ASIAN JOURNAL OF PHARMACEUTICAL AND CLINICAL RESEARCH

INNOVARE ACADEMIC SCIENCES
Knowledge to Innovation

Special Issue (April)

Online - 2455-3891
Print - 0974-2441

Full Proceeding Paper

# CRYPTOGRAPHIC TECHNIQUES FOR SECURE KEY MANAGEMENT IN PERSONNEL CLOUD

VEDIKA B*, ABDUL QUADIR*

Department of Master of Computer Application, School of Computing Science and Engineering, VIT University, Chennai, Tamil Nadu, India.
Email: b.vedika2016@vitstudent.ac.in/abdulquadir.md@vit.ac.in

*Received: 19 January 2017, Revised and Accepted: 20 February 2017*

## ABSTRACT

Cloud computing has become an emerging model of information technology industry as it can be accessed anywhere in the world on a pay-per-use basis. However, one major problem it is facing in today's challenging world is the security issues. Whenever the data are transferred through different connected networks, the threat is there for the users that their data might get leaked from outsiders or unauthorized people. In this study, we propose the solution for the above-mentioned problem. We go through the cryptographic techniques by combining two messages into encrypted data and then sending them to the destination. We also propose a new method of exchanging the keys and to study how the keys get interact at the user and system level.

**Keywords:** Cloud computing, Key management, Challenges and securities, Cryptographic techniques, Encryption and decryption, Combination of two messages.

## INTRODUCTION

Since cloud computing services are highly in demand nowadays, each and every information technology (IT) industry relies upon these services to expand their organizations.

Virtualization storing of data in cloud led to high-profile data losses which have act as an incentive for the use of encryption in the IT world. Several files are stored virtually which must be provided with secure retrieval of data. To maintain the standard of confidentiality, cryptographic techniques are the useful resources for the service providers.

Since data are stored with the third party, that is, cloud service provider (CSP), it increases the threat of data leakage, as the data can be misled to unwanted usage.

Hence, to provide users with greater and secure confidentiality of data, we use the encryption methods by combining two messages into a single encrypted message. In this research, each encrypted key is distributed among different servers so that our data will be secured. If in case, our key gets lost, we can easily restore that key values from these different parties again.

These cryptographic keys are distributed among users through multiple servers located at different locations. This will prevent the latency of threat level (Fig. 1).

The purpose of our project is to provide the detailed view of key management scheme using paired keys and how the key is distributed at all levels to reach the user.

In this paper, we present the whole cryptographic scenario and the survey of how to prevent security threats by combining two different messages and providing a technique and re-encryption strategy for future use.

In the next section, we will discuss some security challenges we face during key management and survey on with various techniques with their drawbacks followed by their solution suggested.

## LITERATURE SURVEY

### Security challenges

*Data breaches*

Due to huge storage of data in cloud, service providers became the prime target of attacking.

*Compromised credentials and broken authentication*

Weak mechanism of authentication and devices can be easily cracked.

*Malicious insider*

They can destroy the whole data and can manipulate data in a wrong way which can be a devastating effect for the organization.

*Permanent data loss*

Data stored in mobile phones or any other machines should be secured if the device gets lost. They should not be mistreated by any other user who found it.

### Security problems

*Authentication and authorization*

Data are stored online and these sensitive data can be accessed from any corner of the world. Hence, there should be identification check for the user who is authorizing the data. Lack of strong authentication and authorization has become a major concern in this field.

*Unavailability of data*

It can be possible that after applying encryption to the data, one can forget the key password or it may be lost accidentally. In such cases, effective restoration of our data is again a major issue.

*Sharing of keys*

Sending of crucial data and information to the end user by sharing keys can generate a threat level to the users. These keys can be used illegally which can damage the organizations' security network, creating a huge loss for them.

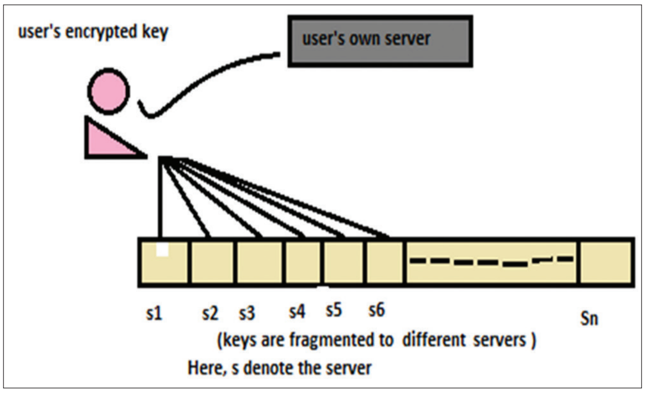

user's encrypted key
user's own server
s1
s2
s3
s4
s5
s6
Sn
(keys are fragmented to different servers )
Here, s denote the server

**Fig. 1: Distribution of keys**

Various cryptographic techniques are taken into consideration. Since data stored on the cloud are so sensible, they must be protected from the unaccredited user.

The most commonly used technique is public key encryption which is also known as asymmetric cryptography; in these, public keys are distributed and are paired with the private keys which are known only to the owner. Two functions which are involved in this method are: First, plain text is authenticated using a public owner who has the accessibility of having a private key, and second, that encrypted cipher text must be decrypted only through that user who is holding that private key.

Another mechanism used is symmetric algorithms for cryptography in which two similar keys are shared between the users which carry the vital piece of information link. Since they are shared among different parties, the threat level also increases accordingly.

As in the Institute of Electrical and Electronics Engineers [9], user-centric-key-management can be used to secure our data from the third party, CSP. This will maintain the integrity of the data more efficiently. Enhanced Shamir's algorithm [10] will split the encryption key into N pieces:

Key 1, key 2, key 3, key 4, key 5…key N

This will store each piece into M storage servers such that each server storage contains only one piece of key. This function is given as follows:

$$\sum_{m=1}^{m=n-1} M = N - 1 \tag{1}$$

If the length of the key is less than the required length, it will assign some special character. Here, N depends on the size of available disks or server to store the data.

To handle the large computing storage of data, system model [4] can be designed where the server system directly communicates with the cryptographic key management system layer [4] which is responsible for secure accessing of data and key management.

Public verification for key can also be done using [3] Robert secret-sharing scheme where knowing both global keys and (n−1) shares would not retrieve us the plain text that was being shared. In RSA algorithm [10], each piece of key is in parts such that it cannot be computed without the secret or private key owned with the owner itself, and the encryption cannot be executed if its exceeds the threshold value (t).

In a study by Wang *et al*. [4], rekeying strategy is used in which key is again encrypted after the user uses the data such that it can be more secured while transferring of data, hence protecting their confidential information. In a study by Chen and Yang [11], exclusion basis system is discussed where multiple users are indulged in sharing the data; at the same time, encryption is imposed on each message by logical statement which enables them to restrict to a subset of users participating simultaneously. However, this technique has a drawback of collusion. In proxy re-encryption schemes [12], cipher text is shared with the use of one secret key, in which third parties encrypt the message for another party such that they can decrypt the cipher text.

In attribute-based encryption (ABE), dataset is related with a set of attributes and these data can only be decrypted if they satisfy the attributes of the access policies attached to the key of the owner. In a study by Jia *et al*. [14], using multicast group key management (GKM), the information between the members of the secured group is kept confidential assigning them with some protocols. In a study by Nabeel *et al*. [15], with hybrid encryption, we can encrypt our data using ABE or proxy server using public key with re-encryption method.

In broadcast GKM (BGKM) [15], secret key is shared among user using identity-based attribute. We use symmetric keys for exchanging confidential data. We can even add more number of users participating in BGKM circle and can update their access control policies also. The user can only able to access its data which are enabled to its domain only using the keys associated with it.

In a study by Li *et al*. [16], using de-duplication technique, we reduce the data storage space, hence making our key management more scalable [16]. Convergent encryption uses a convergent key to encrypt the data which is attained by the hash value of the summary of the file copy to the physical level. Here, users retain the keys with themselves and then send the encrypted files to the online cloud storage. De-key enhances our file storage as well as stops our data from being corrupted and provides a feature of backup storage also. In a study by Binbusayyis and Zhang [13], using cipher text-policy-ABE scheme, a delegation access service to the user can be provided to reduce the load on the centralized storage cloud and authority. ABE protects the confidentiality of our data.

In [17] Tysowski proposed an architecture where keys are divided and portion of keys are stored in the cloud which are instinctively removed based on transmission time or user's action. Instead of sharing files directly, we share the keys of encrypted message to them so that we can be assured of any threat attacks by malicious users online.

## PROPOSED WORK

In this paper, we focus on cipher methodology by combining two messages into a form which cannot be regained by an outsider and re-keying strategy for strong security architecture. We will see the overall flow of these techniques from source to destination level.

We will discuss in detail of how to use these techniques. Using these techniques, we achieve more secured access to data.

In our proposed work, the user before sending the message will first generate the encrypted message (m3) by combining the two messages (m1 and m2). Then, that encrypted message (m3) will further be encrypted through a re-keying strategy and will be sent to different shareholders or parties. At least one key will remain with the owner itself for more secured accessing of data. We then apply the re-keying strategy to this encrypted message using a secret key. If an unwanted user tries to access the data, he/she has to first know the original message from which we have encrypted our data and also he/she needs to collect all the distributed keys shared among shareholders or parties holding the keys.

This proposed architecture will provide us strong security access structure while transferring the data through different nodes (Fig. 2a and b).

## SCHEME FOR COMBINING TWO MESSAGES

Requirements for combining two messages and thus providing the users strong and more feasible techniques are as follows:

a. User should have one meaningless datum in the form of message m2 to combine it with the real message m1.
b. One secret key to re-encrypt the data again so that it cannot be easily cracked by malicious user who tries to access the data illegally.

Let us take function for messages m1 and m2 by:

$$m1 = \{x_{m1}, 1\} \tag{2}$$

$$m2 = \{x_{m2}, 1\} \tag{3}$$

Set a threshold value (t) according to your data access structure. It is given by:

$$\text{Threshold value (t)} \leq \text{Number of parties (n)}-1] \tag{4}$$

Such that, it is possible to decrypt the message and recover the data only if all parties participated in the encryption.

Finally, we generate the encrypted message, let us take a polynomial P(x) where $x \leq t$ (value).

$$P(x) = \{x_{m1} \text{ Union } x_{m2}\} \tag{5}$$

In the result, polynomial value is equal to ($x_{m3}$), secret key= K1. The equation is as follows:

$$x_{m3} = \text{ReEncrypt (Union K1)} \tag{6}$$

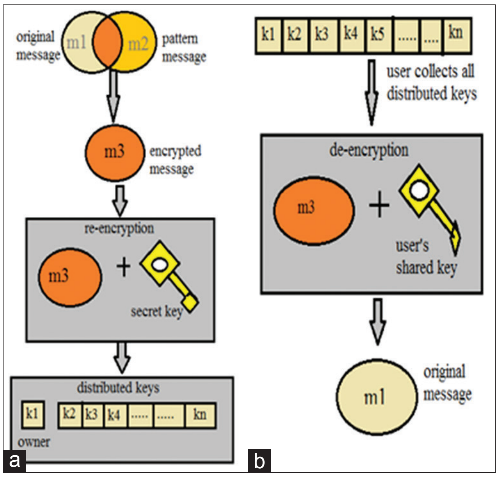


**Fig. 2: Proposed architecture of combining two messages into an encrypted message. (a) Encryption. (b) Decryption**

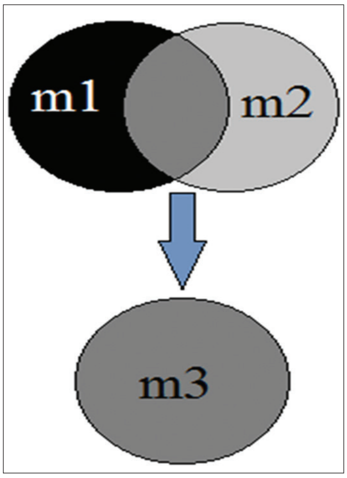


**Fig. 3: Combining two messages**

Let us take an example of colors, if we combine black with white color, it will result into gray color. Like this, we will generate the third message using two messages m1 and m2 to form message m3 (Fig. 3).

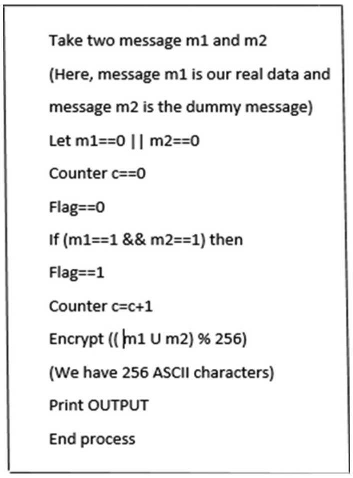
### Re-keying strategy

In re-encryption, the message is encrypted by a third party service provider or proxy server for another party. They can decrypt the message using the key which is being generated by the proxy server. In this way, we can hide our crucial information from the third party or proxy server by not sending them simple text data directly.

### De-keying strategy

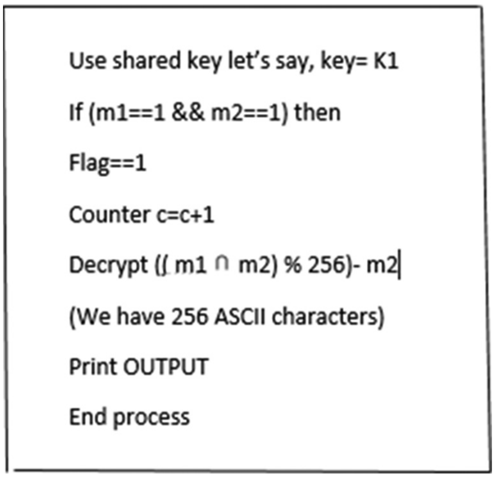
In de-encryption, the message is retrieved back to plain text using a secret key accessible to the user only. De-encryption method allows us to recover our files in a very quick time with much secure manner and we can also save our files from being corrupted. The data can also be restored using cryptographic keys.

### Encryption algorithm of combining two messages



```
Take two message m1 and m2
(Here, message m1 is our real data and
message m2 is the dummy message)
Let m1==0 || m2==0
Counter c==0
Flag==0
If (m1==1 && m2==1) then
Flag==1
Counter c=c+1
Encrypt (( m1 U m2) % 256)
(We have 256 ASCII characters)
Print OUTPUT
End process
```

### Decryption algorithm of extracting the original message

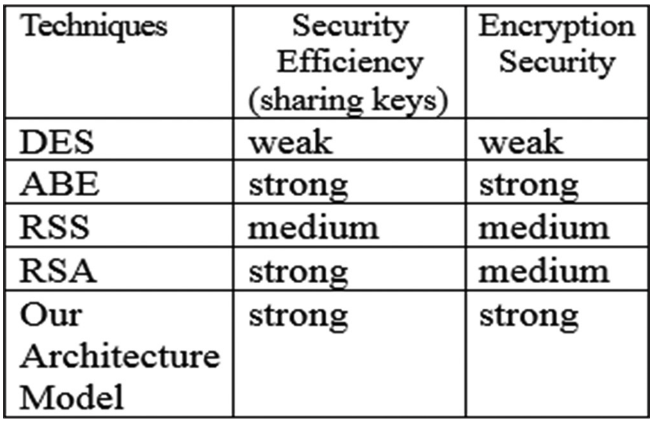


```
Use shared key let's say, key= K1
If (m1==1 && m2==1) then
Flag==1
Counter c=c+1
Decrypt (( m1 ∩ m2) % 256)- m2
(We have 256 ASCII characters)
Print OUTPUT
End process
```

One crucial challenge faced by encryption key management is standard entitled enterprise encryption and key management strategy from enterprise strategy group.

## PERFORMANCE EVALUATION

We have observed that the existing techniques based on the analysis of literature survey cryptographic techniques have some drawbacks of sharing keys, low security of accessing the crucial data, and sometimes recovery is not available if the key is lost by the user.



| Techniques | Security Efficiency (sharing keys) | Encryption Security |
|---|---|---|
| DES | weak | weak |
| ABE | strong | strong |
| RSS | medium | medium |
| RSA | strong | medium |
| Our Architecture Model | strong | strong |

**Fig. 4: Comparison between different cryptographic techniques**

To overcome that challenges faced in cloud key management, we have used the threshold value (t) for secure retrieval of our data for strong data access structure. Since the unauthorized user does not know the original messages m1 and m2, our data could not be misused by malicious user. Using the concept of re-keying technology, we have strengthened our proposed work for more secure management. As it is very difficult for one to keep an eye on different distributed keys simultaneously, it becomes almost unreachable for the unauthorized user to crack into our system and take the information of an organization. Thus, the proposed model in this paper presents improved key management architecture (Fig. 4).

## CONCLUSION

For cloud computing, we proposed a scheme that provides secure data accessing for users in large scale who uses cloud for storage of their sensitive information. In this paper, we depicted the use of key management in cloud computing by combining two messages and then again by applying the concept of re-keying strategy. Our proposed scheme would surely help users to secure against chosen cipher text attacks.

## REFERENCES

1. Pradeep KV, Vijayakumar V. Survey on the key management for securing the cloud. Procedia Comput Sci 2015;50:115-21.
2. Lei S, Zishan D, Jindi G. Research on key management infrastructure in cloud computing environment. Grid and Cooperative Computing (GCC), 2010. 9th International Conference on. IEEE; 2010.
3. Damgård I, Jakobsen TP, Nielsen JB, Pagter JI. Secure Key Management in the Cloud. Oxford, UK: IMACC; 2013.
4. Wang Y, Li Z, Sun Y. Cloud Computing Key Management Mechanism For Cloud Storage; 2015.
5. Shabir M, Iqbal A, Mahmood Z, Ghafoor A. Analysis of classical encryption techniques in cloud computing. Tsinghua Sci Technol 2016;21(1): 102-13.
6. Kumar NS, Lakshmi GV, Balamurugan B. Enhanced attribute based encryption for cloud computing. Procedia Comput Sci 2015;46:689-96.
7. Yan L, Rong C, Zhao G. Strengthen cloud computing security with federal identity management using hierarchical identity-based cryptography. IEEE International Conference on Cloud Computing. Berlin, Heidelberg: Springer; 2009.
8. Ye X, Chen X, Wang H, Zeng X, Shao G, Yin X, *et al*. An anomalous behavior detection model in cloud computing. Tsinghua Sci Technol 2016;21(3):322-32.
9. Yoo SM, Park PK, Shin JS, Oh JS, Ryu HY, Ryou Jc, *et al*. User-centric key management scheme for personal cloud storage. Information Science and Applications (ICISA), 2013 International Conference on. IEEE; 2013.
10. Fakhar F, Shibli MA. Management of Symmetric Cryptographic Keys in Cloud Based Environment; 2013.
11. Chen Y, Yang G. Efficient and Secure Group Key Management Based on EBS and Attribute Encryption; 2011.
12. Tysowski PK, HasanMA. Senior member, hybrid attribute-and re-encryption-based key management for secure and scalable mobile applications in clouds. IEEE Trans Cloud Comput 2013;1(2):172-86.
13. Binbusayyis A, Zhang N. Decentralized Attribute Based Encryption Scheme with Scalable Revocation for Sharing Data in Public Cloud Servers; 2015.
14. Jia H, Chen Y, Mao X, Dou R. Efficient and Scalable Multicast Key Management Using Attribute Based Encryption; 2010.
15. Nabeel M, Shang N, Fellow E. Privacy Preserving Policy Based Content Sharing in Public Clouds; 2013.
16. Li J, Chen X, Li M, Li J, Lee PP, Lou W. Secure deduplication with efficient and reliable convergent key management. IEEE Trans Parallel Distrib Syst 2014;25(6):1615-25.
17. Tysowski PK, Hasan MA. Cloud-Hosted Key Sharing Towards Secure and Scalable Mobile Applications in Clouds; 2013.