ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

Data Security Using Advanced Homomorphic Encryption Scheme in Cloud Storage

Dr.P.Sumitra[1], P.Kaladevi[2]

Assistant Professor, Department of Computer Science & Applications, Vivekanandha College of Arts and Sciences for Women, Elayampalayam, Tiruchengode, India[1]

M.Phil Scholar, Department of Computer Science & Applications, Vivekanandha College of Arts and Sciences for Women, Elayampalayam, Tiruchengode, India[2]

ABSTRACT: Cloud computing techniques are used to share resources. Storing an important data with cloud storage providers comes with serious security risks. The cloud can modify the stored data, leak the stored confidential data, or return some inconsistent data to different users. This may happen due to operator errors, crashes, misconfiguration or bugs. Also, malicious security breaches such as penetration of external adversaries into the cloud storage provider, or an attack from any employee, can be much harder to detect or more damaging than accidental ones. So, the cloud data security requires authentication and integrity analysis for the storage data values. Public data audit ability and data dynamics model ensures the integrity of data storage in Cloud Computing. In existing system, an erasure code provides redundancy by breaking objects up into smaller fragments and storing the fragments in different places. The key is used to recover the data from any combination of a smaller number of those fragments. We enhance this with an advanced homomorphic encryption (AHE) scheme which is integrated with a decentralized erasure code to formulate a secure distributed storage system. The distributed storage system not only supports secure and robust data storage and retrieval. It meets the requirements that storage servers independently perform encoding and re-encryption and key servers independently perform partial decryption. The user forwards his data in the storage servers to another user without retrieving the data back. A major feature of the cloud services is that users' data are usually processed remotely in unknown machines that users do not own or operate. We propose an advanced homomorphic encryption (AHE) scheme, whereby the encryption and decryption capability of the system is enhanced.

KEYWORDS: Homomorphic Encryption, cryptographic Keys, Network Attached Storage, Network File System

I. INTRODUCTION

The general process in cloud storage transfers the application software and databases to the centralized large data centres. The management of the data and services may not be fully trustworthy. Storing data in a third party's cloud system causes serious concern on data confidentiality [1]. A formal method, in order to provide strong confidentiality for messages in storage servers, the messages are encrypted by a cryptographic method before encoding it by an erasure code method, and then it can be stored. When the user wants to use a message, the code word symbols must be retrieved from the storage servers, then it must be decoded, and then decrypting them by using cryptographic keys.

There are several problems in the above mentioned straightforward integration system of encryption and encoding. Three of them are prioritized as follows: First, the user has the responsibility of most computation and the communication traffic between the user systems and storage servers and it is highly complicated process. Second, the user has to manage the cryptographic keys [2] .The security of entire system will be broken, if the user's device that store's the keys has been lost or compromised. Finally, besides data storage and retrieval, it is complicated for the storage servers to directly support some other functions like, directly forwarding the messages from one user to another one. A straight forward solution to supporting the data forwarding function in a distributed storage system is as follows: when the user A wants to forward or send a message to user B, then, he should download the encrypted message and decrypt it by using the secret key.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

Later he needs to encrypt the message by using user B's public key and upload the generated cipher-text. When user B wants to retrieve the forwarded message from user A, he can download the cipher-text and decrypt it by using his secret key. The entire data forwarding process needs the low cost of three communication rounds as, user A's downloading and user A's uploading and user B's downloading. The communication cost consumed for this transaction is linear in the length. The computation cost is the decryption and encryption for the user A, and the decryption for user B. This problem is resolved by using the key derivation hierarchy and advanced homomorphic encryption scheme, that can significantly decrease communication and computation cost of the data owner. In the advanced homomorphic encryption scheme, the elliptic curve encryption supports data encryption and forwarding at the same cost [5]. Thus, the communication cost of the data owner is independent of the length of forwarded message. The advanced homomorphic encryption scheme significantly reduces the overhead of the data encryption and forwarding function for a secure storage system [9].

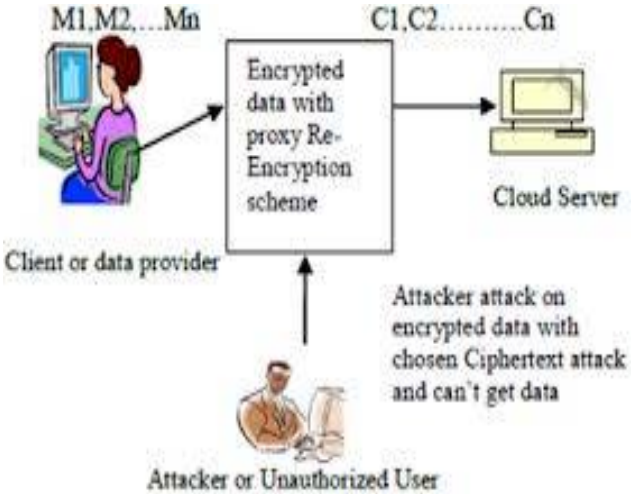

Fig. 1 Homomorphic Encryption Scheme

II. RELATED WORK

Peer-to-peer systems can be characterized as distributed systems in which all nodes have identical capabilities and responsibilities and all communication is symmetric. File sharing in a peer-to-peer networks are confidential and secure. Distributed networked storage can be useful for peer-to-peer networks or redundant arrays of independent disks (RAID) systems [1]. The problem of distributed networked storage is the centralized server. Several efficient proxy re-encryption schemes, such as Improved Proxy Re-encryption Schemes to Distributed Storage [2] offer security improvements over earlier approaches, but they are unidirectional and more time consuming approaches. The Network-Attached Storage (NAS) and the Network File System (NFS) provide extra storage devices over the network such that a user can access the storage devices via network connection. A decentralized architecture for storage systems offers good scalability because a storage server can join or leave without control of a central authority [4]. A message is encoded as a code word and each storage server stores a code word symbol.

The failure is modelled as an erasure error of the stored code word symbol. Each storage server linearly combines the blocks with randomly chosen coefficients and stores the code word symbol and coefficients, to store a message of k blocks. A user queries k storage servers for the stored code word symbols and coefficients and solves the linear system, to retrieve the message. In addition to storage servers their system consists of key servers that hold cryptographic key shares and work in a distributed way. In order to provide strong confidentiality for messages in storage servers, a user can encrypt messages by a cryptographic method before applying an erasure code method [3] to encode and store messages. An erasure code, the message can be recovered from the code word symbols stored in the available storage servers by the decoding process. This provides a trade-off between the storage size and the tolerance threshold of failure servers.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

III. EXISTING WORK

A. System Model

Storage servers provide storage services and key servers provide key management services. The distributed storage system consists of four phases:

a. System Setup

The system manager chooses system parameters and publishes in the system phase. Each user A is assigned a public-secret key pair. User A distributes the secret key SKA to key servers KSi such that each key server KSi holds a key share of SKA.

b. Data Storage

User A encrypts the message M and distributes it to storage servers SSi. A message M is decomposed into k blocks m1;m2; . . .; mk and has an identifier ID. Each storage server is linearly depend upon receiving cipher texts from user A, and combines them with randomly chosen coefficients into a code word symbol and stores it in the storage server.

c. Data Forwarding

User A forwards the encrypted message with an identifier ID stored in storage servers to user B such that user B can decrypt the forwarded message by his secret key. User A uses his secret key SKA and user B's public key PKB to compute a re-encryption key RK^{ID}A->B and then sends RK^{ID}A->B to all storage servers.

d. Data Retrieval

The user A requests to retrieve a message from storage server. Each key server KSi requests to the randomly chosen storage servers to get the code word symbols and does partial decryption on the received code word symbols by using the key share SKA;I, depends on receiving the retrieval request and executing a proper authentication process with user.

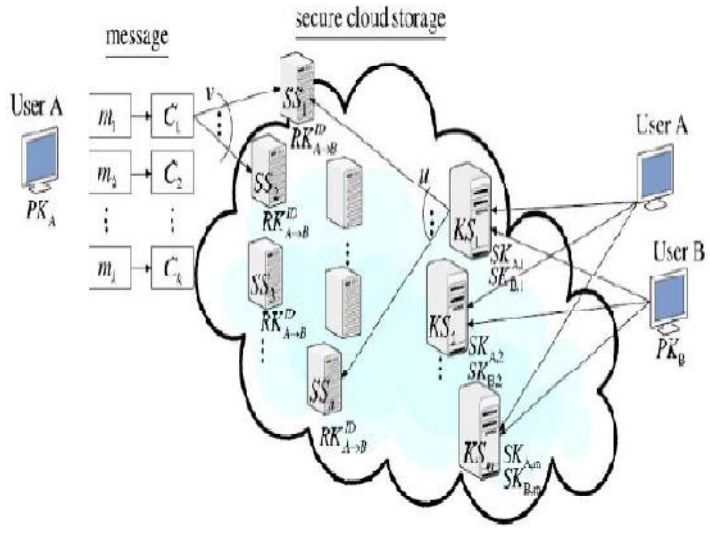

Fig. 2 Base Architecture

PK – Primary Key; SK – Secret Key; M – Message; SS – Storage Server; KS – Key Server; RK – Re-encryption Key.
Fig.2 shows our system model consists of user A and user B, n number of storage servers.SS1; SS2; . . . ; SSn, and m number of key servers KS1; KS2; . . . ; KSm.

III. PROPOSED ATE-SCHEME

In the proposed system we develop the model for Advanced Homomorphic Encryption. Fig. 3 shows the work flow of advanced homomorphic encryption scheme for privacy preserving [6].

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

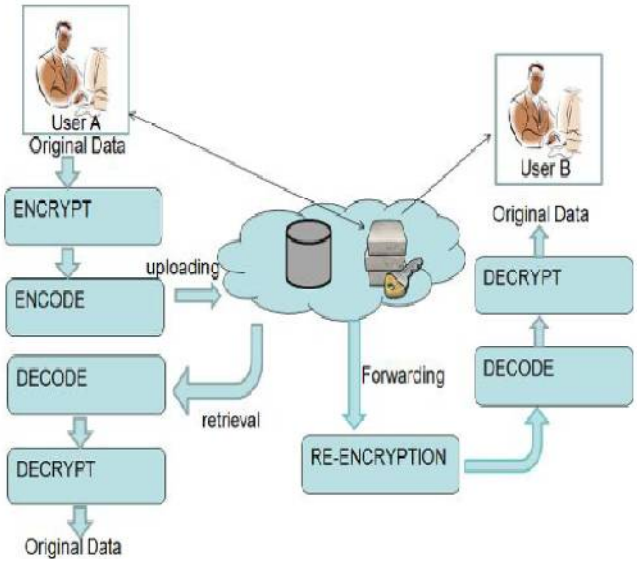

Fig. 3 Work Flow of proposed Scheme

The entire Advanced Homomorphic Encryption model has 7 algorithms are classified as follows

Setup

Developing a cloud environment for run Gen(1ʎ) to obtain (g, h, ê, G1, G2, p), whereê:G1xG2 G2 is a bilinear map, g and h are generators of G1, and both G1 and G2 have the prime order p [8].

A. The polynomial function

The security in this system relies difficult of computing discrete logarithm [8]. The protocols are based on a polynomial function and a set of exponentials. Let p, q be the two large prime numbers such that q|p-1, and g be a generator of order q. The polynomial function of order n is constructed according to the following factor given in Fig 4.

$$f(x) = \prod_{i=1}^{n}(x - x_i) \equiv \sum_{i=0}^{n} a_i x^i \bmod q,$$

where $\{a_i\}$ are coefficients:

$$a_0 = \prod_{j=1}^{n}(-x_j),$$

$$a_1 = \sum_{i=1}^{n}\prod_{i\neq j}^{n}(-x_j),$$

$$\vdots$$

$$a_{n-2} = \sum_{i\neq j}^{n}(-x_i)(-x_j),$$

$$a_{n-1} = \sum_{i=1}^{n}(-x_j),$$

$$a_n = 1.$$

Fig. 4 Polynomial Function of order n.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

B. Key Generation

The decryption keys are denoted by (dij1,dij2), which correspond to (x1, x2) in the 2-degreepolynomial defined above, where dij2 = H(dij1). For simplicity, we denote (dij1,dij2)= (d1,d2).The encryption key corresponding to (d1,d2)is e = (g0,g1,g2), where

g0=$g^a0 = gd1^d2^d$, g1 = $g^a1 = g^{-(d1_1 + d2_2)}$, g2 = g^a2 = g. For simplicity, we have committed the subscripts of eij.

C. Encryption

The encryption algorithm takes as input a message M□ {0,1}*, the encryption key e, a random k□Zq, and a generator h□ Z*p, and outputs a cipher-text (c1,c2), where

C1 □ (hk .gk0, gk1, gk2), C2 = M.hk

D. Encode

Encode (C1; C2; . . . ; Ck). For each cipher-text Ci, the algorithm randomly selects a coefficient gi. If some cipher-text Ci is (0; 1; τ; 1), the coefficient gi is set to 0.

$$
\begin{aligned}
C' &= \left(0, \prod_{i=1}^{k}(\alpha_i^{g_i}), \beta, \prod_{i=1}^{k}(\gamma_i^{g_i})\right) \\
&= \left(0, g^{\sum_{i=1}^{k} g_i r_i}, \tau, \prod_{i=1}^{k} m_i^{g_i} \tilde{e}(g^{a_1}, \tau)^{\sum_{i=1}^{k} g_i r_i}\right) \\
&= (0, g^{r'}, \tau, W\tilde{e}(g,\tau)^{a_1 r'}), \\
\text{where } W &= \textstyle\prod_{i=1}^{k} m_i^{g_i} \text{ and } r' = \sum_{i=1}^{k} g_i r_i.
\end{aligned}
$$

F. Decryption

This algorithm takes as input the cipher-text (c1, c2) and one of decryption keys d1 and d2,and outputs M. hk can be computed from b1 . b di. Thus, M can be commuted as M=C2/hk.

G. Key Derivation

This algorithm takes as input the master decryption key dij1 and a one-way hash function H. It outputs the two child nodes of key dij1. During the key derivation procedure, the left child node can be computed as (d (i+1)(2j-1)1, d(i+1)(2j-1)2) = (H(dij1||(2j-1)), H(H(dij1||(2j-1)))

And the right child node can be computed as(d (i+1)(2j)1, d(i+1)(2j)2) = (H(dij1||2j), H(H(dij1||2j)))By repeating this algorithm, the whole key derivation tree can be generated.

IV. CONCLUSION

Data encryption and key management are important for secure cloud computing. As a traditional approach, tree-based key management has attracted a lot of attention. We found that a traditional tree-based approach has some draw backs in data out sourcing in that a node key holder to derive all child keys, which is also an important feature for key management. Our future enhancement will includes a model to overcome this draw back and also to make use of Advanced Homomorphic Encryption scheme with elliptic curve algorithms for more efficient process.

REFERENCES

[1] Alexandros G. Dimakis, Vinod Prabhakaran, and Kannan Ramchandran, "Decentralized Erasure Codes for Distributed Networked Storage", IEEE Mar 2011.

[2] Giuseppe Ateniese, Kevin Fu, Matthew Green, Susan Hohenberger "Improved Proxy Re-encryption Schemes with Applications to Secure Distributed Storage", IEEE May 2012.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

[3] Hsiao-Ying Lin, Member, IEEE, and Wen-Guey Tzeng, Member, "A Secure Erasure Code-Based Cloud Storage System with Secure Data Forwarding" IEEE. June 2012.
[4] Miao Zhou, Yi Mu, Willy Susilo "Privacy Enhanced Data Outsourcing in the Cloud", 2012.
[5] Hsiao-Ying Lin, Student Member, IEEE, and Wen-Guey Tzeng, Member, IEEE. "A Secure Decentralized Erasure Code for Distributed Networked Storage", Nov. 2010.
[6] Craig Gentry, A Fully Homomorphic Encryption Scheme, 2009.
[7]TEBAA, Maha, Saïd EL HAJJI, and Abdellatif EL GHAZI. "Homomorphic Encryption Applied to the Cloud Computing Security." Proceedings of the World Congress on Engineering. Vol. 1. 2012.
[8] Gennaro, Rosario, and Daniel Wichs" Fully homomorphic message authenticators " May 2012. Cryptology eprint 290, 2012.
[9] Stehle, D. & Steinfeld, R. (2010) "Faster Fully Homomorphic Encryption. In: Advances in Cryptology" – Proceedings of ASIACRYPT'10, Lecture Notes in Computer Science (LNCS), Vol 6477, Springer-Verlag, pp. 377-394