ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 2 Issue 8, August 2013

Secure Re-encryption in Unreliable Cloud using Synchronous clock

Shyam Sarania, Amit Gupta

Abstract— **In this Paper With rapid development of cloud computing, more and more enterprises will outsource their sensitive data for sharing in a cloud. To keep the shared data confidential against untrusted cloud service provider(CSPs), a nature way is to store only the encrypted data, and revoking the access rights from users when they are no longer authorized to access the encryption scheme is proposed which is based on the internal server of cloud server.**

Index Terms—**Attribute-based encryption, cloud computing, proxy re-encryption.**

I. Introduction

With the technological advancement today, many businesses have the capability to access to almost any type of information from any place or location. Today it is most of the companies are potential enough to store data and maintain them with the click of a mouse. Now the question arises, are you still using the traditional method of storing and maintaining data, don't you want to complete with your rivals and would you like to have your business and coustmor detail in online so that you can access them any time when you need them. This thing can be made possible with the help of cloud computing services and the providers.

The use of cloud computing is increasingly popular due to the potential cost saving from outsourcing data to the cloud service provider (CSP). One technique to protect the data from a possible entrusted CSP is for the data owner to encrypt the out sourced data[1],[2]. Flexible encryption schemes such as attribute based encryption (ABE) [3]-[4] can be adopted to provide fine gain access control.

ABE allows data to be encrypted using an access structure comprised of different attributes. Instead of specific decryption keys for specific files, users are issued attribute keys. User must have the necessary attributes that satisfy the access structure in order to decrypt the file For example, a file encrypted using the access structure {(α1 and α2) or α3} means that either a user with attribute α1 and α2 or a user with attribute α3, can decrypt the file.

The key problem of storing encrypted data in the cloud lies in revoking access rights from users, A user whose permission is revoked will still retain the keys issued earlier, and thus can still decrypt data in the cloud. A native solution is to let the data owner immediately re-encryption the data, so that the revoked user cannot decrypt the data using their old keys, while distributing the new keys to the remaining authorized users. This solution will lead to a performance bottleneck, especially when there are frequent user revocations.

An alternative solution is to apply the proxy re-encryption (PRE) technique. This approach takes advantage of the abundant resources in a cloud by delegating the cloud to re-encrypt data [5], This approach is also called command-driven re-encryption scheme, where cloud servers execute re-encryption while receiving commands from the data owner.

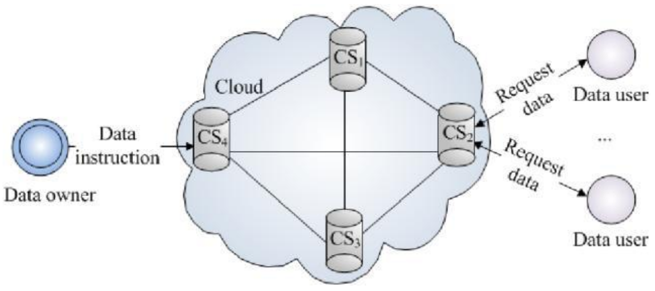


Fig 1. A typical cloud environment

Which should be propagated to CS1, CS2, and CS3? Due to a network outage, CS2 did not receive the command, and did not re-encrypt the data. At this time, if revoked users query CS2, they can obtain the old cipher text, and can decrypt it using their old keys.

A better solution is to allow each cloud server to independently re-encrypt data without receiving any command from the data owner. In this paper, we propose a secure re-encryption scheme in untrusted cloud. It is a time based re-encryption scheme, which allows each cloud server to automatically re-encryption data based on its internal clock. The basic idea of secure re-encryption scheme in untrusted cloud scheme is to associate the data with an access control and an access time. Each user is issued keys associated with attributes and attribute effective times. The data can be decrypted by the users using the keys with attribute effective times satisfying the access time. Unlike the command-driven re-encryption scheme, the data owner and the CSP share a secret key, with which each cloud server can re-encrypt data by updating the data access time according to it's the cloud environment. A cloud is essentially a large scale distributed system where a data owner's data is replicated over multiple servers for high availability. As a distributed system, the cloud will experience failures common to such as server crashes and network outages. As result re-encryption commands sent by the data owner may not propagate toall the timely fashion.

II. Procedure for Paper Submission

A. *Review Stage*

Many researchers have proposed storing encrypted data in the cloud to define against the CSP [1], [2]. Under this approach, users are revoked by having a third party to re-encrypt data such that previous keys can no longer decrypt any data [5]. The solution by [5] for instance, lets the data owner issues a re-encryption key to an untrusted server to re-encrypt the data. Their solution utilizes PRE [5], which allows the server to re-encrypt the stored cipher text that can only be decrypted using a different key. During the process, the server does not learn the content of the cipher text or the decryption keys.

ABE is a new cryptographic technique that efficiently supports fine gained access control. The combination of PRE and ABE was first introduce by [5], and extended by [5]. In [5], a hierarchical attribute-based encryption (HABE) scheme is proposed to achieve high performance and full delegation. The main difference between prior work and ours is that we do not require the underlying cloud infrastructure to be reliable in order to ensure correctness.

Our scheme relies on time to re-encrypt data. However, in a cloud, the internal clock of each cloud server may differ. There have been several solutions to this problem. For instance, [5] proposed a probabilistic synchronization scheme, which exchanges message delay to estimate the maximal difference between two communicating nodes to synchronize the clocks. Work by [5] proposed a clock synchronization scheme for cloud environments, which uses an authoritative time source shared by all participants in a truncation to achieve clock synchronization between virtual cloud policy enforcement points. By applying these techniques to determine the maximal time difference between the data owner and each cloud server, our secure re-encryption scheme in untrusted cloud scheme can always achieve correct access control in untrusted cloud.

III. Preliminary

A. *System model*

The software architecture of the proposed system is shown below

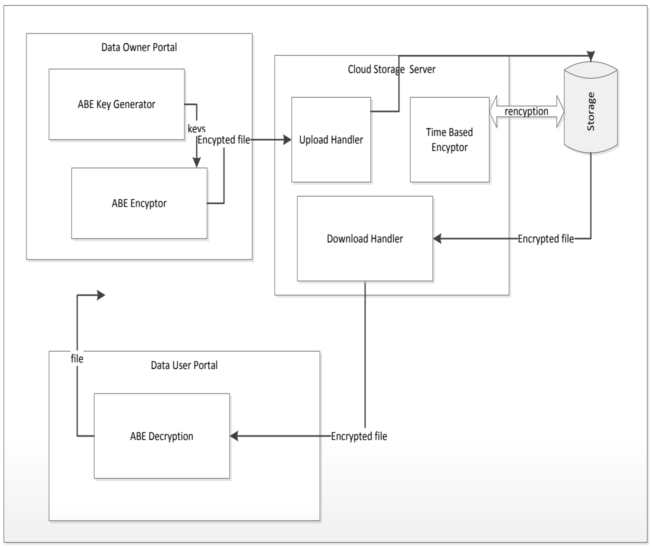


Fig 2. Software architecture

The software consist of three subsystems

Data owner portal: Data owner portal is the interface for the file uploaders. It generates ABE keys for encrypting the file , encrypts the file and store in cloud storage.

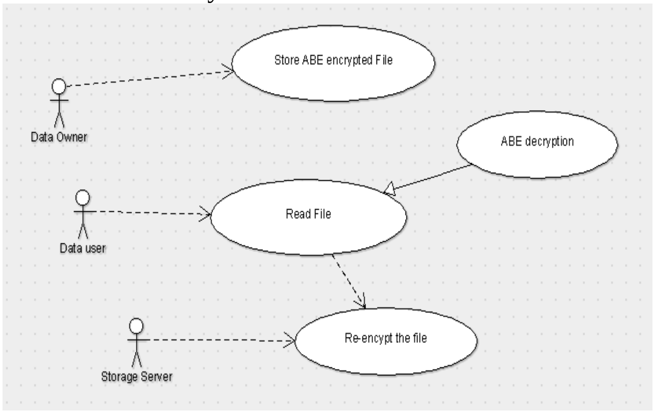
Data user portal : Data user portal is the interface for the file downloader. It generates the ABE keys for decrypting the file, decrypts the file got from the cloud server.

Cloud Storage Server: encrypted contents along with keys are stored in the cloud server. Every time read access arrives based on the time, it re-encrypts the file and store in the cloud storage.

B. *Design model*

The main design goal is to achive scalable user revocation while protecting data security in cloud computing. Specifically, we categorize our goal into the following points:

- **Fine Gained Access Control.** The data owner can specify expressive access structure for each data.
- **Data Consistency.** This requires that all data users who request file F, should obtain the same content in the same time slice.
- **Data Confidentiality.** The CSP and malicious users cannot recover data without the data owner's permission.
- **Cost Efficiency**. The re-encryption cost on the CSP is relatively low.



C. *Adversary Model*

There are two kinds of adversaries in the system: CSP, and malicious users. The CSP will correctly, execute the protocol defined previously, but may try to gain some additional information about the stored data. The malicious user wants to access the data, to which he is not eligible to access. The communication channels are assumed to be secured under existing security protocols such as SSL to protect data security during information transferring. Note that both an CSP, and malicious users, can exist together. We assume that the CSP will not collude may collude to obtain additional information. This assumption is reasonable, and has also been made known in previous research, e.g., the proxy re-encryption system [4], where the semi-trusted proxy server is assumed to not between the CSP and the data owner.

IV. BASIC SECURE RE-ENCRYPTION

In the basic scheme of the algorithm, we consider ideal conditions, where the data owner and all of the cloud servers in the cloud share a synchronized clock, and there are no transmission and queuing delays when executing read and write commands.

A. *Intuition*

The data owner will first generate a shared secret key to the CSP. Then, after the data owner encrypts each file with the appropriate attribute structure and time slice, the data owner uploads the file in the cloud. The CSP will replicate the file to various cloud server stores an encrypted file F with A and TSi. When a user queries that cloud server, the cloud server first uses its own clock to determine the current time slice. Assuming that the current slices is TSi+K, the cloud server will automatically re-encrypted F with TSi+K without receiving any command from the data owner. During the process, the cloud server cannot gain the contents of the cipertext and the decryption keys. Only users with keys satisfying A and TSi+K decrypt F.

B. *Protocol Description*

We devide the description of the basic scheme algorithm into three components: data owner initialization, data user read data and data owner write data. We will rely on the following functions.

1) *Setup* $\rightarrow$ *(Public Key, Master Key, s) :* At TS0, the data owner publishes the system public key, keeps the system master key secret. And send the shared secret key s to the cloud.

2) *Key generation (Public Key, s Public key of user, Attribute, Time)* $\rightarrow$ *(Secrete Key of user, {Secrete Key period of user, Attribute}):* When the data owner wants to grant data user attributes with valid time period, the data owner generated Public key of user and {Secrete Key period of user, Attribute} using the system public key, the system master key, the shared secret key, user's public key, user's attributes and eligible time.

3) Encryption (public key, s, TSt, File) $\rightarrow$ *(cipher text)* At Tst, the data owner encrypts file with access structure attributes, and produces cipher text using the system public key, time slice, and plain text file.

4) Decryption(Public Key, Cipher text, Secret key of user {secret key time of user, aij} 1<=j<=n) $\rightarrow$ *file* : At TSt, user U, who possesses version t attribute secrect keys on all attributes in CCi, recovers file using the system public key, the user identity secret key, and the user attribute secret keys.

5) *Re encryption(cipher text, s, TSt+k)* $\rightarrow$ *Ct+k A :* When the cloud server wants to return a data user with the file at TSt+k, it updates the cipher text from CtA to Ct+K A using the shared secret key.

V. SECURITY ANALYSIS

The Encrypt algorithm in the Time scheme is the same as the Encryption algorithm in HABE, which has been proven to be semantically secure. Therefore, we consider that the time scheme is secure if the following propositions hold:

Proposition 1. The keys produced by the generate key algorithm are secure.

Proposition 2. The cipher text produced by the Re encryption aloritm is semantically secure.

Proposition 3. Given the root secret key and the original cipher text, the CSP can know neither the underlying data, nor UAKs while executing re-encryption. For Proposition 1, we prove that the Generate Key algorithm is as the key generation algorithm in HABE. First, the way to Generation a UIK is same in both algorithms. Then, Given the system public key, the system master key MK, user public key, and attribute, if the data owner takes the time-based attribute public key a as input of the Key Generation algorithm in HABE, then the produced UAK is the same as that of the key generation algorithm that takes time, the initial attribute public key, and root secret key s as inputs. As proven due to the BDH assumption, the malicious users cannot obtain master key, even if all of them collude. Therefore, Proposition 1 is correct. For Proposition 2, we prove that Re-Encryption algorithm is as secure as the Encryption algorithm in HABE. Given system public key and data file with access structure attribute, if the data owner takes the time-based attribute public keys fPK(y) a ga∈A, fPK(y;m) a ga∈A, fPK(y;m;d) a ga∈A, and a random number r′ = r + r′ as inputs of the Encryption algorithm in HABE, then the produced ciphertext is the same as that of the re-encryption algorithm that takes time t=(y; m; d), the original cipher text, and root secret key s as a inputs. Therefore, Proposition 2 is correct For Proposition 3, we first prove that CSP cannot derive the system master key and UAKs from the root secret key s. As compared to HABE, the TimePRE scheme discloses an additional root secret key to the CSP, which is randomly chosen by the data owner and has nothing to do with the system master key. Therefore, the CSP cannot derive the system master key from the root secret key. Based on Proposition 1, the CSP cannot obtain UAKs without the system master key. Then, We prove that the CSP cannot compromise data security given the original cipher text. Note that the original cipher text is encrypted with encryption algorithm, which is semantically secure. Therefore, the cipher text can be decrypted by only the entity who possesses UAKs on the initial attribute public keys. In the TimePRE scheme, a users UAKs are generated on the time-based attribute public key, rather than the initial attribute public key. Therefore only the data owner with the initial attribute secret keys can recover the data from the original cipher text. Neither the users, nor the CSP can decrypt the original ciphertext.

VI. CONCLUSION

In this paper, We proposed the time based scheme to achieve fine gain access control and scalable user revocation in a cloud environment. Our scheme enables each user's access right to be effective in a pre-determined period of time, and enable the CSP to re-encrypt cipher texts automatically, based on its own time. Thus, the data owner can be offline in the process of user revocations. The main problem with our scheme is that it requires the effective time periods to be the same for all attributes associated with a user. Although we provide a possible improvement, the users will be issued more UAKs. Our future work is to allow different effective time periods for different attributes associated with a user, without increasing the number of UAKs associated with each user.

VII. ACKNOLADGMENT

I would like to acknowledge the contribution of certain distinguished people, without their support and guidance this research work would not have been completed.
I take this opportunity to express my sincere thanks and deep sense of gratitude to my project guide ***Prof. Amit Gupta*,** for his guidance and moral support during the course of preparation of this project report. I really thank him from the rock bottom of my heart for always being there with his extreme knowledge and kind nature.
I take this opportunity to thank all my friends and colleagues who started me out on the topic and provided extremely useful review feedback and for their all time support and help in each and every aspect of the course of my project preparation. I am grateful to my college **Sri Balaji College Of Engineering And Technology** for providing me all required stuff and good working environment.
I feel a deep sense of gratitude of my **parents** who formed part of my vision and taught me the good things that really matter in life. I am grateful to my loving **wife** for her excellent support. I would not have been able to finish my work without their help and love.
Finally I would like to express my thanks and gratitude to everyone who has directly or indirectly helped me in order to complete this work.

REFERENCES

[1] S. Kamara and K. Lauter, "Cryptographic cloud storage," *Financial Cryptography and Data Security*, 2010.
[2] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, and I. Stoica, "A view of cloud computing," *Communications of the ACM*, 2010.
[3] A. Sahai and B. Waters, "Fuzzy identity-based encryption," *Advances inCryptology–EUROCRYPT*, 2005.
[4] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. Of ACM ,2006.*
[5] Qin Liu†‡, Chiu C. Tan‡, Jie Wu‡, and Guojun Wang†"Reliable Re-encryption in Unr Unreliable Clouds" in IEEE Globecom 2011 proceedings.

First Author This is shyam sarania form vadodara Gujarat. And I studying in M.tech in computer Science in sri balaji collage of engineering , jaipur in RTU.

Second Author : This is Amit gupta . He is a assistant professor of sri balaji collage of engineering, jaipur.