International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-2, December, 2019

Homomorphic Cryptosystems for Data Security in Cloud Storage

Vidyullata Devmane, B. K. Lande, Dilendra Hiran, Jyoti Joglekar

Abstract: ***Cloud Computing enables users to use remote resources thus reduces the burden on local storage. However, the use of such services gives rise to new set of problems. The users have no control over the data which they have stored on those storages so to achieve data authentication with confidentiality is utmost important. As every user may not have that expertise so they can request for data verification task to Trusted Verifier (TV) which will be an authorized party to check the intactness of outsourced data. Since the data owner stores the data on the cloud in an encrypted format, it becomes difficult to check the integrity of the data without decrypting. But by using homomorphic encryption schemes the integrity checking can be made possible without original copy. In this paper, we have given implementation and performance details of two homomorphic encryption schemes, Rivest Shamir Adleman (RSA) and Paillier. The RSA is multiplicative homomorphic scheme where the Paillier is additive homomorphic scheme. Both the algorithms are partially homomorphic thus limited in their functions. Due to homomorphic property of these algorithms, original contents will not get revealed in the verification process. This framework will achieve authentication of data by maintaining confidentiality.***

Keywords: Homomorphic algorithms, Data Integrity in Cloud Storage, Trusted Verifier.

I. INTRODUCTION

In the need of internet and remote data storage usage has led to the growth of the era of cloud computing. Data center's today come with various computing powers, the increased network bandwidth and reliable remote servers. Due to Cloud Computing, users can take advantages of remote computing resources which reduces maintenance cost. It is easy to place the data into a cloud as users will not be managing any hardware or software complexities required to store data. Amazon Simple Storage Service and Amazon Elastic Compute Cloud are some well known vendors preferred by users [1] [3].

Cloud computing storage providers gives facility to store large amounts of data but many times this data will be at risk and users has to rely on storage providers for the availability of the data, where conventional cryptographic algorithms will not be possible to apply [3][7][8].

Basically verification of the data deployed on cloud storage has to be done without revealing the contents. Even the data stored on cloud may have to be checked frequently but as a large number of users are storing different types of data which makes the verification process more challenging. Service providers gives different facilities to the users so that data can be updated on the cloud storage itself dynamically. Thus to make sure, data correctness under dynamic modifications is also necessary [3].

Further the paper is organized as follows, section II explains the types of homomorphic encryption schemes and gives the theoretical comparison of the RSA and Paillier cryptosystems. Section III provides the analysis of our implementation of the homomorphic property of RSA & Paillier algorithms. Section IV gives the concluding remarks.

II. LITERATURE REVIEW

In group theory, the most important functions between two groups are to preserve the group operations which can give homomorphism. A function f: $G \rightarrow H$ between two groups is a homomorphism when $f(xy) = f(x) \cdot f(y)$ for all x and y in G. That means the given function f can be used to convert the operations performed in domain $G \rightarrow H$ so that they can yield same output for any operation.

Thus homomorphic encryption is a process in which some operations will be possible on cipher texts without using the private key, which will produce results equivalent to operations done on original texts [2].

Homomorphic encryption functions can be partially or fully homomorphic encryption algorithms. Cryptographic algorithms will have some basic properties as mentioned here [11]. Deterministic encryption algorithms: If an encryption algorithm is applied on same original texts it generates always same cipher text with same key. Probabilistic encryption algorithms: If an encryption algorithm is applied multiple times on same original text it will generate different cipher texts. Semantically secure cryptosystem: If cipher text and its length of specific message is given but it will not produce any clues about the original texts then the probabilistic algorithm is called as semantically secure.

Revised Manuscript Received on December 30, 2019.
* Correspondence Author
Vidyullata Devmane*, Ph.D Scholar, Department of Computer Engineering, PAHERU Udaipur, Rajasthan, India.
E-mail: devmane.vidyullata@gmail.com
Dr. B. K. Lande, Professor, Department of Electronics, Datta Meghe College of Engineering, Airoli, Navi Mumbai, India.
E-mail: bklande@gmail.com
Dr. Dilendra Hiran Professor, Pacific Academy Higher Education Research University, Udaipur, Rajasthan, India.
E-mail: sigmapawan72@gmail.com
Dr. Jyoti Joglekar, Professor, Department of Computer Engineering, K. J. Somaiya College of Engineering, Mumbai, India
E-mail: jyotij1968@gmail.com

Malleable Cryptosystem: Transforming one cipher text into another cipher text so that it will produce decrypted results based on similar original contents.

Table-I: Homomorphism of various asymmetric algorithms with its level of security [11].

Algorithm	Homomorphism	Security
RSA	Multiplicative	• Padded RSA supports Probabilistic encryption, • Padded RSA Semantically secure, • Unpadded comparatively not giving better security,
ElGamal	Multiplicative	• Semantically secure, • Unconditionally malleable so vulnerable under chosen cipher text attack, • Padded scheme will give security against chosen cipher text.
Paillier	Additive	• Probabilistic cryptosystem, • Semantically secure against chosen plaintext attack, • It is malleable so does not protect against adaptive chosen cipher text attack.
Gold Wasser Micali(GM)	Multiplicative	• Probabilistic cryptosystem, • Cipher text can be extremely larger size so not efficient, • Semantically secure.
Boneh Goh Nissim	Additive & Multiplicative	• Semantically secure on the basis of the subgroup decision problem.

RSA is a partially homomorphic cryptosystem. It is multiplicatively homomorphic algorithm.

Homomorphic property of RSA Cryptosystem:

Given Ct_i = Encryption (Pt_i) = $(Pt_i)^e$ mod n then following properties hold

$Ct_1 = (C)^e \bmod n$ (1)

$Ct_2 = (Pt_2)^e \bmod n$ (2)

$Ct_1 \cdot Ct_2 = (Pt_1)^e \cdot (Pt_2)^e \bmod n = (Pt_1 \cdot Pt_2)^e \bmod n$ (3)

Where, Pt_1, Pt_2 are message 1, message 2,

Ct_1,Ct_2 are encryption of Pt_1 & Pt_2.

Consider the following example for better understanding of homomorphism, assume the following RSA keys:

Public key= (7,33) ; private key = (3);

and messages $Pt_1 = 2$ & $Pt_2 = 5$

Now encrypting the messages using the public key yields

$Ct_1 = 2^7 \bmod 33 = 29$.

$Ct_2 = 5^7 \bmod 33 = 14$

$Ct_1 \cdot Ct_2 = 29 \cdot 14 \bmod 33 = 406 \bmod 33 = 10$; this is the same as

$Pt_1 \cdot Pt_2 = 2 \cdot 5 \bmod 33 = 10$

$Enc(Pt_1 \cdot Pt_2) = 10^7 \bmod 33 = 10$

Paillier Homomorphic Property :

Paillier cryptosystem is also partially homomorphic and in additive nature.

Homomorphic property of Paillier Cryptosystem:

$Ct_1 = gPt^1 \cdot x_1^n \bmod n^2$ (4)

$Ct_2 = gPt^2 \cdot x_2^n \bmod n^2$ (5)

$g \in Z_n^{2*}$

x is a random number

Additive Homomorphism:

$Ct_1 \cdot Ct_2 = gPt^1 x_1^n \cdot gPt^2 \cdot x_2^n \bmod n^2 = Pt_1 + Pt_2 \bmod n^2$ (6)

Table-II: Theoretical Comparison of RSA and Paillier

Characteristic	RSA	Paillier
Security depends on	Difficult for factoring large prime numbers	Intractability of distinguishing n^{th} residues modulo n^2 from non – n^{th} residue
Type	Randomized and Asymmetric	Probabilistic and Asymmetric
Key Generation	Slow	Moderate
Encryption and Decryption	Fast for smaller key size	Moderate
To make secure	Choose large p & q	Choose large p & q
Application	Secure Internet Banking	E-voting, E-cash

III. IMPLEMENTATION & ANALYSIS

In our implementation, the original files are divided into blocks of bytes and then the blocks are converted into numbers. The number files of the files whose content intactness is to be verified are used to calculate a tag. Any size of the file will work. Only authorized users can perform encryption and decryption, but hash of encrypted files can be checked remotely by TV.

Table III gives performance time required for the sample files of 1.39kb,2.02 kb & 16 kb sizes to encrypt, decrypt. It also shows the size of encrypted file. Multiplication and encryption of 2 files in number format of original files of 2.02 kb produces a file of size 78.4 KB and takes time of 0.96sec.Similarly for file sizes of 1.39kb and 16kb shown the table III.

In RSA system, hash value of two files in number formats of size 16kb each by Message Authentication Code MD5 (MACMD5) algorithm is taking 17.064 seconds for multiplication and 0.047 seconds for hash generation at owner side. For those same files in encrypted format will be taken by verifier to calculate hash. In this example multiplication time is 0.873sec and 0.359sec for hash calculation time at TV side required.

Table-III: RSA results with sample data files

Original File Size(kb)	File in Number Size (kb)	Encrypted File Size (kb)	Encryption Time (sec)	Decryption Time (sec)
1.39	3.48	54.2	0.73	2.359
2.02	5	78.4	0.96	3.562
16	39.7	617	16.32	51.88

Table-IV: Paillier Results with sample data files

Original File Size (kb)	File in Number Size (kb)	Encrypted File Size(kb)	Encryption Time (sec)	Decryption Time (sec)
1.39	3.48	27.1	0.447	0.22
2.02	5.00	39.2	0.61	0.252
16	39.7	309	2.399	19.6

Table IV gives performance time required for the sample files of 1.39kb,2.02 kb & 16 kb sizes to encrypt, decrypt. It also shows size of encrypted file. Addition and encryption of 2 files in number format of original files of 2.02 kb produces a file of size 39.2 KB and takes time of 0.61sec.Similarly for file sizes of 1.39kb and 16kb are also shown in the same table.

In the table IV for Paillier algorithm hash value of two files in number formats of size 16kb each by Message Authentication Code(MACMD5) algorithm is taking 27.706 sec for addition and 0.031 sec for hash generation at owner side. Those files in encrypted format will be taken by verifier to calculate hash. In

this example addition time is 0.358 sec and 0.295sec for hash calculation time.

IV. CONCLUSION

Analysis of RSA and Paillier with homomorphic property is done. We have shown example of homomorphism of RSA for better understanding. Also analysis of encryption, decryption and hash calculation of different files with various file sizes at owner and verifier side by both the algorithms are explained. Using the homomorphic properties of RSA or Paillier cryptosystems, we can check the intactness of remotely stored data without having the original copy, thus can achieve public auditability. Only complexity of performance should not be criteria for the cryptographic protocols but its security strength is very important factor. RSA algorithm with very large prime numbers can give better security but as per basic nature of RSA algorithm it is vulnerable to factorization attack. Decisional composite residuosity problem is intractable and the Paillier encryption scheme was proved under DCR assumption, so can apply Paillier algorithm to secure the data.

REFERENCES

1. P. Mell and T. Grance, “Draft NIST working definition of cloud computing”, referenced on June. 3rd, 2009. http://csrc.nist.gov/groups/SNS/cloudcomputing/index.html
2. S. Ravindran and P. Kalpana, “Data Storage Security Using Partially Homomorphic Encryption in a Cloud” , Volume3, Issue 4, April 2013
3. Cong Wang, S. M. Chow, Qian Wang, “Privacy-Preserving Public Auditing for Secure Cloud Storage”, VOL. 62, NO. 2, February 2013.
4. Sigrun Goluch, “The development of Homomorphic Cryptography”, Vienna University of Technology.
5. Shai Halevi, “On Homomorphic Encryption and Secure Computation”, IBM/NYU/Columbia Theory, May 7, 2010.
6. Boyang Wang, Baochun Li and Hui Li, “Oruta: Privacy Preserving Public Auditing for Shared Data in Cloud”, IEEE 2014.
7. Kan Yang, Xiaohua Jia, “Security for Cloud Storage Systems”, pp. 1-2, Springer, Hong Kong, March 2013.
8. Q.Wang,C. Wang, K. Ren, W. Lou, and J. Li, “Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing”, IEEE Transaction.
9. “Parallel and Distributed Systems”, vol. 22, no. 5, pp. 847-859, 2011.
10. C. Wang, Q. Wang, K. Ren, and W. Lou, “Towards secure and dependable storage services in cloud computing”, IEEE Transactions on Service Computing, 2011.
11. Yi X., Paulet R., Bertino E., “Homomorphic Encryption and Applications”, chapter 2,Springer Publication, Year 2014.

AUTHORS PROFILE

Vidyullata Devmane, received B.E. Computer Engineering degree from Walchand College of Engineering, Maharashtra, India. She possesses Masters degree in Computer Engineering from the University of Mumbai, India. And she is pursuing Ph.D in Computer Engineering from PAHERU, Rajasthan, India. She is currently working as an Associate Professor with the Department of Computer Engineering, Shah & Anchor Kutchhi Engineering College, Mumbai. Her areas of interests are Cryptography & System Security, Advanced Algorithms and System Programming.

B. K. Lande, has received the Ph.D degree from the Indian Institute of Technology (IIT) Bombay, Masters degree from Walchand College of Engineering, Maharashtra, and B.E. degree from Nagpur University, Maharashtra, India. Currently he is working as a Professor in Datta Meghe College of Engineering, Airoli, Navi Mumbai. His areas of research are Controls & Communication, Image Processing, Cryptography & System Security. Reviewer of various reputed publications.

Dr. Dilendra Hiran, is currently working as Professor at PAHER University, Udaipur, Rajasthan, India. Secured Doctoral degree from PAHER University, Udaipur in 2015. Currently holding the post of Principal at Pacific Institute of Computer Application.

Jyoti Joglekar, received the Ph.D degree from the Indian Institute of Technology (IIT) Bombay, Mumbai, India, in 2015. She possesses Master of Engineering degree in computer engineering from the University of Mumbai, Mumbai, India. She is currently a Professor with the Department of Computer Engineering, K. J. Somaiya College of Engineering, Mumbai, India. Her areas of research are satellite image processing and analysis, machine learning and big data analytics, Cryptography & System Security. She is a Life Member of the Indian Society for Remote Sensing and a Fellow member of IETE. Reviewer of various reputed publications.