IJDCST @December Issue- V-1, I-8, SW-9
ISSN-2320-7884 (Online)
ISSN-2321-0257 (Print)

Conditional Proxy Re-Encryption in Secure Erasure Code-Based Cloud Storage System

[1]G.Nagababu , [2]R. Ramesh
[1] MTech Student, K.K.R & K.S.R Institute of Technology and Sciences, Vinjanampadu, Guntur(D.T).
[2] Associate Professor, K.K.R & K.S.R Institute of Technology and Sciences, Vinjanampadu, Guntur(D.T).

Abstract: A cloud storage systemprovides long-term storage services over the Internet and consisting of a collection of storage servers.Storing data in a third party's cloud system causes serious concern over data confidentiality. Data confidentiality can be provided in the general encryption techniques; however, there is limited no. of functionalities. Constructing a secure storage system that supports multiple functions is challenging when the storage system is distributed and has no central authority. Previously, a threshold proxy re-encryption scheme and integrate it with a decentralized erasure code such that a secure distributed storage system is formulated. The main technical contribution is that the proxy re-encryption scheme supports encoding operations over encrypted messages as well as forwarding operations over encoded and encrypted messages. However, we can fully integrate the encryption, encoding, and forwarding. However, the crackers may attack the cipher text. Hence, we propose the conditional proxy for the re-encryption of the cipher text, whereby only ciphertext satisfying one condition set by sender can be transformed by the proxy and then decrypted by Receiver. We formalize its security model and propose an efficient C-PRE scheme, whose chosen-ciphertext security is proven under the 3-quotient bilinear Diffie-Hellman assumption.

Keywords: Cloud Storage, Encryption, C-PRE, Diffie-Hellman.

I. INTRODUCTION

Viewing from the recent years as high-speed networks and the ubiquitous Internet access become available, as many services are provided on the Internet such that users can use them from anywhere at any time. In the concept of cloud computing, it treats the all resources as a unified entity i.e. cloud. The resource management and the computations are not concerned by the user. In the designing of the cloud system we focus on the confidentiality, functionality and the robustness. A cloud storage system is considered as a large scale distributed storage system that consists of many independent storage servers. The major requirement in the storage of the cloud computing is robustness. One way to provide data robustness is to replicate a message such that each storage server stores a copy of the message, because the message can be retrieved as long as one storage server survives. A storage server failure corresponds to an erasure error of the codeword symbol. As, the number of failure servers is under the tolerance threshold of the erasure code. The message can be recovered from the codeword symbols stored in the available storage servers by the decoding process. It provides a tradeoff between the storage size and the tolerance threshold of failure servers. A decentralized erasure code is suitable for use in a distributed storage system. Each storage server independently computes a codeword symbol for the received message symbols and stores it, after the message symbols are sent to storage servers. A user can encrypt messages by a cryptographic method before applying an erasure code method to encode and store messages, in order to provide strong confidentiality for messages in storage servers. When the user wants the message first he want to retrieve the codeword symbol from the cloud and then decode, finally he needs to decrypt the data by the cryptographic keys. In the above encryption and encoding techniques there are three problems:

a. The user has to do most computation and the communication traffic between the user and storage servers is high
b. The user has to manage his cryptographic keys
c. It is hard for storage servers to directly support other functions, besides data storing and retrieving

We address the problem of forwarding data to another user by storage servers directly under the command of the data owner. The system model that consists of distributed storage servers and key servers. A user distributes his cryptographic key-to-key servers that shall perform cryptographic functions on behalf of the user. The encryption scheme supports encoding operations over encrypted messages and forwarding operations over encrypted and encoded messages. Our system meets the requirements that storage servers independently perform encoding and re-encryption and key servers independently perform partial decryption. The notion of proxy re-encryption (PRE) was initially introduced by Blaze, Strauss and Bleumer introduced in [3]. In PRE system, Receiver is allowed to decipher public key encryptions for sender with the assistance from an authorized proxy. The proxy can then convert any cipher-text under Sender's public key into ciphertext under Receiver's public key. Proxy re-encryption has found many practical applications like encrypted email forwarding, outsourced filtering and secure distributed file Systems. The notion of conditional proxy re-encryption or C-PRE, sender can flexibly assign her delegate (Bob) the decryption capability based on the conditions attached to the messages.

II. RELATED WORK

We briefly review distributed storage systems, integrity checking mechanisms, and proxy re-encryption schemes.

Distributed Storage Systems

The Network-Attached Storage (NAS) and the Network File System (NFS) provide extra storage devices over the network such that a user can access the storage devices via network connection. A decentralized architecture for storage systems offers good scalability because a storage server can join or leave without control of a central authority. A message is encoded as a codeword and each storage server stores a codeword symbol. The failure is modeled as an erasure error of the stored codeword symbol. Each storage server linearly combines the blocks with randomly chosen coefficients and stores the codeword symbol and coefficients, to store a message of k blocks. A user queries k storage servers for the stored codeword symbols and coefficients and solves the linear system, to retrieve the message. In addition to storage servers their system consists of key servers that hold cryptographic key shares and work in a distributed way.

Proxy Re-Encryption Schemes

In a proxy re-encryption scheme, a proxy server can transfer a ciphertext under a public key PKA to a new one under another-public key PKB by using the re-encryption key. When a user wants to share his messages, he sends a re-encryption key to the storage server. Type-based proxy re-encryption schemes provide a better granularity on the granted right of a re-encryption key. In a key-private proxy re-encryption scheme, given a re-encryption key, a proxy server cannot determine the identity of the recipient.

In the pioneer work presented the first bidirectional PRE scheme i.e. refers to Remark 1 for the definitions of bidirectional/unidirectional PRE. Both of these schemes are only secure against chosen-plaintext attack. Applications often require security against chosen-ciphertext attacks, to fill the gaps presented a construction of CCA-secure bidirectional PRE scheme from bilinear pairings. In a proxy encryption scheme [4, 5, and 6], a delegator allows a delegate to decrypt ciphertext intended for her with the help of a proxy: an encryption for the delegator is first partially decrypted by the proxy and then fully decrypted by the delegate.

III. PRELIMINARIES & SCENARIO

The general scenario of the storage system that pretends the issue of the confidentiality.

System Model

Storage servers provide storage services and key servers provide key management services. As shown in the fig.1

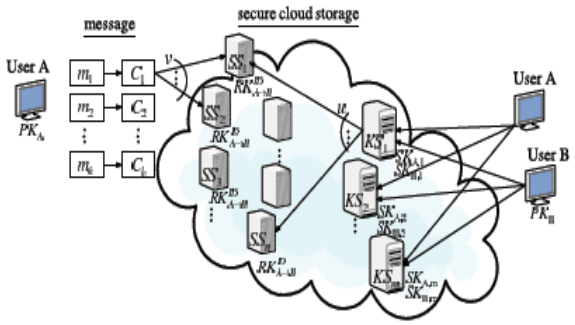

Fig. 1. A general system model of our work.

Fig.1 shows our system model consists of users, n storage servers SS1; SS2; . . . ; SSn, and m key servers KS1; KS2; . . . ; KSm. Our distributed storage system consists of four phases:

- *System setup*
  The system manager chooses system parameters and publishes in the system phase. Each user A is assigned a public-secret key pair. User A distributes his secret key SKA to key servers such that each key server KSi holds a key share SKA
- *Data storage*
  User A encrypts his message M and dispatches it to storage servers. A message M is decomposed into k blocks m1;m2; . . .;mk and has an identifier ID. Each storage server linearly depends upon receiving cipher texts from a user, combines them with randomly chosen coefficients into a codeword symbol and stores it.
- *Data forwarding*
  User A forwards his encrypted message with an identifier ID stored in storage servers to user B such that B can decrypt the forwarded message by his secret key. A uses his secret key SKA and B's public key PKB to compute a re-encryption key $RK^{ID}_{A->B}$ and then sends $RK^{ID}_{A->B}$ to all storage servers.
- *Data retrieval*
  User A requests to retrieve a message from storage servers. Each key server KSi requests u randomly chosen storage servers to get codeword symbols and does partial decryption on the received codeword symbols by using the key share SKA;I, depends on receiving the retrieval request and executing a proper authentication process with user.

Threat Model

We consider data confidentiality for both data storage and data forwarding. An attacker wants to break data confidentiality of a target user; the attacker colludes with all storage servers, non-target users, and up to (t-1) key servers. The attacker may try to generate a new re-encryption key from stored re-encryption keys. As shown in the fig.2, we formally model this attack by the standard chosen plaintext attack of the proxy re-encryption scheme in a threshold version. The challenger C provides the system parameters, after the attacker A chooses a target user *T*. The challenger gives him (t-1) key shares of the secret key SK_T of the target user T to model (t-1) compromised key servers.

A Straightforward Solution

A straightforward solution to supporting the data forwarding function in a distributed storage system is, when the owner A wants to forward a message to user B, then he downloads the encrypted message and decrypts it by using his secret key. The whole data forwarding process needs three communication rounds for A's downloading and uploading and B's downloading. The computation cost is the decryption and encryption for the owner A and the decryption for user B. The Proxy re-encryption schemes can significantly decrease communication and computation cost of the owner. The communication cost of the owner is independent of the length of forwarded message and the computation cost of re-encryption is taken care of by storage servers.

Complexity Assumptions

Let G and GT be two cyclic multiplicative groups with the same prime order q. The security of our proposed schemes is based on a complexity assumption called 3-Quotient Bilinear Diffie-Hellman assumption. Decisional version of this assumption was recently used to construct a unidirectional proxy re-encryption with chosen-ciphertext security. The n-QBDH problem in groups

$(G;G_T)$, given a tuple $(g; g^{1/a}; g^a; \ldots ; g^{(an\square 1)}; g^b)$ 2 G^{n+2} with unknown a, b.

The construction of the secure cloud storage system has introduced the algebraic settings, an erasure code over exponents. By using the bilinear map, which we assumed in the preliminaries of the conditioned proxy re-encryption.

IV. MODEL OF CONDITIONAL PROXY RE-ENCRYPTION

There are some definitions and scheme consisting of seven algorithms used in the C-PRE:

Global Setup (λ): The key generation algorithm takes as input a security parameter λ.

KeyGen (i): The key generation algorithm generates the public/secret key pair (pk_i; sk_i) for user U_i.

RKeyGen (ski; pkj): The partial re-encryption key generation algorithm takes as input a secret key ski and another public key pkj .

CKeyGen (ski;w): The condition key generation algorithm run by user i takes as input a secret key ski and a condition w.

Encrypt (pk; m;w): The encryption algorithm takes as input a public key (pk) a plaintext m 2 M and a condition w.

ReEncrypt (CTi; rki;j ; cki;w): The re-encryption algorithm run by the proxy takes as input a ciphertext CTi associated with w under public key pk_i.

Decrypt (CT; sk): The decryption algorithm takes as input a secret key sk and a ciphertext CT.

Security Notions

The semantic security of a C-PRE encryption should be preserved against both the delegate and the proxy if they do not possess the proper condition key.

Setup: Challenger C runs algorithm Global Setup (λ) and gives the global parameters param to A.

Phase 1: An adaptively issues queries q_1; …; q_m where query q_i is one of the following:

Uncorrupted key generation query (i): C first runs algorithm KeyGen (i) to obtain a public/secret key pair (pki; ski) and then sends pki to A.

Corrupted key generation query (j): C first runs algorithm KeyGen (j) to obtain a public/secret key pair (pkj ; skj) and then gives (pkj ; skj) to A.

Partial re-encryption key query (pki; pkj): C runs algorithm RKeyGen (ski; pkj) to generate a partial re-encryption key rki;j and returns it to A.

Condition key query (pki;w): C runs algorithm CKeyGen (ski;w) to generate a condition key cki;w and returns it to A.

Decryption query (pk; (w;CT)) or (pkj ;CTj): Here (pk; (w;CT)) and (pk;CT) denote the queries on original ciphertext and re-encrypted ciphertext respectively.

V. A SECURE C-PRE SCHEME

We first present our C-PRE scheme and then briefly explain the intuition behind the construction. The data confidentiality of our cloud storage system is guaranteed even if all storage servers and up to (t-1) key servers are compromised by the attacker. We analyze the simulation of the partial re-encryption key oracle. If B does not abort during the simulation of the partial re-encryption key queries and the response to A's partial re-encryption key queries is perfect. Let Abort denote the event of B's aborting during the whole simulation. The responses to adversary A's re-encryption queries are perfect and unless A can submit valid original ciphertext without querying hash function H_1. The simulation of the decryption oracle is perfect with the exception that simulation errors may occur in rejecting some valid ciphertext. For constructing the pre-secure scheme we have follows the seven algorithms in the C-PRE and the each algorithm follows the individual phase in the encryption and the encoding. It can be verified that all the correctly generated original/re-encrypted ciphertext can be correctly decrypted. Re-encrypted ciphertext generated by a proxy who does not have both the right partial re-encryption key and the right condition key.

VI. CONCLUSION

We consider a cloud storage system consists of storage servers and key servers. The threshold proxy re-encryption scheme supports encoding, partial decryption and forwarding in a distributed way. By using the threshold proxy re-encryption scheme, we present a secure cloud storage system that provides secure data storage and secure data forwarding

functionality in a decentralized structure. Each storage server independently performs encoding and re-encryption and each key server independently perform partial decryption. we tackle the problem of how to control the proxy in PRE systems at a fine-grained level. We formalize its definition and its security notions and propose a CCA-secure C-PRE scheme. Further, we extend this C-PRE scheme to support multiple conditions with reasonable overhead. It remains as an interesting open problem how to construct CCA-secure C-PRE schemes with anonymous conditions or boolean predicates.

VII.REFERENCE

[1] J. Kubiatowicz, D. Bindel, Y. Chen, P. Eaton, D. Geels, R.Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, and B. Zhao, "Oceanstore: An Architecture for Global-Scale Persistent Storage," Proc. Ninth Int'l Conf. Architectural Support for Programming Languages and Operating Systems (ASPLOS), pp. 190- 201, 2000.

[2] G. Ateniese, K. Fu, M. Green, and S. Hohenberger. Improved Proxy Re-encryption Schemes with Applications to Secure Distributed Storage. In Proc. of NDSS 2005, pp. 29-43, 2005.

[3] M. Blaze, G. Bleumer, and M. Strauss. Divertible Protocols and Atomic Proxy Cryptography. In advances in Cryptology-Eurocrypt'98, LNCS 1403, pp. 127-144, Springer-Verlag, 1998.

[4] Y. Dodis, and A.-A. Ivan. Proxy Cryptography Revisited. In Proc. of NDSS'03, 2003.

[5] M. Jakobsson. On Quorum Controlled Asummetric Proxy Re-Encryption. In Proc. of PKC'99, LNCS 1560, pp.112-121, Springer-Verlag, 1999.

[6] Masahiro Mambo and Eiji Okamoto. Proxy Cryptosystems: Delegation of the Power to Decrypt Ciphertexts. IEICE Trans. Fund. Electronics Communications and Computer Science, E80-A/1:54-63, 1997.

[7] G. Ateniese, K. Fu, M. Green, and S. Hohenberger. Improved Proxy Re-encryption Schemes with Applications to Secure Distributed Storage. ACM Transactions on Information and System Security (TISSEC), 9(1):1-30, February 2006.

[8] P. Druschel and A. Rowstron, "PAST: A Large-Scale, Persistent Peer-to-Peer Storage Utility," Proc. Eighth Workshop Hot Topics in Operating System (HotOS VIII), pp. 75-80, 2001.

[9] A. Adya, W.J. Bolosky, M. Castro, G. Cermak, R. Chaiken, J.R. Douceur, J. Howell, J.R. Lorch, M. Theimer, and R. Wattenhofer, "Farsite: Federated, Available, and Reliable Storage for an Incompletely Trusted Environment," Proc. Fifth Symp. Operating System Design and Implementation (OSDI), pp. 1-14, 2002.