



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

Data Security in Cloud Using Key Aggregate Cryptosystem

Savita Kacheshwar. Dabhade, M.C. Kshirsagar,

M. E, Dept of Computer Engineering, Vishwabharti COE, Ahmednagar, Pune University, Maharashtra, India

Professor, Dept of Computer Engineering, Vishwabharti COE, Ahmednagar, Pune University, Maharashtra, India

ABSTRACT: Cloud technology is very constructive and useful in present new technological era, where a person uses the internet and the remote servers to give and maintain data as well as applications. These applications can be used by the end users using the cloud communications without need of any extra software installation. Moreover, the end users' data files can be accessed and manipulated from any other computer using the internet services. Despite the flexibility of data and application accessing and usage that cloud computing environments provide, there are many questions still coming up on how to gain a trusted environment that protect data and applications in clouds from hackers and intruders.

Cloud storage should be able to store and share data securely, efficiently, and flexibly with others in cloud storage. The expenses, complication involved generally increase with the number of the decryption keys to be shared. The encryption and decryption key are different in public key encryption. Since we are proposing new era of aggregate key cryptography concept. To construct constant length ciphertext is also one of significant task that we have to become visible. In this paper, we propose a simple, efficient, and publicly verifiable approach to ensure cloud data security while sharing between different users. Since we introduce here, aggregate key cryptography system. Cryptographic techniques are usually applied to address this data sharing issue.

KEYWORDS: Cloud storage, data sharing, aggregate key encryption, Steganography.

I. INTRODUCTION

Cloud computing is considered as the next step in the evolution of on-demand information technology which combines a set of existing and new techniques from research areas such as service-oriented architectures (SOA) and virtualization. With the rapid development of versatile cloud computing technology and services, it is routine for users to leverage cloud storage services to share data with others in a friend circle, e.g., Dropbox, Google Drive and AliCloud.

The shared data in cloud servers, however, usually contains users' sensitive information such as personal profile, financial data, health records, etc. and needs to be well protected. As the ownership of the data is separated from the administration of them, the cloud servers may migrate users' data to other cloud servers in outsourcing or share them in cloud searching. Therefore, it becomes a big challenge to protect the privacy of those shared data in cloud, especially in cross-cloud and big data environment. In order to meet this challenge, it is necessary to design a comprehensive solution to support user-defined authorization period and to provide fine-grained access control during this period [16][17].

Cryptography of data is mainly the mix up the content of the data, such as text, image, audio, video and so forth to make the data indecipherable, undetectable or worthless during transmission of data. It is known as termed Encryption. Main aim of cryptography system is to secure data from attacker. At the same time decryption is exact opposite process of getting back the original data from encrypted data, which reinstate the original data. To encrypt data at cloud storage, both symmetric and asymmetric key generation algorithms are used. But when we use these algorithms, it's having serious problems while managing gigantic database and its transaction.

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

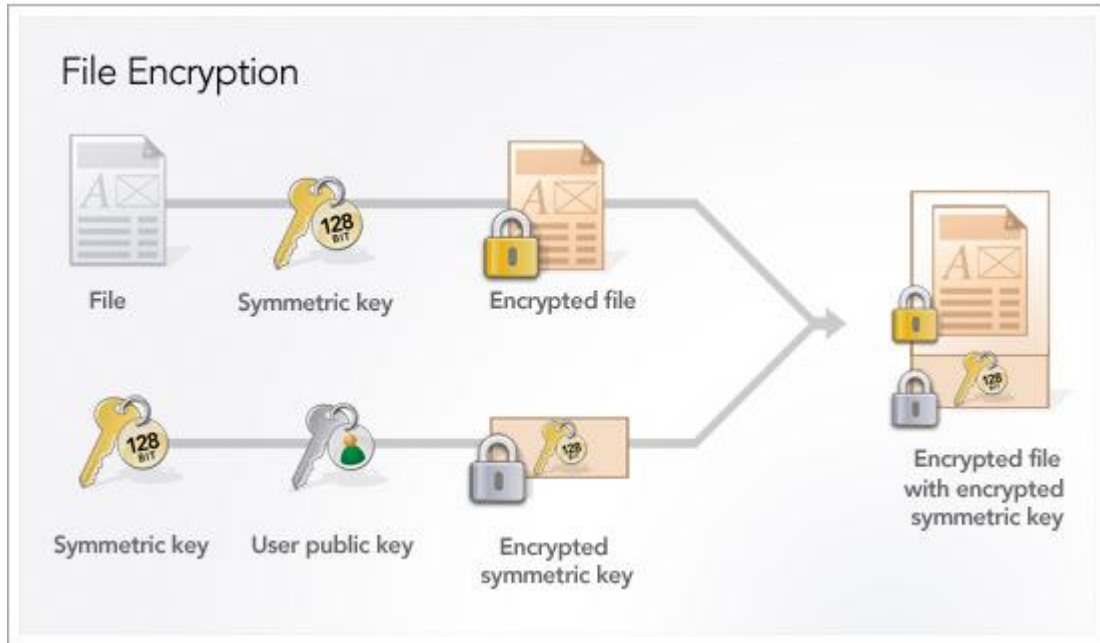


Fig 1: Symmetric-key Encryption

Symmetric-key algorithms are algorithms for cryptography that use the same cryptographic keys for both encryption of plaintext and decryption of ciphertext. The keys may be identical or there may be a simple transformation to go between the two keys. The keys, in practice, represent a shared secret between two or more parties that can be used to maintain a private information link. This requirement that both parties have access to the secret key is one of the main drawbacks of symmetric key encryption, in comparison to public-key encryption.

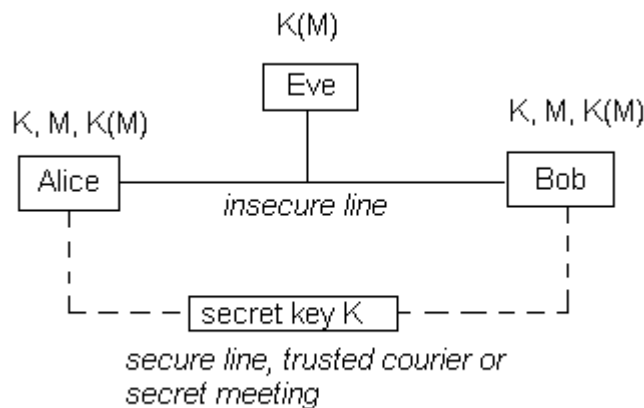


Fig 2: Public-key cryptography

Public-key cryptography, also known as asymmetric cryptography, is a class of cryptographic protocols based on algorithms that require two separate keys, one of which is secret (or private) and one of which is public. Although different, the two parts of this key pair are mathematically linked. The public key is used for encryption like, to encrypt

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

data or to validate a digital signature; at the same time we have to use the private key for the opposite operation, here to decryption is done on ciphertext or we have to create a digital signature.

1. It is concern with costs and complexities for number of shared data decryption keys which is to be shared.
2. The encryption and decryption key are dissimilar in public key encryption.
3. Regarding availability of files, there are a series of cryptographic plans which go as far as allowing a third-party auditor to verify the availability of files on behalf of the data owner without disclosing anything about the data or without compromising the data owner's anonymity.

II. RELATED WORK

Cloud computing is an internet base environment where users can store the data remotely in the cloud. Any cloud computing environment architecture can be divided basically into three layers, the characteristics layer, the models layer (infrastructure as a services, platform as a services, and software as a services), and the deployment layer [1].

These layers aim to (i) develop and adopt the rapidly evolving of cloud technology, (ii) abstract the details of inner implementations, and, (iii) facilitate the information retrieving service anywhere, anytime [2]. The following subsections explain the Cloud Data Encryption Based Quantum (CDEQ) model and the Cloud Encryption Model (CEM) in details as they are the most popular models used in the encryption process based clouds.

A. Cloud Data Encryption Based Quantum (CDEQ)

Cloud data encryption based quantum technology platform dispels all security fears through cloud data transmission [3], [4]. This technology offers: simple low-cost data protection, tools and security services integration, and an efficient disasters recovery. Quantum technology solves one of the key challenges in distributed computing. It can preserve data privacy when users interact with remote computing centers [6]. Its power came from the deployment of the Quantum Cryptography or Quantum Key Distribution (QKD) mechanisms, which are considered as the art of the encryption/ decryption process [7], [8], see fig.1. Through quantum channels, data is encoded based on prepared states known as photons. These photons are then sent as "keys" for encryption/ decryption secured messages [9]. The advantage of using such photons in data transmission lays in the no-cloning theorem (the quantum state of a single photon cannot be copied).

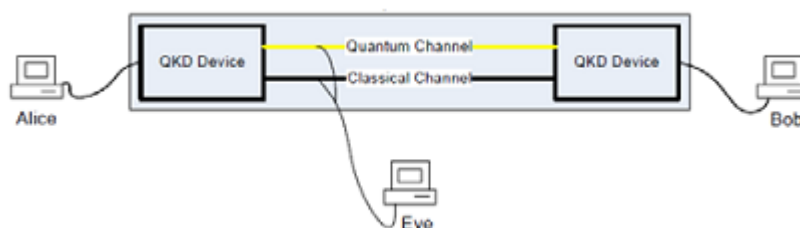


Fig. 3: Schematic of QKD

Bowfins looking for the perfect alliance between cloud computing and the quantum computing, which guarantees data protection for hosted files on remote computers or servers. He encrypted heavy duty of data by using the data processing servers as quantum computer, which succeeds in hiding input, processing and output data from malicious and attacks [10].

B. Cloud Encryption Models(CEM)

The two most important fields of information security in cloud environment are encryption and authentication. Generally, the encryption mechanism has become one of the basic priorities in maintaining the data security in the cloud, two popularity models based on data encryption technique are going to be explained briefly.

a) Cipher Cloud

Cipher Cloud provides a unified cloud encryption gateway with award-winning technology to encrypt sensitive data in real time before it's sent to the cloud. It also protects enterprise data by using operations-preserving

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

encryption and tokenization in both private and public cloud communication without affecting functionality, usability, or performance [10]. Cipher cloud provides ability to create a unified data protection policy across all clouds that users probably used to store data, such as Google, Amazon, Azure and others. [11]. One the cipher cloud advantages are the offering of multiple AES-compatible encryption and tokenization options, including format and function-preserving encryption algorithms. Users see the real data when accessing an application through the Cipher Cloud security gateway, whereas the data stored in a cloud application is encrypted [12].

By applying encryption in a cloud security gateway, Cipher Cloud eliminates the inherent security, privacy, and regulatory compliance risks of cloud computing [13].



Fig. 4: Cipher Cloud Model

Cipher Cloud's highly secured encryption preserves both the format and function of the data, so that cloud applications remain operational, but their real content remains locked within the enterprise [13]. After then, the process is reversed when employees access cloud applications through the appliance decrypting data in real time so that users see the actual data rather than the encrypted version that resides within the cloud.

b) Cryptographic Cloud Storage

Kamara and Lauter et al [14] proposed a virtual private storage services that would satisfy the standard demands (Confidentiality, integrity, Authentication .etc.). Most of the demands are done by encrypting the documents stored in the cloud. However, such encryption leads to hardness in both the search processes through documents and the collaboration process in real time editing.

Fig. 3 shows the architecture of the cryptographic storage service that is used in solving the security problems of "back-ups, archival, health record systems, secure data exchange and e-discovery" [14]. It contains three main components: Data Processor (DP) that processes data before sending it to the cloud, Data Verifier (DV) which verifies data's integrity and finally, Token Generator (TG) that generates tokens allowing the service provider to retrieve documents.

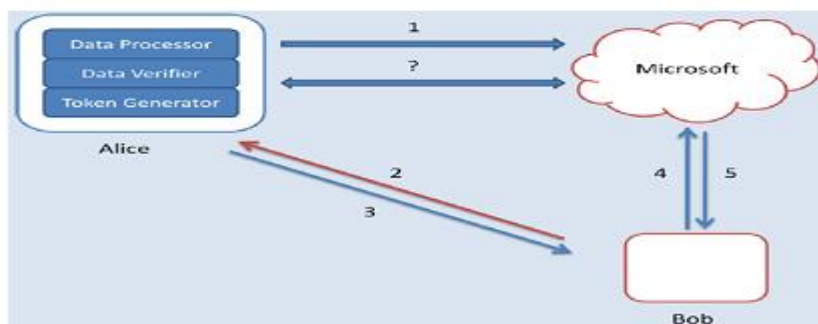


Fig. 5: Cryptographic Cloud Storage Architecture

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

Before uploading data to the cloud, Alice uses the data processor to encrypt and encode the documents along with their metadata, and then she sends them into the cloud. When she wants to download some documents, Alice & Bob uses the TG to generate a token and a decryption key.

The token is sent to the storage provider to select the encrypted files to be downloaded. After that, the DV is invoked to verify the integrity of the data using a master key. The document is decrypted using the decryption key [14].

III. THEORETICAL FOUNDATION

AGGREGATION OF SECRET KEYS:

Establishing a special type of public-key encryption which we call key-aggregate cryptography system (KACS). In KACS, users encrypt a message not only under a public key, but also under an identifier of ciphertext called class. The key owner holds a master-secret called master-secret key, which can be used to extract secret keys for different classes. More importantly, the extract key can be an aggregate key which is compact as a secret key for a single class, but aggregates the power of many such keys, i.e., the decryption power for any subset of ciphertext classes[15].

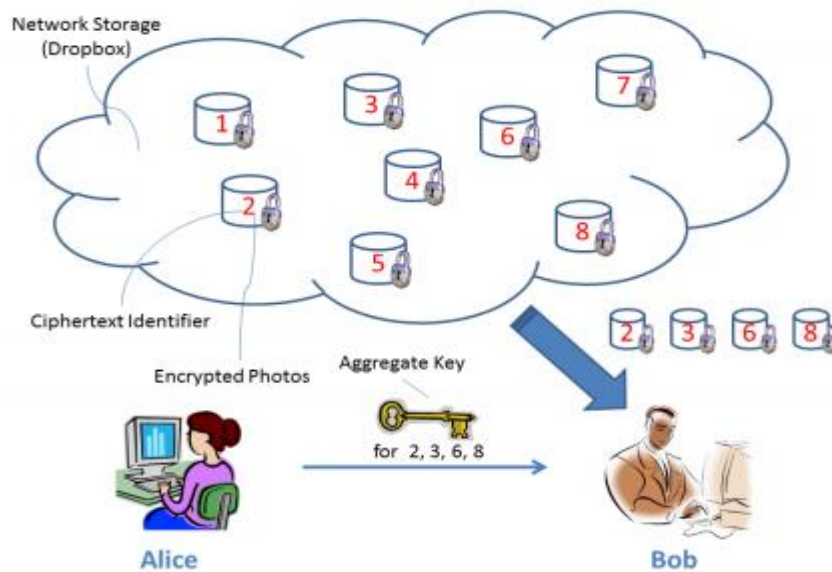


FIG 6: ALICE SHARE FEW FILES WITH BOB

Fig 6. Alice shares files with identifiers 2, 3, 6 and 8 with Bob by sending him a single aggregate key. Since our objective of our developing system is to design an efficient public-key encryption scheme which supports flexible delegation in the sense that any subset of the ciphertexts is decryptable by a constant-size decryption key (generated by the owner of the master-secret key).”

IV. SYSTEM ARCHITECTURE

Here we describe the main idea of data sharing in cloud storage using KACS, illustrated in following figure. Suppose Alice wants to share her data m_1, m_2, \dots, m_n on the server. She first performs Setup ($1^\lambda; n$) to get param and execute KeyGen to get the public/master-secret key pair (pk, msk).

The system parameter param and public-key pk can be made public and master-secret key msk should be kept secret by Alice. Anyone (including Alice herself) can then encrypt each m_i by $C_i = \text{Encrypt}(pk, I, m_i)$. The encrypted data are uploaded to the server.

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

With param and pk, people who cooperate with Alice can update Alice's data on the server. Once Alice is willing to share a set S of her data with a friend Bob, she can compute the aggregate key K_S for Bob by performing $\text{Extract}(\text{msk}, S)$. Since K_S is just a constant size key, it is easy to be sent to Bob via a secure e-mail.

After obtaining the aggregate key, Bob can download the data he is authorized to access. That is, for each $i \rightarrow S$, Bob downloads C_i (and some needed values in param) from the server. With the aggregate key K_S , Bob can decrypt each C_i by $\text{Decrypt}(K_S, S, I, C_i)$ for each $i \rightarrow S$.

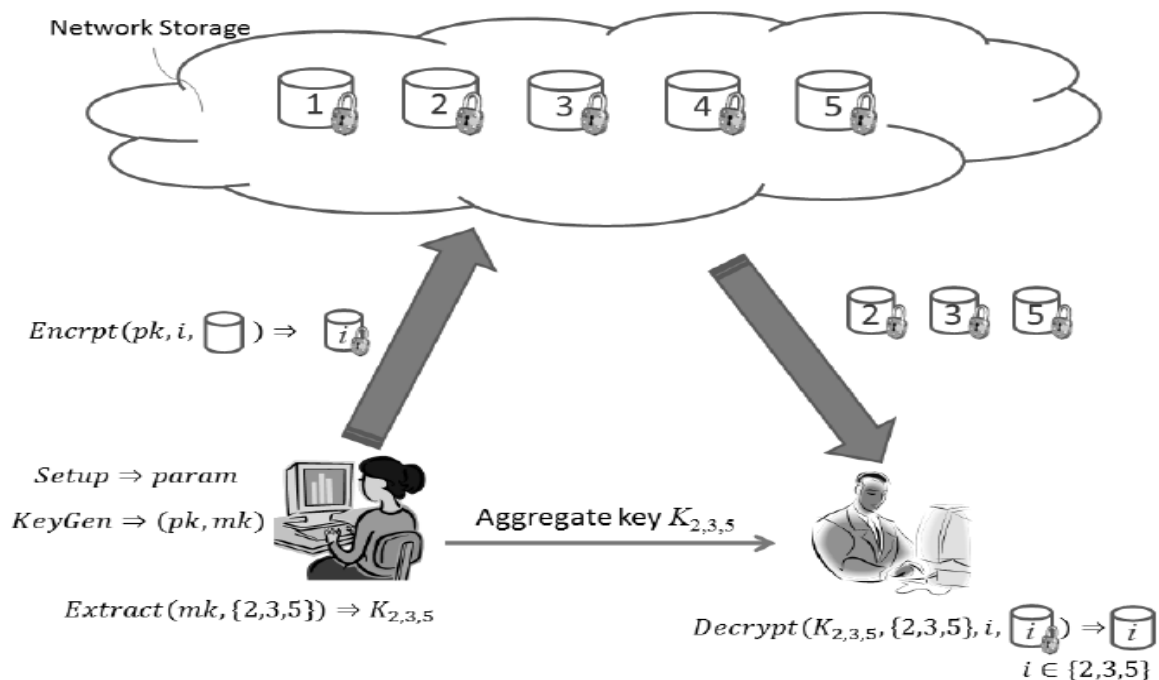


Fig 7. System Architecture

Since our proposed system contains following mechanism;

1. Setup ($1^\lambda, n$):

It is executed by the data owner to setup an account on an untrusted server. On input a security level parameter 1^λ and the number of ciphertext classes n , it outputs the public system parameter param, which is omitted from the input of the other algorithms for conciseness.

2. KeyGen:

As mentioned above, it is executed by the data owner to randomly generate a public/master-secret key pair (pk; msk).

3. Encrypt (pk, I, m):

It is executed by anyone who wants to encrypt data on cloud and share with others. On input a public-key pk, an index i denoting the ciphertext class, and a message m , it outputs a ciphertext C .

4. Extract (msk,S):

It is executed by the data owner for pass on the decrypting power for a certain set of ciphertext classes to a receiver. On input the master-secret key msk and a set S of indices corresponding to different classes, it outputs the aggregate key for set S denoted by K_S .



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

5. Decrypt (Ks,S, i,C):

It is executed by a receiver who received an aggregate key KS generated by Extract. On input KS , the set S , an index i denoting the ciphertext class the ciphertext C belongs to, and C , it outputs the decrypted result m if $i \in S$.

V. SYSTEM IMPLEMENTATION

a. Problem Statement:

Consider a scenario where two employees of a company would like to share some confidential business data using a public cloud storage service. For instance, Alice wants to upload a large collection of financial documents or photos to the cloud storage, which are meant for the directors of different departments to review. Suppose those documents/photos contain highly sensitive information that should only be accessed by authorized users, and Bob is one of the receiver and is thus authorized to view documents/photos related to his department. Due to concerns about potential data leakage in the cloud, Alice encrypts these documents/photos with different keys, and generates keyword ciphertexts based on department names, before uploading to the cloud storage.

Alice then uploads and shares those documents with the receiver using the sharing functionality of the cloud storage. In order for Bob to view the documents related to his department, Alice must delegate to Bob the rights both for keyword search over those documents, and for decryption of documents related to Bob's department. With a traditional approach, Alice must securely send all the searchable encryption keys to Bob.

We solve this problem by introducing a special type of public-key encryption which we call key-aggregate cryptography system (KACS). In KACS, users encrypt a message not only under a public-key, but also under an identifier of ciphertext called class. That means the ciphertexts are further categorized into different classes. The key owner holds a master-secret called master-secret key, which can be used to extract secret keys for different classes. More importantly, the extracted key have can be an aggregate key which is as compact as a secret key for a single class, but aggregates the power of many such keys, i.e., the decryption power for any subset of ciphertext classes.

b. KEY AGGREGATE FRAMEWORK:

The proposed system is basically design on the basis of key aggregation encryption. Here we are using two keys to encrypt and decrypt the data which are secret key and its aggregate key. The data owner creates the public system parameter and generates a secret key which is public key. Data can be encrypted by any user and he may decides ciphertext block associated with the plaintext file which want to be encrypted.

The data owner have rights to use the secret key from which he can generate an aggregate key which is use for decryption of a set of ciphertext blocks. The both keys can be sent to end user in very secure manner. The authenticated user having an aggregate key can decrypt any block of ciphertext. This project consists of five algorithms which are used to perform the above operations.

These algorithms implementations having following steps are as follow:

Step1. Setup and create the account on the server for sharing of data. This account is generated by data owner.

Step2. KeyGen algorithm is used for the generation of public key. The data owner generates a public secret key to encrypt the data over cloud. It also creates an aggregate key to access the block of ciphers of limited size.

Step3. Encrypts the data provided by the data owner by using the secret key. This encrypted data is then share among the cloud.

Step4. The aggregate key is used for extracting the particular block of the ciphers from the cipher file. But other encrypted data remains secure.

Step5. Decrypt: The encrypted data is then decrypted by using the same secret key which is use for encryption.

c. Aggregation of Secret Keys:

Introducing a special type of public-key encryption which we call key-aggregate cryptography system (KACS). In KACS, users encrypt a message not only under a public key, but also under an identifier of ciphertext called class. The key owner holds a master-secret called master-secret key, which can be used to extract secret keys for different classes.



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

More importantly, the extract key can be an aggregate key which is compact as a secret key for a single class, but aggregates the power of many such keys, i.e., the decryption power for any subset of ciphertext classes.

Framework:

Following are different framework activity perform for executing KACS encryption;

Step1:The data owner establishes the public system parameter via Setup.

Step2: It generates a public/ master-secret key pair via KeyGen.

Step3: Messages is encrypted via Encrypt .

Step4: Cipher text class is associated with the plaintext message which is to be encrypted.

Step5: The data owner uses the master-secret to generate an aggregate decryption key for a set of ciphertext classes via Extract.

Step6: The generated keys is passed to delegates securely through secure e-mails or secure channels.

Step7: Receiver with an aggregate key decrypts any ciphertext provided that the ciphertext's class is contained in the aggregate key.

VI. CONCLUSION AND FUTURE WORK

In this paper, we addressed an important issue of secure data sharing on untrusted storage. We investigated the challenges pertained to this problem and proposed data security in cloud using key aggregate cryptosystem.

In this paper, proposed system is found to be very efficient for sharing the data on cloud. For this we have used Key aggregate encryption algorithm which support delegation of secret keys for different ciphertext classes in cloud storage. It also produces constant-size ciphertexts such that efficient delegation of decryption rights for any set of ciphertexts which is here possible. Since in traditional methods unexpected privilege escalation will expose all data. And that we are able to avoid and provide more security by using key aggregate algorithm.

ACKNOWLEDGMENT

It is my great pleasure to express my deep sense of gratitude to my project guide Prof. M.C. Kshirsagar, for his valuable guidance, inspiration and wholehearted involvement during every stage of this paper presentation. I am very much grateful to the entire VBCOE for giving me all facilities and work environment which enable me to complete my task. I express my sincere thanks to Prof. S. G. Joshi Head of Computer Department, Prof. M.C. Kshirsagar (PG Coordinator), VB College of Engineering, Ahmednagar who gave me their valuable and rich guidance and help in presentation of this research paper.

REFERENCES

- [1] G. Clarke, Microsoft's Azure Cloud Suffers First Crash, The Register, March 16, 2009, http://www.theregister.co.uk/2009/03/16/azure_cloud_crash/
- [2] P. Ferrie, Attacks on Virtual Machine Emulators, White Paper, Symantec Corporation, January 2007, http://www.symantec.com/avcenter/reference/Virtual_Machine_Threats.pdf
- [3] G. Fowler, B. Worthen, The Internet Industry is on a Cloud – Whatever That May Mean, The Wall Street Journal, March 26, 2010
- [4] L. Youseff, M. Butrico, D. D. Silva, Toward a Unified Ontology of Cloud Computing, Grid Computing Environments Workshop, held with SC08, November 2008. <http://www.cs.ucsb.edu/~lyouseff/CCOntology/CloudOntology.pdf>
- [5] G. Ateniese, R. Di Pietro, L. V. Mancini, and G. Tsudik. Scalable and efficient provable data possession. In Proceedings of the 4th international conference on Security and privacy in communication networks (SecureComm '08), pages 1{10, New York, NY, USA, 2008. ACM.
- [6] Hodges, A. (2005), 'Can quantum computing solve classically unsolvable problems?'
- [7] H.K. Lo, H.F. Chau, Unconditional security of quantum key distribution over arbitrary long distances. Science 1999; 283(5410): 2050-2056.
- [8] L. Lydersen, R. Di Pietro, L. V. Mancini, C., Elser, D., Skaar, J. and Makarov, V. Hacking commercial quantum cryptography systems by tailored bright illumination. Nat. Photonics 4, 686, 2010.
- [9] K. Inoue, Quantum Key Distribution Technologies. IEEE Journal of Selected Topics in Quantum Electronics, vol. 12, no.4, July/August 2006.
- [10] J. Brodtkin. Gartner: Seven cloud-computing security risks. <http://www.infoworld.com/d/security-central/gartnerseven-cloudcomputing-security-risks-853>, 2008.
- [11] C.C.A : CipherCloud Gateway Architecture, www.ciphercloud.net.



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. 3, Issue 5, May 2015

- [12] M. v. Dijk and A. Juels. On the impossibility of cryptography alone for privacy-preserving cloud computing. In Hot topics in Security (HotSec'10), pages 1{8. USENIX Association, 2010.
- [13] Kamara and Lauter . CS2: A Searchable Cryptographic Cloud Storage System,IJSIR,2012.
- [14] G. Ateniese, S. Kamara, and J. Katz. Proofs of storage from homomorphic identification protocols. In Advances in Cryptology - ASIACRYPT '09, volume 5912 of Lecture Notes in Computer Science, pages 319{333. Springer, 2012}.
- [15] Cheng-Kang Chu, Sherman S. M. Chow, Wen-Guey Tzeng, Jianying Zhou “Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage” IEEE Transactions on Parallel and Distributed Systems. Volume: 25, Issue: 2, 2014.
- [16] S. S. M. Chow, Y. J. He, L. C. K. Hui, and S.-M. Yiu, “SPICE - Simple Privacy-Preserving Identity-Management for Cloud Environment,” in Applied Cryptography and Network Security – ACNS 2012, ser. LNCS, vol. 7341. Springer, 2012, pp. 526–543.
- [17] L. Hardesty, “Secure computers aren’t so secure,” MIT press, 2009, <http://www.physorg.com/news176107396.html>.
- [18] C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou, “Privacy-Preserving Public Auditing for Secure Cloud Storage,” IEEE Trans. Computers, vol. 62, no. 2, pp. 362–375, 2013.
- [19] B. Wang, S. S. M. Chow, M. Li, and H. Li, “Storing Shared Data on the Cloud via Security-Mediator,” in International Conference on Distributed Computing Systems - ICDCS 2013. IEEE, 2013.
- [20] S. S. M. Chow, C.-K. Chu, X. Huang, J. Zhou, and R. H. Deng, “Dynamic Secure Cloud Storage with Provenance,” in Cryptography and Security: LNCS, vol. 6805. Springer, 2012, pp. 442–464.
- [21] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, “Aggregate and Verifiably Encrypted Signatures from Bilinear Maps,” in Proceedings of Advances in Cryptology - EUROCRYPT '03, ser. LNCS, vol. 2656. Springer, 2003, pp. 416–432.