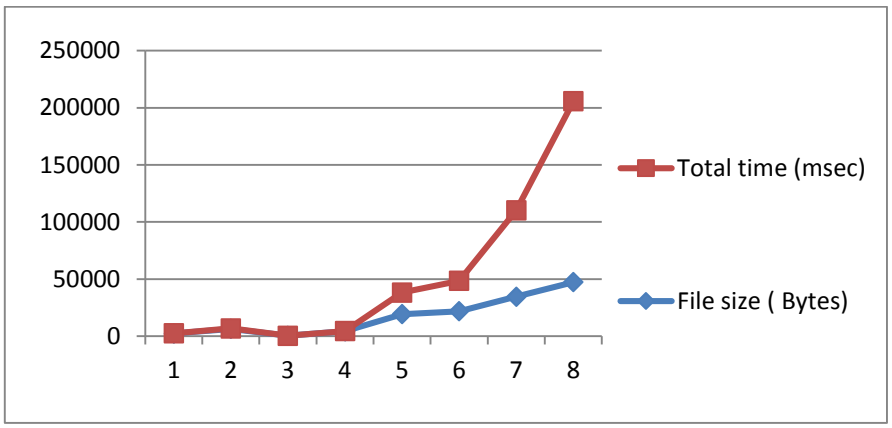


**Fig 8**

## CONCLUSION

Homomorphic encryption technique is used to provide security on cloud. Homomorphic encryption is a new concept of security which enables providing results of calculations on encrypted data without knowing the raw data on which the calculation was carried out, with respect of the data confidentiality. In this proposed paper RSA and ElGamal algorithms for homomorphic encryption .Hence data is very secure and confidential. Due to encryption and reencryption process of RSA and Elgamal schemes followed by homomorphic operations, encryption time is increased but provides more security. Hence data is much secure in the cloud by applying hybrid encryption scheme.

## REFERENCES

1. 'Data Storage Security Using Partially Homomorphic Encryption in a Cloud ', Sunanda Ravindran , Parsi Kalpana International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 4, April 2013 ISSN: 2277 128X. \
2. Faster RSA Algorithm for Decryption Using Chinese Remainder Theorem**,** G.N. Shinde1 and H.S. Fadewar, ICCES, vol.5, no.4, pp.255-261.
3. C. Lamprecht "Investigating the efficiency of Cryptographic algorithm in Online Transaction"ISN 1473-804X online, 1473-8031 I.J. of Simulation Vol.7, No.2.
4. AlexanderMay, "Cryptanalysis of Unbalanced RSA with Small CRT-Exponent", CRYPTO 2002, LNCS 2442, pp 242-256, 2002.
5. JohannnesBlomer,Martin Otto, "a newCRT-RSA Algorithm Secure Against Bellcore", CC'03, October 27-30,Washington, DC, USA.

6. Craig Gentry, A Fully Homomorphic Encryption Scheme, 2009.http://crypto.stanford.edu/craig/craig-thesis.pdf.
7. Understanding Homomorphic Encryption http://en.wikipedia.org/wiki/Homomorphic_encryption.
8. Homomorphic Encryption Applied to the Cloud Computing Security Maha TEBAA, Saïd EL HAJJI, Abdellatif EL GHAZI.
9. Vic (J.R.) Winkler, "Securing the Cloud, Cloud Computer Security, Techniques and Tactics", Elsevier.
10. R. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public key cryptosystems. Communications of the ACM, 21(2):120-126, 1978.
11. A Fully Homomorphic Encryption Implementation on Cloud Computing Shashank Bajpai and Padmija Srivastava *Cloud Computing Research Team, Center for Development of Advanced Computing [C-DAC], Hyderabad.*
12. Fully homomorphic encryption equating to cloud security: An approach Bhabendu Kumar Mohanta, Debasis Gountia, IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 9, Issue 2 (Jan. - Feb. 2013), PP 46-50.