EFFICIENT SECURE DATA SHARING IN CLOUD COMPUTING WITH REVOCABLE STORAGE IDENTITY BASED ENCRYPTION METHOD

S.ARAFATH HASAN[1]

*[1]Departmentmet of Computer Science, Alagappa University,Karaikudi, Tamilnadu, India. Arafath.hasan@gmail.com[1]

Abstract— — —Cloud computing provides a flexible and convenient way for data sharing, which brings various benefits for both the society and individuals. But there exists a natural resistance for users to directly outsource the shared data to the cloud server since the data often contain valuable information. Thus, it is necessary to place cryptographically enhanced access control on the shared data. Identity-based encryption is a promising cryptographically primitive to build a practical data sharing system. However, access control is not static. That is, when some user's authorization is expired, there should be a mechanism that can remove him/her from the system. Consequently, the revoked user cannot access both the previously and subsequently shared data.. To overcome security problem the proposed system contain two levels of security and to reduce the unwanted storage space de-duplication[1,2] technique is involved. To increase the level of security one technique is a session password .Session passwords can be used only once and every time a new password is generated. To protect the confidentiality of sensitive data while supporting de-duplication[1,2]the convergent encryption technique has been proposed to encrypt the data before outsourcing, Symmetrickey algorithm uses same key for both encryption and decryption. In this paper,I will focus on session based authentication for both encryptions for files and duplication check for reduce space of storage on cloud.

Keywords – cloud Computing, security, privacy, Secret data, Deduplication

I INTRODUCTION

CLOUD computing is a paradigm that provides massive computation capacity and huge memory space at a low cost . It enables users to get intended services irrespective of time and location across multiple platforms (e.g., mobile devices, personal computers), and thus brings great convenience to cloud users. Among numerous services provided by cloud computing, cloud storage service, such as Apple's iCloud , Microsoft's Azure and Amazon's S3 , can offer a more flexible and easy way to share data over the Internet, which provides various benefits for our society . However, it also suffers from several security threats, which are the primary concerns of cloud users. Firstly, outsourcing data to cloud server implies that data is out control of users. This may cause users' hesitation since the outsourced data usually contain valuable and sensitive information. Secondly, data sharing is often implemented in an open and hostile environment, and cloud server would become a target of attacks. Even worse, cloud server itself may reveal users' data for illegal profit. Thirdly, data sharing is not static. That is, when a user's authorization gets expired, he/she should no longer possess the privilege of accessing the previously and subsequently shared data. Therefore, while outsourcing data to cloud server, users also want to control access to these data such that only those currently authorized users can share the outsourced data. Furthermore, to overcome the above security threats, such kind of identity-based access control placed on the shared data should meet the following security goals: • Data confidentiality: Unauthorized users should be prevented from accessing the plaintext of the shared data stored in the cloud server. In addition, the cloud server, which is supposed to be honest but curious, should also be deterred from knowing plaintext of the shared data. • Backward secrecy: Backward secrecy means that, when a user's authorization is expired, or a user's secret key is compromised, he/she should be prevented from accessing the plaintext of the subsequently shared data that are still encrypted under his/her identity. • Forward secrecy: Forward secrecy means that, when a user's authority is expired, or a user's secret key is compromised, he/she should be prevented from accessing the plaintext of the shared data that can be previously accessed by him/her.

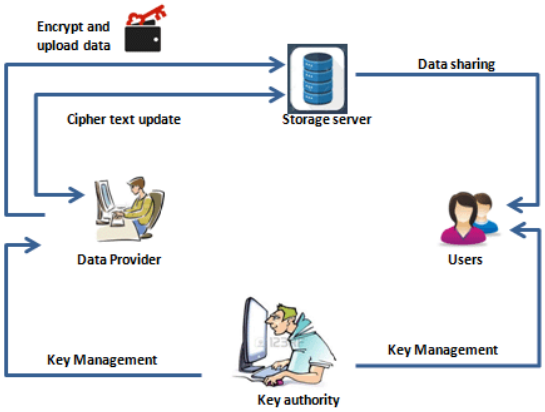


Fig: basic key security in cloud storage

A. Revocable identity-based encryption The concept of identity-based encryption was introduced by Shamir , and conveniently instantiated by Boneh and Franklin . IBE eliminates the need for providing a public key infrastructure (PKI). Regardless of the setting of IBE or PKI, there must be an approach to revoke users from the system when necessary, e.g., the authority of some user is expired or the secret key of some user is disclosed. In the traditional PKI setting, several techniques are widely approved, such as certificate revocation list or appending validity periods to certificates. However, there are only a few studies on revocation in the setting of IBE. Boneh and Franklin first proposed a natural revocation way for IBE. They appended the current time period to the Cipher Text, and non-revoked users periodically received private keys for each time period from the key authority. Unfortunately, such a solution is not scalable, since it requires the key authority to perform linear work in the number of non-revoked users. In addition, a secure channel is essential for the key authority and non-revoked users to transmit new keys. Boldyreva, Goyal and Kumar introduced a novel approach to achieve efficient revocation. They used a binary tree to manage identity such that their RIBE scheme reduces the complexity of key revocation to logarithmic (instead of linear) in the maximum number of system users. However, this scheme only achieves selective security. Subsequently, by using the aforementioned revocation technique, Libert and Vergnaud proposed an adaptively secure RIBE scheme based on a variant of Water's IBE scheme, Chen et al. Constructed a RIBE scheme from lattices.

Seo and Emura proposed an efficient RIBE scheme resistant to a realistic threat called decryption key exposure, which means that the disclosure of decryption key for current time period has no effect on the security of decryption keys for other time periods. Inspired by the above work and, Liang et al. introduced a cloud-based revocable identity-based proxy re-encryption that supports user revocation and Cipher Text update. To reduce the complexity of revocation, they utilized a broadcast encryption scheme to encrypt the Cipher Text of the update key, which is independent of users, such that only non-revoked users can decrypt the update key. B. Forward-secure cryptosystems In 1997, Anderson introduced the notion of forward security in the setting of signature to limit the damage of key exposure. The core idea is dividing the whole lifetime of a private key into T discrete time periods, such that the compromise of the private key for current time period cannot enable an adversary to produce valid signatures for previous time periods. Subsequently, Bellare and Miner provided formal definitions of forward-secure signature and presented practical solutions. Since then, a large number of forward-secure signature schemes has been proposed. Canetti, Halevi and Katz proposed the first forward-secure public-key encryption scheme. Specifically, they firstly constructed a binary tree encryption, and then transformed it into a forward-secure encryption with provable security in the random oracle model. Based on Canetti et al's approach, Yao et al. proposed a forward-secure hierarchical IBE by employing two hierarchical IBE schemes, and Nieto et al. designed a forward-secure hierarchical predicate encryption.

II RELATED WORK

The objective of this literature review is to study the work carried and published by different researchers and authors in the domain of Document Annotation and tagging. Kun He, Jing Chen, Ruiying Du, Qianhong Wu, Guoliang Xue, and Xiang Zhang proposed in "DeyPoS: Deduplicatable Dynamic Proof of Storage for Multi-User Environments" the comprehensive requirements in multi-user cloud storage systems and introduced the model of deduplicatable dynamic PoS. They designed a novel tool called HAT which is an efficient authenticated structure. Based on HAT, proposed the first practical deduplicatable dynamic PoS scheme called DeyPoS and proved its security in the random oracle model.The theoretical and experimental results show that our DeyPoS implementationis efficient, especially when the file size and the number of the challenged blocks are large.[1] Jianghong Wei, Wenfen Liu, Xuexian Hu proposed in paper "Secure Data Sharing in Cloud Computing Using Revocable-Storage Identity-Based Encryption" proves cloud computing brings great convenience for people. Particularly, it perfectly matches the increased need of sharing data over the Internet. In this paper, to build a cost-effective and secure data sharing system in cloud computing, they proposed a notion called RS-IBE, which supports identity revocation and ciphertext update simultaneously such that a revoked user is prevented from accessing previously shared data, as well as subsequently shared data. Furthermore, a concrete construction of RS-IBE is presented. [2] Pietro and Sorniotti proposed in paper

"Boosting Efficiency and Security in Proof of Ownership for Deduplication" proves another proof of ownership scheme which improves the efficiency. Xu et al.[4] proposed a client-side deduplication scheme for encrypted data, but the scheme employs a deterministic proof algorithm which indicates that every file has a deterministic short proof. Thus, anyone who obtains this proof can pass the verification without possessing the file locally. Other deduplication schemes for encrypted data were proposed for enhancing the security and efficiency. Note that, all existing techniques for cross-user deduplication on the client-side were designed for static files. Once the files are updated, the cloud server must regenerate the complete authenticated structures for these files, which causes heavy computation cost on the server-side. [3] The concept of proof of storage was introduced by Ateniese et al. in paper "Provable data possession at untrusted stores", and Juels and Kaliski, respectively. The main idea of PoS is to randomly choose a few data blocks as the challenge. Then, the cloud server returns the challenged data blocks and their tags as the response. Since the data blocks and the tags can be combined via homomorphic functions, the communication costs are reduced. The subsequent works extended the research of PoS, but those works did not take dynamic operations into account. Erway et al. and later works focused on the dynamic data. Among them, the scheme in is the most efficient solution in practice. However, the scheme is stateful, which requires users to maintain some state information of their own files locally. Hence, it is not appropriate for a multiuser environment. Halevi et al. introduced the concept of proof of ownership which is a solution of cross-user deduplication on the client-side. It requires that the user can generate the Merkle tree without the help from the cloud server, which is a big challenge in dynamic PoS. [5] Zheng and Xu proposed in paper "Secure and efficient proof of storage with deduplication" proves a solution called proof of storage with deduplication, which is the first attempt to design a PoS scheme with deduplication. Du et al. Introduced proofs of ownership and retrievability, which are like but more efficient in terms of computation cost. Note that neither can support dynamic operations. Due to the problem of structure diversity and private tag generation, cannot be extended to dynamic PoS. Wang et al. and Yuan and Yu considered proof of storage for multi-user updates, but those schemes focus on the problem of sharing files in a group. Deduplication in these scenarios is to deduplicate files among different groups. Unfortunately, these schemes cannot support deduplication due to structure diversity and private tag generation. In this paper, they consider a more general situation that every user has its own files separately. [6] Jingwei Li, Jin Li, Dongqing Xie, and Zhang Cai "Secure Auditing and Deduplicating Data in cloud" proves both data integrity and deduplication in cloud, they propose SecCloud and SecCloud+. SecCloud introduces an auditing entity with maintenance of a MapReduce cloud, which helps clients generate data tags before uploading as well as audit the integrity of data having been stored in cloud. In addition, SecCloud enables secure deduplication through introducing a PoS protocol and preventing the leakage of side channel information in data deduplication. Compared with previous work, the computation by user in SecCloud is greatly reduced during the file uploading and auditing phases. SecCloud+ is an advanced construction motivated by the fact that customers always want to encrypt their data before uploading, and allows for integrity auditing and secure deduplication directly on encrypted data. Compared with previous work, the computation by user in SecCloud is greatly reduced during the file uploading and auditing phases. SecCloud+ is designed motivated by the fact that customers always want to encrypt their data before uploading, and enables integrity auditing and secure deduplication on encrypted data. [7]

III PROBLEM FORMULATION

The specific problem addressed in this paper is how to construct a fundamental identity-based cryptographically tool to achieve the above security goals. We also note that there exist other security issues that are equally important for a practical system of data sharing, such as the authenticity and availability of the shared data.

IV PROPOSED WORK IMPLEMENTATION

It seems that the concept of revocable identity-based encryption (RIBE) might be a promising approach that fulfils the aforementioned security requirements for data sharing. RIBE features a mechanism that enables a sender to append the current time period to the Cipher Text such that the receiver can decrypt the Cipher Text only under the condition that he/she is not revoked at that time period. As indicated in Figure 1, a RIBE-based data sharing system works as follows: Step 1: The data provider (e.g., David) first decides the users (e.g., Alice and Bob) who can share the data. Then, David encrypts the data under the identities Alice and Bob, and uploads the Cipher Text of the shared data to the cloud server. Step 2: When either Alice or Bob wants to get the shared data, she or he can download and decrypt the corresponding Cipher Text. However, for an unauthorized user and the cloud server, the plaintext of the shared data is not available. Step 3: In some cases, e.g., Alice's authorization gets expired, David can download the Cipher Text of the shared data, and then decrypt-then-re-encrypt the shared data such that Alice is prevented from accessing the plaintext of the shared data, and then upload the re-encrypted data to the

cloud server again. Obviously, such a data sharing system can provide confidentiality and backward secrecy. Furthermore, the method of decrypting and re-encrypting all the shared data can ensure forward secrecy. However, this brings new challenges. Note that he process of decrypt-then-re-encrypt necessarily involves users' secret key information, which makes the overall data sharing system vulnerable to new attacks. In general, the use of secret key should be limited to only usual decryption, and it is inadvisable to update the cipher text periodically by using secret key. Another challenge comes from efficiency. To update the Cipher Text of the shared data, the data provider has to frequently carry out the procedure of download-decryptencrypt- upload. This process brings great communication and computation cost, and thus is cumbersome and undesirable for cloud users with low capacity of computation and storage. One method to avoid this problem is to require the cloud server to directly re-encrypt the Cipher Text of the shared data. However, this may introduce cipher text extension; namely, the size of the Cipher Text of the shared data is linear in the number of times the shared data have been updated. In addition, the technique of proxy re-encryption can also be used to conquer the aforementioned problem of efficiency. Unfortunately, it also requires users to interact with the cloud server in order to update the Cipher Text of the shared data.

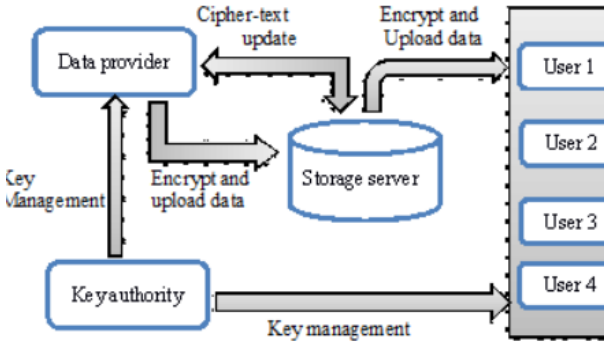


Fig: proposed model framework

EXISTING SYSTEM

Natural revocation way for IBE is proposed in this non-revoked users periodically received private keys for each time period from the key authority. Unfortunately, such a solution is not scalable, since it requires the key authority to perform linear work in the number of non-revoked users. In addition, a secure channel is essential for the key authority and non-revoked users to transmit new keys.

DISADVANTAGES

- It's not scalable.
- It's not secure.

PROPOSED SYSTEM

We introduce a notion called revocable storage identity-based encryption (RS-IBE) for building a cost-effective data sharing system that fulfills the three security goals. More precisely, the following achievements are captured in this paper:

• We provide formal definitions for RS-IBE and its corresponding security model;

• We present a concrete construction of RS-IBE. The proposed scheme can provide confidentiality and backward/forward2 secrecy simultaneously;

• We prove the security of the proposed scheme in the standard model, under the decisional ℓ-Bilinear Diffie-Hellman Exponent (ℓ-BDHE) assumption. In addition, the proposed scheme can withstand decryption key exposure;

ADVANTAGES

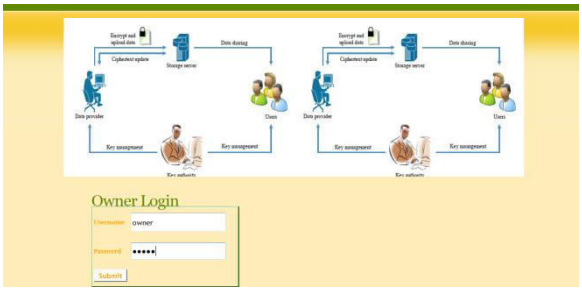
- The procedure of ciphertext update only needs public information.
- The additional computation and storage complexity, which are brought in by the forward secrecy.

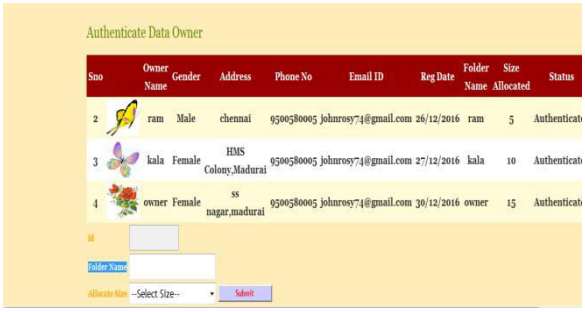
MODULES

- Data Provider
- Users
- Storage Server
- Key Authority

VI. EXPERIMENTAL RESULTS

VI. EXPERIMENTAL RESULTS





V CONCLUSION

Cloud computing brings great convenience for people. Particularly, it perfectly matches the increased need of sharing data over the Internet. In this paper, to build a cost-effective and secure data sharing system in cloud computing, we proposed a notion called RS-IBE, which supports identity revocation and Cipher Text update simultaneously such that a revoked user is prevented from accessing previously shared data, as well as subsequently shared data. Furthermore, a concrete construction of RS-IBE is presented. The proposed RS-IBE scheme is proved adaptive-secure in the standard model, under the decisional ℓ-DBHE assumption. The comparison results demonstrate that our scheme has advantages in terms of efficiency and functionality, and thus is more feasible for practical applications.

REFERENCES

[1]. L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, "A break in the clouds: towards a cloud definition," ACM SIGCOMM Computer Communication Review, vol. 39, no. 1, pp. 50–55, 2008.

[2]. iCloud. (2014) Apple storage service. [Online]. Available: https://www.icloud.com/

[3]. Azure. (2014) Azure storage service. [Online]. Available: http://www.windowsazure.com/

[4]. Amazon. (2014) Amazon simple storage service (amazons3). [Online]. Available:http://aws.amazon.com/s3/

[5]. K. Chard, K. Bubendorfer, S. Caton, and O. F. Rana, "Social cloud computing: A vision for socially motivated resource sharing," Services Computing, IEEE Transactions on, vol. 5, no. 4, pp. 551–563, 2012.

[6]. C. Wang, S. S. Chow, Q. Wang, K. Ren, and W. Lou, "Privacypreserving public auditing for secure cloud storage," Computers, IEEE Transactions on, vol. 62, no. 2, pp. 362–375, 2013.

[7]. G. Anthes, "Security in the cloud," Communications of the ACM, vol. 53, no. 11, pp. 16–18, 2010.

[8]. K. Yang and X. Jia, "An efficient and secure dynamic auditing protocol for data storage in cloud computing," Parallel and Distributed Systems, IEEE Transactions on, vol. 24, no. 9, pp. 1717–1726, 2013.

[9]. B. Wang, B. Li, and H. Li, "Public auditing for shared data with efficient user revocation in the cloud," in INFOCOM, 2013 Proceedings IEEE. IEEE, 2013, pp. 2904–2912.

[10]. S. Ruj, M. Stojmenovic, and A. Nayak, "Decentralized access control with anonymous authentication of data stored in clouds," Parallel and Distributed Systems, IEEE Transactions on, vol. 25, no. 2, pp. 384–394, 2014.

S.ARAFATH HASAN, MCA., D.O.B: 31.03.1983, place of birth from Ilayangudi, Sivagangai District. Details of qualifications M.C.A from Jamal Mohamed College form Trichy2008. B.Sc.,(Computer Science) Dr.Zakir Husain College from Ilayangudi, 2005. He has working as Assistant Professor. 4 years Experience from Dr.Zakir Husain College from Ilayangudi, sivagangai District. Then he worked as a Software Developer in Structural Engineering Research Centre (Government of India), Council of Scientific and Industrial Research (CSIR), Taramani, Chennai, India, from June 2008 to Aug 2010. His areas of project Developed the Software for useful in modeling earthquake occurrences for Seismic Hazard Analysis.