ISSN (Print) : 0974-6846
Indian Journal of Science and Technology, Vol 9(8), DOI: 10.17485/ijst/2016/v9i8/87964, February 2016 ISSN (Online) : 0974-5645

Ensuring Confidentiality of Cloud Data using Homomorphic Encryption

K. Suveetha[1] and T. Manju[2]

[1]Computer Science and Information Security,
[2]Thiagarajar College of Engineering GST Road, Thiruparankundram, Madurai - 625015, Tamil Nadu, India;
suveethak@gmail.com, tmanju@tce.edu

Abstract

Background/Objectives: To operate on encrypted data which will provide confidentiality and data privacy to cloud users. **Method/Statistical Analysis:** We have applied paillier algorithm's multiplicative property of homomorphic encryption to calculate total interest on encrypted banking dataset. **Findings:** Confidentiality of cloud user's data is achieved through Paillier homomorphic encryption. Cloud service provider can execute computations on ciphered user's data stored in cloud data center without knowing the secret key. Using RSA and Paillier algorithms performance is evaluated considering both encryption and decryption time. **Improvements/Applications**: Improvements can be done by optimizing the algorithm, which reflects changes in encryption and decryption time based on file size chosen. Banking application was selected, which included sensitive information of bank customers.

Keywords: Cloud Computing, Homomorphic Encryption algorithms, Cloud Computing issues.

1. Introduction

Nowadays, the need of resources is increasing drastically to process and maintain huge data in various fields. But to own these resources and to maintain them is a difficult task. Cloud computing is the recent efficient and effective technology which provides these resources on demand with minimal management effort1. Some of the issues in cloud computing are security and privacy, performance, reliability, availability, scalability, elasticity, interoperability, portability, resource management, scheduling, energy consumption, virtualization, bandwidth cost and what to migrate1. Cloud Service models are PaaS (Platform-as-a-Service), IaaS (Infrastructure-as-a-Service), SaaS (Software-as-a-Service (SaaS). Cloud Deployment models are public cloud, private cloud , community cloud, hybrid cloud[1].

2. Challenges and Security Issues in Cloud

Security is the main factor in storing the sensitive information in cloud. The main security components are Confidentiality, privacy, integrity, availability[1]. It is a tedious process to store the banking in cloud. The user's details are the sensitive information and high security should be provided. The following are the main security threats in storing and exchanging banking data in cloud.

2.1 Confidentiality

Confidentiality is the process of keeping users sensitive information stored in the cloud private unless the user will give permission to release. It should be maintained by the authorized persons and cloud service provider. The

* Author for correspondence

users store the data in the encrypted form, we can convert the plaintext to ASCII code to ensure confidentiality[2]. When data is transmitted via internet, confidentiality should be maintained[3]. While storing and retrieving the data, key management issues should be solved.

2.2 Integrity

Integrity is the process of ensuring that the user's sensitive information are captured or provided is the original representation of the information and has not been modified. Since many participants (customers, any authorized persons and cloud service providers) are involved in the cloud based data exchange, any modification can occur due to the participants intentionally or unintentionally. Integrity can also be maintained by the use of digital signature and Message Authentication Code (MAC)[3]. Hence integrity is a big issue to banking data in cloud.

2.3 Access Control

The unauthorized usage of sensitive information (Customer data) in cloud can be prevented with the help of access control policies. Many organizations allow the users who have previously registered with their valid credentials can only access the resources. The access control policies vary depends on their role. The access control policies outsourced in cloud should not be leaked out.

2.4 Availability

The sensitive data of cloud users stored in the cloud can be accessed by authorized users only at any time. Suppose if the resources are not available to the customers from other place, he cannot view the sensitive information. The unavailability may occur due to the poor internet connection and also theft of information by the unauthorized users[3].

2.5 Data Ownership

Most of the unauthorized users get access to the sensitive information of bank data due to the missing of the owner's identity in the encrypted banking data. The usage of cryptographic algorithms, by issuing public and private key make it difficult for the attackers to gain access.

2.6 Privacy

The authorized users can access their sensitive data at any time and can do any operations like read, write and update etc and also determine how their sensitive data is shared with others. It involves maintaining confidentiality.

2.7 Authentication

The sensitive data should be accessed only by the authorized users. The credentials provided by the users must match with the stored credentials of the users in the authentication process. If the credential details are leaked out, unauthorized users can get a chance to access the sensitive data of authorized users.

3. Cryptographic Methods

When user's data is uploaded in the cloud, it has to be secured. So it is encrypted by users, using various cryptographic algorithms, which is stored in the cloud data storage. Cryptography is a technique, which provides security by encrypting the messages that can be non readable. The plain text can be understood by any. The encoded message using cryptographic techniques called as cipher text. Data can be stored in any database in encrypted form. If computations needed to perform on encrypted data, then it is necessary to decrypt those data but the decrypted data are not secure any more, thus, a new cryptosystem was proposed that permits computation on the encrypted data. This concept is called privacy homomorphism[4]. However, decryption is not performed; the result obtained is same as computations performed on plaintext. While handling encrypted data, additions and multiplications on plaintext can be performed by using homomorphic encryption[5,6].

There are three techniques **1**) Symmetric Key Cryptography **2**) Asymmetric Key Cryptography **3**) Hash function Cryptography.

- Symmetric Key Cryptography:

In Symmetric Cryptography, for encrypting and decrypting the message, the same secret key is used. A secret key used in symmetric cryptography can be a string or text. The process involved in symmetric technique is shifting each letter by a number of places in the alphabet.

- Asymmetric Cryptography:

In Asymmetric Cryptography, secret key is used for encrypting the message but the same secret key is not used for decrypting the same message[7]. The problem by using the secret key is, it is exchanged over internet, where intermediate unauthorized users can hack them. The public key will be available to anyone to send the message, secret decryption key is kept confidential, only the authorized person who have the secret decryption key

can view the message. Confidentiality and integrity can be achieved. A problem with asymmetric encryption is, it is slower than symmetric encryption. It requires more processing power for encryption and decryption process.

- Hash Function Cryptography:

In hash function cryptography, it generates a fixed-size blocks of data. The output of the hash function is called Message Digest (MD). Any simple change in hash function after it was generated, it will affect second value of the hash function. Any slight into the original text will reflect a vast differences in hash values. It solves problem of integrity of the messages. The most widely used hash function techniques are: MD5, SHA1.

4. Encryption Methods

4.1 Attribute Based Encryption

An Attribute Based Encryption makes user sensitive data more secure. Attribute-Based Encryption (ABE) is public-key based, where encryption and decryption is possible based on user attributes[8]. The cipher text and private key of a user are depends on the attributes. Cipher text is decrypted only if the set of attributes of the user key and the attributes of the cipher text are equivalent. Decryption can be possible only when the number of matching meets a threshold value t. An adversary that holds multiple keys should only be able to access data if at least one individual key grants access. Collusion-resistance is security feature of Attribute Based Encryption. The problem with Attribute Based Encryption (ABE) method is that data owner if needs to encrypt, requires every authorized user's public key[9]. Private keys can be created by key generation authority and attributes of the users will be handled by attribute management authority, authorities cannot pool data, thereby avoiding collision attack.

4.2 Homomorphic Encryption

Homomorphic encryption is the technique that allows computations on cipher text messages and produce a ciphered result which when converted to plaintext, is same as the result when operations performed on the plaintext[9]. It has been used for encrypting data as well as for private information retrieval[9,10]. It is the encryption scheme which means the operations on the encrypted data[11]. It can be applied in any system by using various public key algorithms. When the data is transferred to the public area, there are many encryption algorithms for secure the storage of data[11]. But when there is a need to process data located on remote server and to preserve privacy, homomorphic encryption is useful**.** The case study on various principles and properties of homomorphic encryption is given and then various homomorphic algorithms using asymmetric key, and symmetric key systems such as RSA, ElGamal, Paillier, Goldwasser Micali, Benaloh, Okamoto uchiyama algorithms Figure 1 depicts the process of homomorphic encryption.

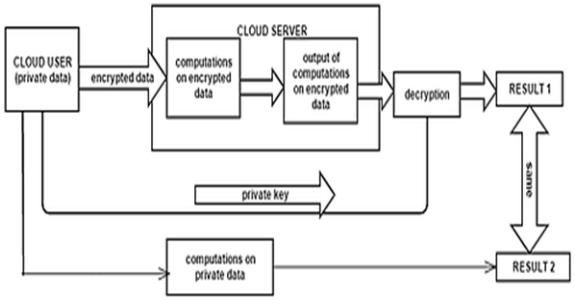

Figure 1. Homomorphic encryption.

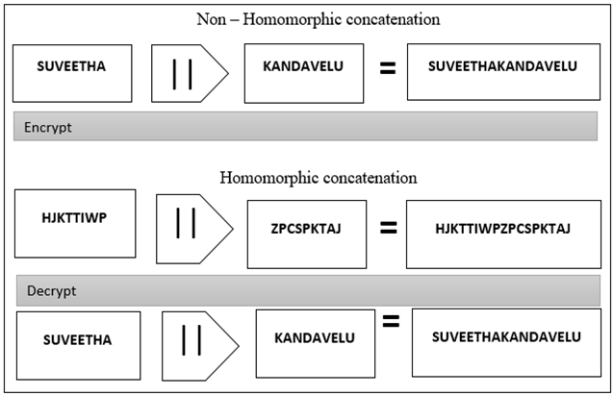

Figure 2. Sample Homomorphic Encryption.

There are many methods that guarantee privacy, such as data encryption and usage of tamper- resistant hardware. Even then, the problem arises when there is a need for computations on encrypted private data. In this situation, where homomorphic encryption can be implemented[11]**.** Security plays a major role in the proliferating usage of the internet or the usage of public cloud for storing sensitive data. Security is essential for maintaining integrity, confidentiality, availability of the user's data. Figure 2 depicts the same homomorphic encryption technique.

4.2.1 Functions of Homomorphic Encryption

Homomorphic Encryption H has a set of four functions.

- H = {**Key Generation,** Encryption, Decryption,

Evaluation} 1. Key generation: Client will create a pair of keys, public key (pk) and secret key (sk) for the conversion of plaintext to cipher text.

- **Encryption**: Using secret key (sk) client encrypts the plain text PT and generate Esk (PT) with the use of public key (pk), this cipher text CT will be transmitted to the server.

Evaluation: Server has a function f for doing evaluation of cipher text CT and calculate the function using public key pk.

Decryption: Generated Eval (f) PT)) will be converted to plaintext y the client using its sk and it gets the original message.

4.2.2 Properties of Homomorphic Encryption

Homomorphic Encryption has mainly two properties,

Additive Homomorphic Encryption: A Homomorphic encryption is additive[12], if : Ek (PT1⊕PT2) = Ek (PT1) ⊕ Ek (PT2)[13]

Multiplicative Homomorphic Encryption: A Homomorphic encryption is multiplicative,

Ek (PT1) *(PT2) = Ek (PT1 *PT2)

4.2.3 Homomorphic Encryption Algorithms:

Table 1 shows the various homomorphic encryption algorithms and their properties.

RSA

In RSA Cryptosystem, public key is modulus m, exponent ∈, then the encryption process of a message x is given by $\varepsilon(x) = x^e \bmod m$. The multiplicative homomorphic property is then,

$$\varepsilon(x_1).\varepsilon(x_2) = x_1^e x_2^e \bmod m = (x_1 x_2)^e \bmod m = \varepsilon(x_1.x_2)$$

ElGamal

In the ElGamal cryptosystem, the public key is (G q g h), where $h=g^x$, and in a cyclic group G of order q with generator g, and x is the secret decryption key, then the encryption process of a message m is $\varepsilon(m) = (g^r, m, h^r)$, for some random $r \in \{0,\ldots,q-1\}$.. The homomorphic property is then

$$\varepsilon(m_1).\varepsilon(m_2) = (g^{r1}, m_1, h^{r1})(g^{r2}, m_2, h^{r2})$$
$$= (g^{r1}+{}^{r2}, (m^1.m^2)h^{r1}+{}^{r2}) = \varepsilon(m_1.m_2).$$

Goldwasser-Micali

In the Goldwasser-Micali cryptosystem, public key is modulus m and quadratic non-residue x, then the encryption of a bit b is $\varepsilon(b) = x^b r^2 \bmod m$, for some random $r \in \{0,\ldots,m-1\}$. The homomorphic property is then

$$\varepsilon(b_1).\varepsilon(b_2) = x^{b1} r_1^2 x^{b2} r_2^2 = x^{b1+b2}(r_1 r_2)^2 = \varepsilon(b_1 \oplus b_2)$$

where ⊕ denotes addition modulo 2, (i.e. exclusive-or).

Benaloh

In the Benaloh cryptosystem, public key is the modulus m and the base g with a blocksize of c, then the encryption process of a message x is $\varepsilon(x) = g^x r^c \bmod m$, for some random $r \in \{0,\ldots,m-1\}$. The homomorphic property is then

$$\varepsilon(x_1).\varepsilon(x_2) = (g^{x1} r_1^c)(g^{x2} r_2^c) = g^{x1+x2}(r_1 r_2)^c = \varepsilon(x_1 + x_2 \bmod c)$$

Paillier

In the Paillier cryptosystem, public key is the modulus m and the base g, then the encryption process of a message x is $\varepsilon(x) = g^x r^m \bmod m^2$, for some random $r \in \{0,\ldots,m-1\}$. The homomorphic property is then

$$\varepsilon(x_1).\varepsilon(x_2) = (g^{x1} r_1^m)(g^{x2} r_2^m) = (g^{x1+x2}(r_1 r_2)^m = \varepsilon(x_1 + x_2 \bmod m^2)$$

Table 1. Homomorphic Encryption algorithms

Scheme	Homomorphic property	Algorithm
RSA	Mulitplicative	Asymmetric
Elagamal	Mulitplicative	Asymmetric
Goldwasser Micali	XOR	Asymmetric
Benaloh	Additive	Symmetric
Paillier	Additive	Asymmetric
Okamoto Uchiyama	Additive	Asymmetric

5. Related Work

Deepak Puthal et al.[1] explained various cloud services such as IaaS, PaaS, SaaS and security issues in IaaS, PaaS, SaaS. They have also discussed cloud architecture and challenges such as handling uncertainties, managing dynamic changes in workload, optimization of virtual network, VMs Consolidation is efficient for controlling heterogeneous assignment, public auditing, data availability.

Monikandan et al.[2] proposed a confidentiality

technique, MONcrypto is build on data obfuscation method to secure the data in cloud data center. By using this techniques, it lessen the size of cipher text data that is uploaded to data center.

Jayalekshmi et al.[3] discussed about the common threats such as data loss, leakage, malicious insiders, account or service hijacking in cloud and explained various privacy issues in public cloud like confidentiality, integrity, and availability.

Jiadi Yu et al.[4] proposed order-preserving encryption (OPE) for server-side ranking that leaks data privacy. To handle this leakage and eliminate it, they proposed a two-round searchable encryption (TRSE) that allows top-k multikeyword retrieval. In TRSE, they proposed a homomorphic encryption and vector space model. The homomorphic encryption facilitate server by operations only on cipher text and the vector space model helps to provide search accuracy. The sensitive data leakage can be diminished and security of data can be enhanced.

Monique Ogburna et al.[5] discussed the concepts and significance of homomorphic encryption and limitations associated with this type of encryption scheme.

Ramalingam Sugumar et al.[7] proposed a symmetric encryption algorithm for data security in cloud storage. SEA converted plaintext into ASCII code which was used to process encryption of cloud users data. Cloud provider and cloud users cannot access the data in cloud. SEA reduced the time taken for encryption and decryption process and also helps cloud provider to maximize cloud data security.

Saikeerthana et al.[8] proposed a Cipher text Policy Attribute Based Encryption, where users private keys are created depends upon the attributes of the users. In proposed scheme, keys are generated by key generation authority and attributes of the users will be handled by attribute management authority, authorities are not able to pool data, thereby avoiding collision attack.

C. Gentry[9] proposed the various homomorphic schemes and described about some what homomorphic encryption and fully homomorphic encryption.

Thomas Plantard described[13] about hidden ideal lattices possible in homomorphic encryption techniques.

C. Gentry[10] proposed a fully homomorphic encryption that keeps user's data secure. It allows to process on ciphered data without using decryption key, even when the function of the data was very complex. Any third party could perform complicated processing on encrypted data.

Jung Hee Cheon et al.[11] proposed a hybrid homomorphic encryption. In this method both Public-Key Encryption (PKE) and Somewhat Homomorphic Encryption (SHE) were combined, to reduce the storage requirements of most SHE or Fully Homomorphic Encryption (FHE) and use the applications, by combing the ElGamal and Goldwasser-Micali algorithms.

Licheng Wang et al.[12] proposed a fast variant scheme by using Paillier's scheme to quicken exponentiations in decryption. Performance evaluation represents protocols could attain around 46.9% in encryption process and a 50% in decryption process with the cost of two times the cipher text length than Paillier's additively homomorphic encryption method.

Lifei Wei et al.[14] described about the protocol SecCloud secure computation auditing protocol, addressing protected storage and computation auditing in cloud and thereby obtaining privacy cheating discouragement by designated verifier signature. They had also implemented protected cloud computing environment, or SecHDFS.

Mazhar Ali et al.[15] described about various cloud challenges like confidentiality, integrity, availability, sharing the resources.

Naveed Islam et al.[16] stated a approach for sharing images using public key cryptosystems homomorphic properties, i.e. RSA and Paillier, the proposed scheme use secret sharing that check the influence of the dealer over the protocol and allows each player to perform with the help of his key-image. At the encryption process, every player encrypts his own key-image using public key of the dealer. The dealer encrypts the secret-to-be-shared image with the same public key and then, the l encrypted key-images including the encrypted to-be shared image are multiplied homomorphically to get another encrypted image.

P. Y. A. Ryan[17] described the thought of voter-verifiability, and it explains voting protocol, Pret a Voter protocol, for accomplishing voter-verifiability. Paillier encryption supports the secret key holder to get back the randomization as well as the plaintext, auditing of the ballot receipts and avoids the need to provide Zero-Knowledge Proofs.

Govinda Ramaiah et al.[18] proposed a Somewhat Homomorphic public key encryption method, with extremely smaller public key, and it was good to encrypt integer plaintexts rather than single bits, with approximately lower message expansion and computational complications.

Zekeriya Erkin et al.[19] proposed generating recommendations by encrypting private data and transform them. They involved a third party and using data packing, constructed a structure without a need for user participation and also compares multiple values that were packed in one encryption.

6. Proposed Work

In banking dataset, user's data are very sensitive, which can be stored in cloud, where it can be hacked by unauthorized users, encrypting these sensitive information by cloud users before sending to the cloud achieves security to data. If cloud service provider wants to perform some calculations on encrypted data, it is possible to perform operations on ciphered data without using the decryption key, we applied paillier algorithm multiplicative property and additive property to banking dataset for calculating total interest on encrypted banking dataset without revealing the secret key, thereby providing confidentiality to cloud users.

7. Experimental Results

Results of the employment of the Paillier Homomorphic technique has been discussed, file size vs. encryption time in addition with file size vs. decryption time presented in terms of homomorphic encryption analyzed in terms of encryption and decryption time.

7.1 Encryption Time vs. File Size

Different file sizes are considered and respective time taken for encryption was calculated. The Figure 7.1 shows the graphical representation of the same. Table 7.1 shows the encryption time of files ranges from 62KB to 368KB and Encryption time.

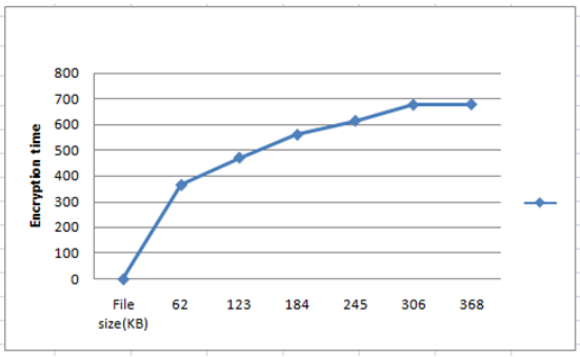

Figure 7.1. Different File Size vs. Encryption time.

Table 7.1. Shows different file size and decryption time

File size (KB)	Encryption Time (ms)
62	366
123	472
184	561
245	615
306	677
368	679

7.2 Decryption Time vs. File Size

Different file sizes are considered and respective decryption time is calculated. Figure 7.2 shows the graphical representation of the same. Table 7.2 shows the decryption time of files ranges from 62KB to 368KB and decryption time.

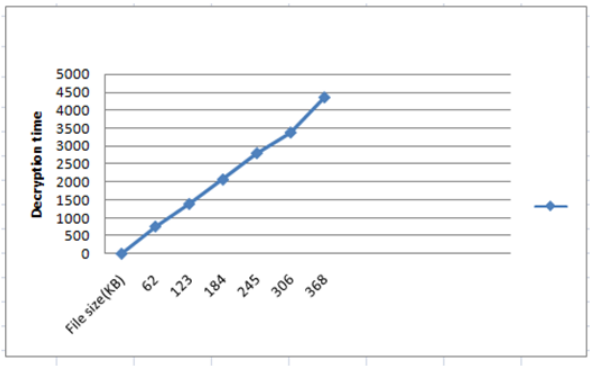

Figure 7.2. Different File Size vs. Decryption time.

Table 7.2. Shows different file size and decryption time

File size (KB)	Decryption Time (ms)
62	759
123	1385
184	2066
245	2798
306	3373
368	4350

8. Conclusion

Cloud provider's responsibility is to secure user's data, they hold on behalf of organizations and users. Although cloud computing has many advantages, there are still many actual problems that need to be solved.

Homomorphic encryption can be carry out for securing data in cloud service provider, when getting storage as a service from cloud. Homomorphic encryption can enable us to calculate operations on encrypted data without decrypting it. Thus data security and confidentiality can be achieved through homomorphic encryption. In future, the various issues of cloud computing such as integrity verification, resource management problem can be addressed.

9. References

1. Puthal D, Sahoo BPS, Mishra S, Swain S. Cloud Computing Features, Issues and Challenges: A Big Picture. International Conference on Computational Intelligence & Networks. 2015 Jan; p. 116-23.
2. Monikandan S, Arockiam L. Confidentiality Technique to Enhance Security of Data in Public Cloud Storage using Data Obfuscation. Indian Journal of Science and Technology. 2015 Sep; 8(24):1-10.
3. Jayalekshmi MB, Krishnaveni SH. A Study of Data Storage Security Issues in Cloud Computing. Indian Journal of Science and Technology. 2015 Sep; 8(24).
4. Yu J, Lu P, Zhu Y, Xue G, Li M. Toward Secure Multike-word Top-k Retrieval over Encrypted Cloud Data. IEEE Transactions on dependable and secure computing. 2013 Aug; 10(4):239-50.
5. Ogburna M, Turnerb C, Dahalc P. Homomorphic Encryption. Elsevier: Procedia computer Science.2013; 20:502-09.
6. Plantard T, Susilo W, Zhang Z. Fully Homomorphic Encryption Using Hidden Ideal Lattice. IEEE transactions on information forensics and security. 2013 Dec; 8(12):2127-37.
7. Sugumar R, Imam SBS. Symmetric Encryption Algorithm to Secure Outsourced Data in Public Cloud Storage. Indian Journal of Science and Technology. 2015 Sep; 8(23):1-5.
8. Saikeerthana R, Umamakeswari. Secure Data Storage and Data Retrieval in Cloud Storage using Cipher Policy Attribute based Encryption. Indian Journal of Science and Technology. 2015 May; 8(S9):318-25.
9. Gentry C. A fully homomorphic encryption scheme, PhD thes is. Stanford University; 2009.
10. Gentry C. Computing arbitrary functions of encrypted data. Magazine Communications of the Association for Computing Machinery. 2010 Mar; 53(3);97-105.
11. Cheon JH, Kim J. A Hybrid Scheme of Public-Key Encryption and Somewhat Homomorphic Encryption. IEEE transactions on information forensics and security. 2015 May; 10(5):1052-63.
12. Wang L, Wang L, Pan Y, Zhang Z, Yang Y. Discrete logarithm based additively homomorphic encryption and secure data aggregation. Information and communications security. 2009; 5927:493-502
13. Wang Q. Wang C, Ren K, Lou W, Li J. Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing. IEEE transactions on parallel and distributed systems. 2011 May; 22(5):847-59.
14. Wei L, Zhu H, Cao Z, Dong X, Jia W, Chen Y, Vasilakos AV. Security and privacy for storage and computation in cloud computing. Elsevier: Information Sciences. 2013; 258:371-86.
15. Ali M, Khan SU, Vasilakos AV. Security in cloud computing: Opportunities and challenges. Elsevier: Information Sciences. 2015; 305:357-83.
16. Islam N, Puech W, Hayat W, Brouzet Robert. Application of homomorphism to secure image sharing. Elsevier: Optics Communications. 2011; 284(19):4412-29.
17. Ryan PYA. Pret a Voter with Paillier encryption. Elsevier: Mathematical and Computer Modelling. 2008; 48(9-10):1646-62
18. Ramaiah YG, Kumari GV. Efficient Public key Homomorphic Encryption Over Integer Plaintexts. IEEE Conference 2012.
19. Erkin Z, Veugen T, Toft T, Lagendijk TK. Generating Private Recommendations Efficiently Using Homomorphic Encryption and Data Packing. IEEE transactions on information forensics and security. 2012 June; 7(3):1053-66.