HOMOMORPHIC ENCRYPTION BASED DATA SECURITY ON FEDERATED CLOUD COMPUTING

Anitha R.[1] and Vijayakumar V.[2]
[1]Computer Science and Engineering, Sri Venkateswara College of Engineering, Pennalur, Sriperumbudur, India
[2]School of Computing Sciences and Engineering, VIT University, Kelambakkam, India
E-Mail: ranitha@svce.ac.in

ABSTRACT

Cloud computing usage has increased rapidly in both industries and in research. In recent days as the data grows rapidly, in order to meet the business needs federated cloud is adopted. In federated cloud, as the data is stored and processed away from the user and the cloud service provider, privacy and integrity of the data plays a crucial role. This paper proposes a practical and efficient method for providing security to the data stored at the federated cloud environment using homomorphic techniques. This method provides security by storing the encrypted data in the cloud. The cipher key which is generated for encrypting the data plays a major role. This paper explores important aspects within this context and examines the role of metadata in data security which improves the performance in a secured manner. The proposed novel homomorphic based key distribution protocol is the key area under focus. This proposed work aims to promote the use of homomorphism in multi-clouds due to its ability to reduce security risks using the enhanced modified feistel technique.

Keywords: data security, federated cloud, homomorphic encryption, key distribution.

INTRODUCTION

Homomorphic Encryption is relatively a recent approach that reconsiders the concept of public-key cryptography. This technology expanded exceptionally, which eventually stimulated concerns over ensuring data security in federated cloud networks. According to a recent survey conducted by Cisco Global Cloud Networking Academy, it has been revealed that the security of data is a major hindrance to implement the services in cloud. As data is moved between disparate networks the need for innovative security models for user access to cloud resources is highly required. As a result, it allows companies or organizations to offload the data in a secured manner. In the past few years, the security requirements for data are very strong and many algorithms have evolved based on homomorphic techniques [10]. Only a few algorithms play a comprehensive role in maintaining security to the data at its rest and also in motion. The proposed model also strives to improve the security during data retrieval in cloud scenario without the need to use a centralized control over the encryption and decryption techniques that may be used. The proposed model also deals with collaborative security which involves key generation mechanism and a key distribution technique between the federated clouds. Both these party shares their work on top of the encrypted data. The proposed model aims at performing arbitrary computations on the encrypted data called, homomorphic techniques. As such techniques give rise to privacy; the model tends to perform critical operation on encrypted data. Homomorphic encryption is evolved to solve such critical issues. The homomorphic properties of ciphers have been implemented in various real time applications. Using homomorphic encryption data protection is achieved through which allows additive and multiplicative operations over encrypted bits. The cloud service provider accepts encrypted user query data to perform processing without being aware of its content. The results of the user query which is again an encrypted data is sent to the user. The user alone decrypts the data and views the result of the query. The public-key and private-key cryptosystems are designed with various fault attacks. In the past years, homomorphic Encryption allows simple computation on encrypted Constructing an encryption scheme that is both additively and multiplicatively homomorphic remained a major challenge. The additive and multiplicative homomorphism forms a complete set of operations. The first cipher key generation takes place at the user level and is based on the related metadata attributes using enhanced modified feistel cipher key algorithm. The decentralized key distribution mechanism is proposed. The system ensures that the encryption and decryption keys cannot be compromised without the involvement of the all the clouds in the federated network thereby rendering a collaborative security environment. Different from previous works in secure data outsourcing, in order to focus on the multiple CSP the key management complexity in the federated network improves the security. Extensive analytical and experimental results are presented which show the security, scalability, and efficiency of our proposed scheme. Our contributions can be summarized as follows:

1. We propose a model to create a cipher key C based on the attribute of metadata stored using a modified feistel network to access the data in a secured mode in a federated cloud environment.

2. We have also proposed a novel security policy which involves the multiple clouds in a federated network by means of key creation and distribution policies.

The rest of the paper is organized as follows: Section 2 summarizes the related work and the problem statement. Section 3 describes the system architecture model and discusses the detailed design of the system model. Section 4 describes the key generation mechanism and its distribution in a federated environment. Section 5 describes the enhanced modified feistel network structure design and issues of the proposed model. The performance evaluation based on the prototype implementation is given in Section 6 and Section 7 concludes the paper.

RELATED WORKS

The related work discusses about the previous work carried out in the area of cloud security and we have also discussed about how the technique of homomorphic encryption is used in cloud federated cloud computing environment. Cachin *et al*. [1] argue that when multiple clients use cloud storage or when multiple devices are synchronized by one user, it is difficult to address the data corruption issues. Hendricks *et al*. [2] state that the Byzantine fault-tolerant replication protocol is the solution to avoid data corruption caused by some components in the cloud. Chirag Modi *et al*. [3] discussed a survey paper where they discussed about the factors affecting cloud computing storage adoption, vulnerabilities and attacks, and identify relevant solution directives to strengthen security and privacy in the cloud environment. They discuss about the various threats like abusive use of cloud computing, insecure interfaces, data loss and leakage, identity theft and metadata spoofing attack. J. Ravi Kumar *et al*. [4] shows that third party auditor is used periodically to verify the data integrity stored at cloud service provider without retrieving original data. In this model, the user sends a request to the cloud service provider and receives the original data. If data is in encrypted form then it can be decrypted using his secret key. However, the data stored in cloud is vulnerable to malicious attacks and it would bring irretrievable losses to the users, since their data is stored at an untrusted storage servers. Shizuka Kaneko *et al*. [5] have proposed a query based hiding schema information using a bloom filter. The query given is processed and the attributes of the query is used for key generation. The key generated is used to hide confidential information from the data administrator. As the query gets changes every time the key generation process becomes more complex. Marcos K. Aguilera *et al*. [6] has proposed a practical and efficient method for adding security to network-attached disks (NADs). The design specifies a protocol for providing access to the remote block-based devices using homomorphic schemes. R. Anitha *et al*. [7] has described about the effective usage of feistel network in cloud computing under various aspects. The model described in paper [7] has been adopted in homomorphic encryption technique. Sujitha *et al*. [8] has discussed about the partially and fully homomorphic system. C. Orencik and E. Sava [9] described the Private Information Retrieval (PIR) protocol using homomorphic techniques and provide security during data retrieval.

SYSTEM MODEL

The system framework for the proposed model is as shown in Figure-1. The framework explains about how the data is encrypted and how the keys are shared between the clouds in the federated network. The framework explains about the components involved and their functionalities. The user uploads the data in an encrypted form. The key generation for encryption technique is done by enhanced modified feistel algorithm. The data is stored in the data server in an encrypted form. The data blocks are organized in the data server using Bloom filter based data arrangement algorithm. In this model the user uploads the encrypted file where the cipher key-C_k for encryption process is generated using modified matrix cipher key generation algorithm where the plain text from the user is taken as input in the form of matrices. This model proposes a modified cipher key function F which introduces the novel obfuscations in the matrix along with the key matrix. The cryptanalysis carried out in this paper clearly indicates that this cipher cannot be broken by the brute force attack. This model provides high strength to the cipher, as the encryption key induces a significant amount of matrix obfuscation into the cipher.

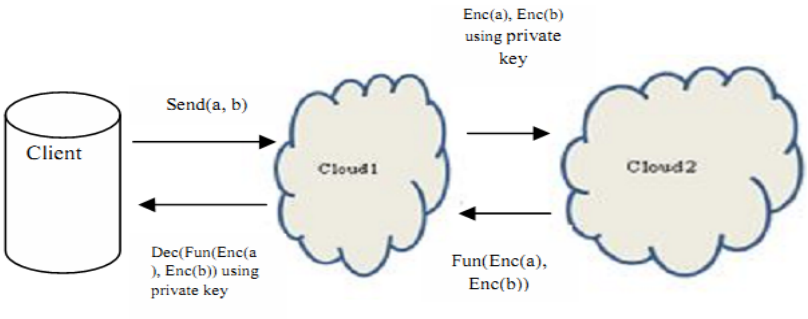
Enc(a), Enc(b) using private key
Send(a, b)
Client
Cloud1
Cloud2
Dec(Fun(Enc(a), Enc(b)) using private key
Fun(Enc(a), Enc(b))

Figure-1. System model.

While downloading the file the key the cipher key C_k is used to decrypt the file. Let m and c denote the plaintext and cipher text of the integer respectively. Our encryption scheme can be expressed as the following formulation: $c = pq + 2r + m$, where p denotes the secret key, q denotes the multiple parameter and r denotes the noise to achieve proximity against brute-force attacks. The public key is $pq + r$. On the basis of homomorphism property, the encryption scheme can be described as four stages: KeyGen, Encrypt, Evaluate and Decrypt.

- KeyGen (λ): The secret key SK is an odd η-bit number randomly selected from the interval [2η−1, 2η).
- Encrypt (P, C_k):
- Decrypt (p, χ): Output m = (χ mod p) mod x

Here the data set is encrypted using enhanced modified feistel symmetric encryption scheme. Note that the evaluate stage sets no limit to how many addition or multiplication operations can be executed without encryption. In fact, the cipher text of an integer, which is another integer, can be applied as many evaluations as needed.

KEY GENERATION MECHANISM

Feistel ciphers are a special class of iterated block ciphers where the cipher text is calculated from the attributes of metadata by repeated application of the same transformation or round function. Development of the cipher key "Cmxn" using Modified Feistel Function is described below. This paper proposes a complex procedure for generating the cipher key "Cmxn" based on matrix manipulations, which could be introduced in symmetric ciphers. The proposed cipher key generation model offers two advantages. First, the procedure is simple to implement and has complexity in determining the key through crypt analysis. Secondly, the procedure produces a strong avalanche effect making many values in the output block of a cipher to undergo changes with one value change in the secret key. As a case study, matrix based cipher key generation procedure has been introduced in this cloud security model and key avalanche have been observed. Thus the cloud security model is improved by providing a novel mechanism using enhanced modified Feistel network where the cipher key Cmxn is generated with the matrix based cipher key generation procedure. The Cipher key generation procedure is based on a matrix initialized using secret key and the modified feistel function F. The input values used in various feistel rounds are taken from the previous round. The selection of rows and columns for the creation of matrix is based on the number of attributes of the metadata and the secret key matrix "K_{mxn}" and the other functional logic as explained in the following subsections. The procedure for encryption is explained in steps as follows:

Procedure for encryption

Step-1: Input plain text in Matrix form
Step-2: Partitions input block into two halves mxn [(mxn/2) and (mxn/2)]
2.1 Processing the Matrix value
2.1.1 Perform a substitution on left data half.
2.1.2 Based on round function of right half and sub-key.
2.1.3 Then have permutation swapping halves.
Step-3: Then the two halves pass through n rounds of processing then combine to produce the cipher block.
Step-4: Each round i has as input L_{i-1} & R_{i-1} derived from the previous round as well as a sub-key ki derived from the overall K.
Step-5: Computation is done for each round.
Step-6: A substitution is performed on the left half of the data.
Step-7: XOR the output of that function and the left half of the data.

The novel encryption mechanism is incorporated in the system model using the cipher key C_k and has been discussed in detail in this section. The algorithm for encryption is as shown below section.

ENCRYPTION MECHANISM

The encryption mechanism for the proposed model is explained below using enhanced modified feistel network structure.

Modified Feistel network structure

The Matrix L_{mxn} which is a concatenated value of m1 || m3 is considered as the left value of the feistel network structure and Matrix R_{mxn} = m2 || m4 is considered as the right value of the feistel network structure. Using MD5 cryptographic hash algorithm the key matrix K_{mxn} is generated whose size is m x n where "m" is the number of attributes of metadata and "n" is the size of the MD_5 algorithm. The development of the cipher key in the feistel network is done through the number of rounds until the condition is satisfied. In this symmetric block ciphers, matrix obfuscation operations are performed in multiple rounds using the key matrix and the right side value of the feistel network structure. The function F plays a very important role in deciding the security of block ciphers. The concatenated value of L_{mxn} and R_{mxn} in the last round will be the cipher key C_{mxn}. Figure-2 below represents the one round modified feistel network structure.

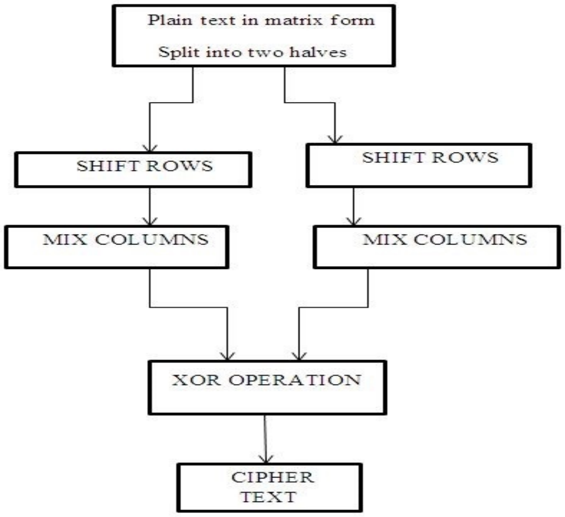

Figure-2. One round of modified Feistel network.

The cipher key generation mechanism has been discussed in detail in this section. The cipher key C_k has been generated using enhanced matrix based feistel cipher key algorithm.

Enhanced modified matrix based Feistel Network Algorithm

The below section describes the Feistel Function F which plays a major role in the proposed encryption technique.

Algorithm for creation of Feistel function F:

```
1       Read P, K
2.      $P_0$ = Left half of P.
3.      $Q_0$ = Right half of P.
4.         for i = 1 to r
              begin
                 $P_{i-1}$ = Mix $P_{i-1}$
                 $P_{i-1}$ = $P_{i-1}$ K
$Q_{i-1}$ = Shift ($Q_{i-1}$)
                 $Q_{i-1}$ = Mix $Q_{i-1}$
                 $Q_{i-1}$ = $Q_{i-1}$ K
                 ($P_i$, $Q_i$) = Shuffle ($P_{i-1}$, $Q_{i-1}$)
                 End
5.      C = $P_r$ ||$Q_r$
         /* || represents concatenation */

6.      Write(C) Cipher key C = L $_{mxn}$ || R $_{mxn}$   / * || represents concatenation */
        End
```

During file retrieval homomorphic encryption allows specific types of computations to be carried out on the corresponding cipher text. The result is the cipher text of the result of the same operations performed on the plaintext. That is, homomorphic encryption allows computation of cipher text without knowing anything about the plaintext to get the correct encrypted result.

IMPLEMENTATION DETAILS

The proposed model is analysed by executing set of experiments. The experiments are carried out in a cloud setup using eucalyptus which contains cloud controller and walrus as storage controller on a 5 node cluster. Each node has two 3.06 GHz Intel (R) Core TM Processors, i-7 2600, CPU @ 3.40GHZ, 4 GB of memory and 512 GB hard disks, running eucalyptus. The federated cloud network environment is created by installing the cloud controller in 5 physical systems. KDD Cup 2003 dataset is used for our experiments. The experimental results are as shown in Figure-3 and Figure-4. Performance analysis is done based on the experimental set up. The discussion of avalanche effect is as shown in Table-1. Avalanche effect is that by changing only one bit in a matrix, leads to a large change in the existing key, hence it is hard to perform an analysis of cipher text, when trying to come up with an attack. Higher the avalanche effect, higher the strength of the cipher key. The avalanche effect is calculated by the formula,

$$\text{Avalanche Effect} = \frac{\text{Number of values changed in the cipher Key } C_k}{\text{Total number of values in the cipher Key } C_k}$$

Table-1. Comparison of avalanche effect of cipher key.

File ID	DES	AES	BlowFish	EFCA
3	0.2833	0.3833	0.53	0.77
4	0.3979	0.4979	0.69	0.87
5	0.527	0.627	0.727	0.927
7	0.477	0.577	0.57	0.897
12	0.3	0.4	0.59	0.8083
15	0.4	0.5	0.541	0.851
20	0.3	0.4	0.725	0.8125
28	0.5	0.6	0.66	0.866
39	0.4	0.5	0.75	0.9375
40	0.5	0.65	0.7	0.895
51	0.21	0.31	0.421	0.7721

Figure-3 and Figure-4, shows the time taken for encryption at the sender cloud and the time taken for decryption at the receiver end in a federated network using enhanced modified feistel network algorithm. From Figure-3 and Figure-4, it is observed that the time taken for encryption using EFCA is less when compared to the existing encryption algorithms as the number of rounds taken for executing the proposed algorithm is less when compared to existing encryption algorithms.

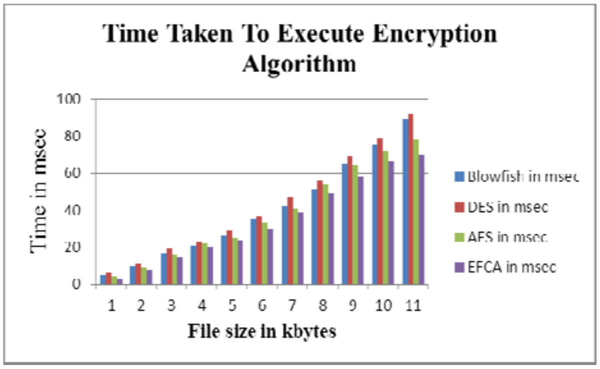

Figure-3. Comparison of time taken executing encryption algorithm.

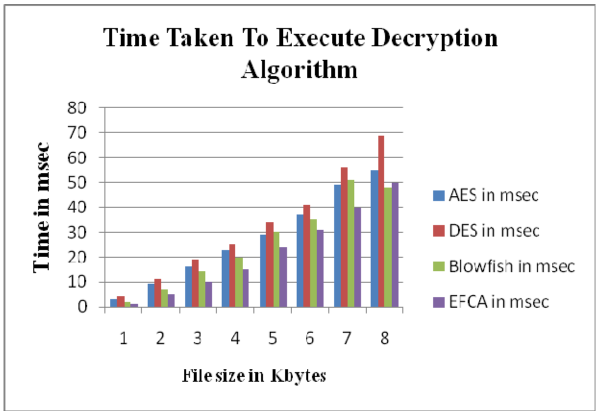

Figure-4. Comparison of time taken executing decryption algorithm.

CONCLUSIONS

This paper motivates and solves the problem of data security in federated cloud environment using homomorphic encryption technique. The proposed homomorphic based encrypted technique preserves the data from invisibly leaking the sensitive information. The novel key distribution mechanism around the federated cloud, devise a new technology which makes the data owner and the CSP's confident on the security of the data stored in cloud environment, since the encryption and decryption keys cannot be compromised without the involvement of all the clouds in the federated network. By security analysis, we show that the proposed scheme guarantees data privacy. According to the efficiency evaluation of the proposed scheme over real dataset, extensive experimental results demonstrate that our scheme ensures practical efficiency.

REFERENCES

C. Cachin, I. Keidar and A. Shraer, Trusting the cloud, ACM SIGACT News, Vol. 40, pp. 81-86, 2009.

J. Hendricks, G. R. Ganger and M. K. Reiter, Low-overhead byzantine fault-tolerant storage, Proceedings of twenty-first ACM SIGOPS symposium on Operating systems principles, ACM, pp. 73-86, 2007.

Chirag Modi, Dhiren Patel, Bhavesh Borisaniya, Avi Patel, and Muttukrishnan Rajarajan, "A survey on security issues and solutions at different layers of Cloud computing", Journal of Super Computers, pp.561–592, 2013.

J. Ravi kumar and M. Revati, "Efficient Data Storage and Security in Cloud ", In Proc. International Journal of Emerging trends in Engineering and Development, vol.6, no.2, 2012.

Shizuka Kaneko, Toshiyuki Amagasa and Chiemi Watanabe, "Semi-Shuffled BF: Performance Improvement of a Privacy-Preserving Query Method for a DaaS Model using a Bloom filter", in Proc. International Conference on Parallel and Distributed Processing Techniques and Applications, 2011.

Aguilera, M. K, Lillibridge.m and Maccormick, "Block-Level Security for Network-attached disks", In Proc. The 2nd Usenix conference on File and Storage Technologies, pp.159–174, 2003.

Anitha, R, Pradeeban Paramjothi, and Saswati Mukherjee, "Security as a Service using Data Steganography in Cloud Computing", in Proc. of International Conference on Cloud Security Management, pp. 81-89, 2013.

Sujitha. G, Rajeswaran, Thiagarajan, Vidya. K, Mercy Shalinie. S, "Preserving Privacy of Cloud Data Using Homomorphic Encryption in MapReduce, "International Journal of Hybrid Information Technology, vol. 7, no. 3, pp. 363-376, 2014.

C. Orencik, E. Sava, Efficient and Secure Ranked Multi-Keyword Search on Encrypted Cloud Data, in Proc. of EDBT- ICDT, pp.186 -195, ACM: New York, USA, 2012.

Craig Gentry, A Fully Homomorphic Encryption Scheme, http://crypto.stanford.edu/craig/craig-thesis.pdf, 2009.