Management of Symmetric Cryptographic Keys in Cloud Based Environment

Faiza Fakhar*, Muhammad Awais Shibli

*School of Electrical Engineering & Computer Science, National University of Science & Technology, Islamabad, Pakistan.

10msitffakhar@seecs.edu.pk , awais.shibli@seecs.edu.pk

***Abstract*— Although, cloud computing provides innumerable benefits to its customers but it fails to solve information security concerns especially in public cloud. Symmetric Cryptographic Key is sensitive data and it is required to be stored at cloud platform to solve several problems of encrypted data such as searching/manipulation on encrypted data. This paper presents a technique that will manage symmetric cryptographic keys on cloud-based environment. Proposed technique is based on secret splitting technique enhanced Shamir's algorithm. Proposed technique is implemented in OpenStack private cloud environment for performance analysis and it is found out that the technique works efficiently. In addition, it is fulfilling the best practices metrics, given by National Institute of Standard and Technology.**

***Keywords* – Public Key Cryptographic Standard (PKCS7); Public Key Infrastructure (PKI); Cryptographic Key Management (CKM); National Institute of Standard & Technology (NIST); Secure Shell (SSH).**

I. INTRODUCTION

Due to rapid increase in technology, business trends and technologies are varying day by day. To compete and to formulate inimitable position globally; corporations require extra effort in discovering and adapting to new technologies. An emerging paradigm in next generation computing is Cloud Paradigm, which is an enhancement of grid or distributed computing. Cloud computing provides several benefits to its consumer such as availability, flexible cost model, on demand self services, elastic resources, etc. It facilitates their consumers by providing software, platform and infrastructure as service. Thus, users of cloud do not have to be concerned about the complexities of software, hardware or infrastructure. Furthermore, increasing bandwidth and trustworthy network connection make it possible for businesses to subscribe to high quality services according to their requirements.

While adopting cloud paradigm few concerns must be addressed. One of them is information security, which is an imperative element of quality of service. Since the consumer of cloud does not have access to the physical storage/servers, cloud multitenant environment and cloud provider's administrative access on all data causes threats for integrity and privacy of sensitive information. Traditional information security mechanism such as cryptography, digital signature, access control mechanism, trust management are not sufficient to fulfil the need of information security on cloud paradigm especially in public clouds, due to the limitation of other security factors. These factors include local laws and jurisdiction concerns, as cloud server exist on different locations. Moreover, Cloud Security Alliance (CSA), the global leaders in cloud security provide a security guidance [1] about best practices in cloud computing. It also provides suggestion on critical area of cloud computing. In domain eleven of this guidance, they identified that the cryptographic key management at public or hybrid cloud is a challenge and cryptographic keys should be stored on enterprise domain due to their security. However, searching of encrypted data from a large data set is problematic if cryptographic keys are stored at enterprise premises. Furthermore, cryptographic key management is required at cloud when any application process requires working on plain text at cloud platform and data must access cryptographic key.

Cryptographic key management includes all operations that can be performed on cryptographic key except encryption/decryption. These operations comprise but are not limited to generation, revocation, sharing and storage of cryptographic keys. One possible solution towards cryptographic key management for cloud is, to download data on client terminals for appropriate operation, and after that operation, upload data back on cloud server. This solution deviates from the benefits of cloud paradigm since all computations are performed on client machine. In addition, there is an overhead involved in upload and download. Section II will provide a detail analysis of other existing techniques related to cryptographic key management for cloud epitome.

In this research, we proposed an effective, robust and incorporate able security protocol for symmetric cryptographic key management in cloud based environment. This technique is based on secret splitting and on the fly computation mechanism. This paper contributes in following three aspects:

1) Provide a mechanism for secure storage of sensitive data on public/private/hybrid cloud. This storage scheme can be further utilized to store sensitive data other than cryptographic keys.
2) Provide symmetric cryptographic key as cloud service and user may embed this service in other

utilities such as mobile/PDA's digital signature utilities etc.

3) On the fly computation of cryptographic key on client terminal will ensure key access security.

Rest of the paper is organized as follows: section II discusses related work of symmetric Cryptographic Key Management (CKM) in cloud-based environment. Detail of proposed symmetric CKM in cloud-based environment is discussed in section III. Several avenues for performance evaluation proposed by Nation Institute of Standard & Technology (NIST) [2] are identified in section IV. Performance evaluation and implementation details are discussed in section V. Finally, Section VI concludes this research.

II. LITERATURE REVIEW

Extensive research in cloud security includes several techniques for information security based on traditional security mechanisms such as cryptography, homomorphism encryption techniques, private information retrieval etc. However, strong algorithms and cryptographic key management are two main aspects of all proposed techniques.

Re-Encryption based key management scheme is presented in [3] that is used in cloud based mobile application. Core concept of this scheme is the deployment of manager that can be an entity such as secure server between mobile device and cloud. Manager communicates to cloud as well as end users. It issue public, private key pairs for each user and maintain Access Control List (ACL) to enforce authorization. Public key repository for all users is maintained on cloud and any one from system user can access it but cannot decode it (as all private keys are maintaining by server). Users use their private key to encrypt any request and upload the cipher on cloud. Other users who require the data make a request to cloud controller. Cloud controller sends that request to manger. Based on Access Control List, manger decides to give the control of data to any user. First, it fetches the data from cloud storage and decrypts it using private key of the sender. Then manger will re-encrypt data with requester public key and send cipher text to requester. Requester decrypts the data with his/her private key and same process continues.

In this technique, single authority of control such as manager can be a bottleneck in the design of the scheme. Therefore, key benefits of cloud such as elasticity, prevention of denial of service attacks cannot be fully utilized. Furthermore, a secure hardware and deployment environment is required for manager server. Manger contains private keys of users, so it should be fully secured with each aspect. At each data retrieval request, extra processing such as re-encryption will be required that would increase the processing and retrieval time.

A research at Microsoft in [4] proposes and implements a solution for distributed key management that includes all the operation on cryptographic keys as well as cryptographic algorithms agility. The main idea is to store keys at any distributed access control repository such as Microsoft Active Directory. However, due to the TAP theorem Microsoft active directory architecture are limited to provide availability and partition tolerance property and not the atomicity property. An age based key-life cycle management is used that automatically generates, activates and expires cryptographic-keys based on end user data operations. A policy file has the information about key length, lifetime, crypto algorithm, etc. User will be unable to see any key until it is fully distributed. Furthermore, due to lake of consistency many issues on key may arise.

StrongAuth has StrongAuth Key Appliance in [5] that uses third party library to develop an enterprise Key management Infrastructure, which support the services of public key infrastructure (PKI) as well as provides symmetric key management libraries. However, this library does not include any features that can securely manage keys at cloud platform. It requires a separate server for key storage and compromise of this server can create bottleneck for key security.

A scheme progressive elliptic curve encryption is presented in [6] that used multiple encryption keys to encrypt a part of data multiple time such that final cipher can be decrypted in one run using single key. This scheme is based on public/private key cryptography and consumers of application manage cryptographic private keys. Furthermore, N re-encryption will be required for N users in case of single data piece sharing.

Key management for database in cloud base environment is presented in [7]. Homomorphism in joint encryption scheme [8] is used in which subset of a group of users can encrypt/decrypt data. This scheme presents an idea that how an encrypted database can be shared among different users in cloud environment.

A report in [9] presented the summary of workshop organized by National Institute of Standard and technology (NIST) on cryptographic key management. This report summarise the talks and propose techniques of different speakers on cryptographic key management. Some speaker such as Vijay Bharadwaj from Microsoft, Lee Badger from NIST and Miles Smid from Orion Security Solutions, highlights the need of secure cryptographic key management system on cloud paradigm and its implications in their presentations. Furthermore, new technology section of this report also identifies that a cloud based key management system is required that can be used to securely communicate and provide trusted services for all its consumers.

National Institute of Standard and technology draft in [10] provide guidelines on security and privacy in public cloud computing. They highlight key security and privacy issues in public cloud environment. One of them is data isolation and cryptographic key management at the premises of public cloud provider.

There are several secret splitting algorithms. Shamir's secret splitting [11] and Rabin [12] secret

splitting are the core of them. Shamir's algorithm split a secret in N parts with K threshold value. It distributes these N parts among N users so that each user has only one part. Secret can be regenerated if K out of N pieces are available. In Rabin algorithm secret is divided in N pieces and can be regenerate if K<N. It reduces storage complexity of the secret but security may be compromised in case of repeating patterns [13]. A survey research on both algorithms performance in cloud environment is conducted in [14]. They claimed distributed nature of cloud paradigm is more suitable for secret splitting than encryption. Furthermore, cryptographic keys and other secure data can be best handled with secret sharing on cloud paradigm.

A scheme on distributed key management is proposed in [15], which uses RSA algorithm for encryption/decryption. The main concept is to split key in multiple parts and divide among the group of users. If all users work on same text, a cipher text can be generated that will be equal to the cipher generated by actual key.

III. PROPOSED SYMMETRIC CKM PROTOCOL FOR CLOUD BASED ENVIRONMENT

Literature review highlights the fact that providing cryptographic key management at cloud is problematic.

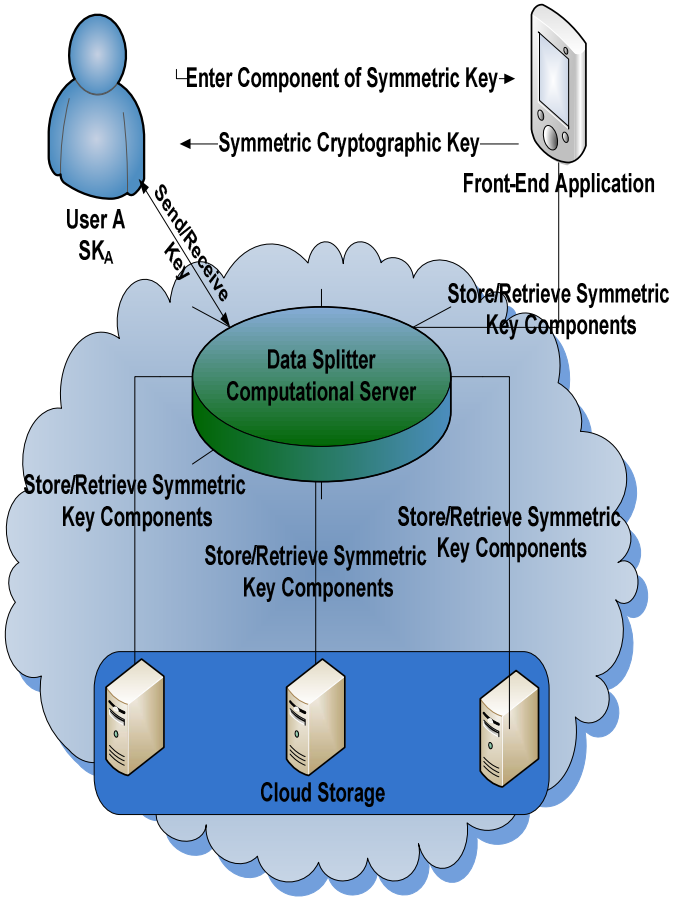

Figure 1: ARCHITECTURE OF PROPOSED PROTOCOL

This research provides a protocol that will generate, store, distribute and revoke symmetric cryptographic key as per consumer requirement on cloud platform. Figure 1 shows a high-level architecture for symmetric cryptographic key management.

A. *Cryptographic Key Generation/Storage for Proposed Protocol*

User can generate new symmetric cryptographic key K or can store already existing cryptographic keys as required using proposed technique. Key splitter in figure 1 will use *Enhance Shamir's Algorithm* to split key K in N pieces

$$K_1, K_2, \ldots, K_n.$$

Proposed scheme store each piece on J storage disks such that each storage contain only one piece of K and

$$\sum_{J=1}^{J=n-1} J = N - 1$$

One main piece lets K_n of key will be assigned to consumer of application. This piece of key has information of all other pieces and actual key cannot be regenerated without this piece. If key length is less than minimum length required it will padded by some special character. Size of N will depend on available disks to store data. All data will travel on network using secure shell (SSH) for safe communication. Key will be stored in a database that store key component with its specific id and sequence number. Single storage contains one component of generated key at the same time. Different vendor's storage services on cloud can also be used. For example, one component can be stored on one cloud vendor's storage. Other can be stored on other vendor's storage and so on.

B. *Key Transfer*

User can transfer completely computed key or the component of key on public cloud for data processing. Public Key Cryptographic Standard (PKCS7) will used to transfer such key that is developed by RSA Laboratories and used to wrap data in an envelope to securely transfer it. This protocol used to wrap message in an envelope and signed by sender. Receiver knows the decryption key to decrypt the encrypted message [18].

C. *Cryptographic Key Retrieval for Proposed Protocol*

On the request of key retrieval all, the components will fetch the key from key store through computational server, as shown in figure 1 and send to client terminal via Public Key Cryptographic Standard (PKCS7). Client machine will prompt consumer of application to enter his/her piece of key. Original Key will compute on the fly after taking information from consumer on consumer's terminal. During recalculation of cryptographic key, Enhanced Shamir's will recover all missing part and still regenerate original key if and only if count of missing part

will less than threshold value and main part of key will available that is own by key owner.
$K = K_1$ Operation K_2 Operation ,..., K_n

D. *Proposed Enhanced Shamir's Algorithm*

The goal of basic Shamir's algorithm is to divide cryptographic key K in n safe pieces

$K_1, K_2, ... K_n$

Such that knowledge of any J pieces can be used to compute K easily. These pieces are assigned to N nodes. J is known as threshold value for this algorithm and scheme is called (K, J) threshold scheme. Proposed enhancement in Shamir's algorithm is to divide Key in n parts $K_1, K_2, ... K_n$ such that there exist a special part K_t which contains the information of all other parts, and K cannot be computed without Kt. In addition, threshold is set to h, and there exist $K_1, K_2, .., K_h$ parts on key retrieval operation and does not include K_t part in it. However, K cannot be computed without especial part K_t unlike in original Shamir's algorithm. Figure 2 describes a high-level algorithm for proposed technique.

```
Input:
1. Secret Key to store
2. User own Key component
Output:
1. Complete Key in case of Key retrieval.
2. Key storage and User's Component in case of key storage
request.
Begin:
If(request== "keyStorage")
{
        Split key;
        Store on different storage;
        Display user's component of key to user;
}
Else if(request=="key transfer")
{
        Use PKCS7 protocol
}
Else if(request== "retrieval of key")
{
        Collect all components of specific key;
        Send to user machine;
        Take user's part of key;
        if(allPartsExist)
        {
                Generate actual key based on all information.
        }
        Else if(userPartExsit &&
storeedComponentSize==collectedComponentSize-1)
        {
                TrytoRecoverKey();
        }
}
DisplayOutput();
End
```

Figure 2: ALGORITHM OF PROPOSED TECHNIQUE

Multiple storage disks act as key store on cloud platform. These disks can be resides in same locality as well as in different locations as shown in figure 1. These disks will play the role of nodes used in Shamir's algorithm and are assigned single piece of key. There will be N-1 disks so that each disk D_i contains K_i where $0 > i \leq n-1$

IV. NIST ASSESSMENT CRITERIA & PROPOSED CKM

NIST provides a report in [2], which identifies best practices for cryptographic key management system. Some of them are following:

A. *Security Mechanism*

According to NIST any CKM should have the capability to manage security mechanism such as access control etc. Our proposed protocol manages access control by giving access to authorise data owner. It maintains integrity of data to some extant so that if any component of key other than user owned component is missing than key can still be recovered. Privacy of data is maintained by distributing key on different drives therefore, un-authorised user cannot retrieve the key completely.

B. *Key Management System*

Key management system includes all the operations that can be performed on cryptographic keys except encryption/decryption. These may include but not limited to storage, generation, distribution, retrieval of cryptographic keys. Our proposed system is performing all operations mentioned above successfully.

C. *Key Management System Survivability*

This aspects of ideal CKM evaluate the survivability of CKM in case of any misshape. Proposed system has ability to recover its original cryptographic key in case of misplacing a single component. However, that component should be other than user's owns component since without user's component key cannot be recalculated.

V. IMPLEMENTATION DETAILS & PERFORMANCE EVALUATION

Figure 3 describes a cloud model that used to test the proposed scheme. In this protocol, Client/Consumer/Application users are the end-users who want to use proposed protocol as a service. Cloud Providers are different vendors who own servers and computational environments. Cloud storage is different servers owned and managed by cloud provider.

OpenStack [16] cloud is an infrastructure based open source cloud. This private cloud is used to implement proposed scheme. This private cloud is setup on three Linux machines. One machine is used for controller and other two are used for object store and compute servers. Controller nodes consist of all required software such as Nova api, Network Controller etc. Data stores on three partitions are created in the form of database and deployed

in our propose protocol. It was demonstrated that no one can access Full key other than the owner of key as well as key can be regenerated if any component is missing except consumer owned component.

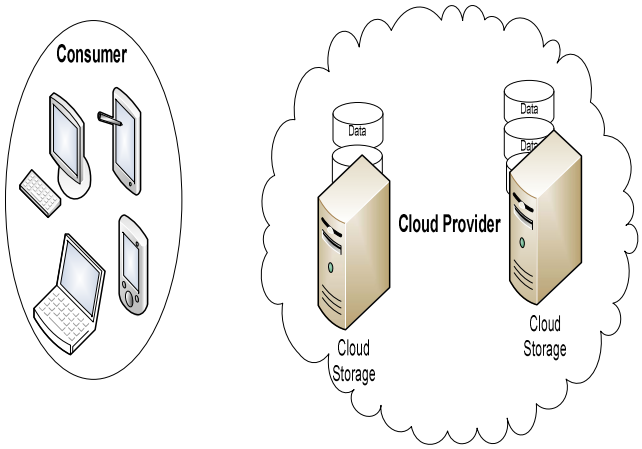

Figure 3: ARCHITECTURE OF CLOUD

Existing techniques on cryptographic key management on public cloud paradigm are not much successful. Some of them require re-encryption of keys, which not only increase the computation bourdon but also require some storage for re-encrypted key stores. Furthermore, it prevents to completely utilize the cloud benefits as cipher text generated from these keys cannot use on cloud platform directly. It requires some kind of access control mechanism. Proposed protocol is designed for large-scale data storage and fulfilled the design requirement given by NIST in [2].

An important element of cryptanalysis is available amount of information for attacker. In proposed scheme all data is dispersed on different servers/storage there are very less chances about data leakage. Key store contains millions of key component and it is extremely difficult to keep eye on a single key component and its Meta data. Therefore, a hacker cannot have much information about the key and its related materials. Furthermore, if somehow hacker gets all component of key from cloud storage, it would be impossible to regenerate cryptographic key without user owned component.

VI. CONCLUSION & FUTURE WORK

Sensitive data storage on cloud platform is challenging while adopting cloud services for data storage. Cryptographic keys are sensitive data and required on cloud platform in different cases but cannot store directly on cloud. This paper discuses symmetric key management on cloud based environment.

Proposed technique is based on secret splitting and use enhanced Shamir's algorithm for secret splitting. On the fly computation of cryptographic key, enable integrity and privacy concerns related to key management on cloud platform. National Institute of Standards and Technologies identifies good practices for cryptographic key management systems; proposed technique is fulfilling all these measures. In addition to this proposed technique is tested on given cloud architecture and open stack platform and achieve desire results. Future directions of this research are as follows:

- This research can used to develop public key infrastructure as service in cloud platform in future.
- Trust framework can also develop for public clouds using this research.

ACKNOWLEDGMENT

The author would like to thank Mr. Syed Rauf ul Hassan for his help to review menu script and his valuable discussion on proposed technique. Author also wants to say thank to SEECS, NUST for their financial and moral support.

REFERENCES

[1]. Cloud Security Alliance, "Security guidance for critical areas of focus in cloud computing version 3.0" visited on 26th December 2011 at http://www.cloudsecurityalliance.org/guidance/csaguide.v3.0.pdf"

[2]. Elaine Barker, Dennis Branstad, Santosh Chokhani and Miles Smid, "A Framework for Designing Cryptographic Key Management Systems," NIST draft Special publication 800-130 at Computer Security Division, National Institute of Standards and Technology, 15, June 2010.

[3]. Piotr K. Tysowski, M.Anwarual Hasan, "Re- Encryption-Based Key Management towards Secure and Scalable Mobile Applications in Clouds"

[4]. Tolga Acar, Mira Belenkiy, Carl Ellison, Lan Nguyen,"Key Management in Distributed Systems" research at Microsoft, 2010.

[5]. "An Introduction to Strong Key", white paper StrongAuth.Inc, October 2011.

[6]. Gansen Zhao, Chunming Rongy, Jin Liz, Feng Zhangx and Yong Tang, "Trusted Data Sharing over Untrusted Cloud Storage Providers," 2nd IEEE International Conference on Cloud Computing Technology and Science.

[7]. Nadia Bennani, Ernesto Damiani and Stelvio Cimato, "Toward cloud-based key management for outsourced databases," 2010 34th Annual IEEE Computer Software and Applications Conference Workshops.

[8]. R. Cramer, I. Damg°ard, and J. B. Nielsen. Multiparty computation from threshold homomorphic encryption. In B. Pfitzmann, editor, EUROCRYPT, volume 2045 of Lecture Notes in Computer Science, pages 280–299. Springer, 2001.

[9]. Elaine Barker, Dennis Branstad, Santosh Chokhani and Miles Smid, "Cryptographic Key management workshop summary," NIST Interagency report 7609 at Computer Security Division, National Institute of Standards and Technology, January 2010.

[10]. Wayne Jansen and Timothy Grance, "Guidelines on security and privacy in public cloud computing," NIST draft Special publication 800-144 at Computer Security Division, National Institute of Standards and Technology, January 2011.

[11] Shamir, A.: How to share a secret. In: Commun. ACM, vol. 22, no. 11, pp. 612–613 (1979)

[12] Rabin, M.O.: Efficient dispersal of information for security, load balancing, and fault tolerance. In:
Journal of The ACM 36(2), pp. 335–348 (1989)

[13]. Resch, Jason; Plank, James (February 15, 2011). "AONT-RS: Blending Security and Performance in Dispersed Storage Systems". Usenix FAST'11, 2011

[14]. S.Jaya Nirmala, S.Mary Saira Bhanu, Ahtesham Akhtar Patel, "A Comparative study of the secret sharing algorithms for secure data in the cloud," International Journal on Cloud Computing: Services and Architecture(IJCCSA),Vol.2, No.4, August 2012.
[15]. G. Zhao, S. Otenko, and D. Chadwick, "Distributed key Management for secure role based messaging," in Proceeding of The IEEE 20th International Conference on Advanced Information Networking and Applications (AINA2006), Vienna, Austria, April 2006.
[16]. http://www.openstack.org visit on 24th August 2012.
[17]. http://en.wikipedia.org/wiki/Cryptanalysis visit on 15th December 2012.
[18]. http://www.rsa.com/rsalabs/node.asp?id=2308 visited on 17th December 2012.