Enhancing Cloud Storage Security against Roll-back Attacks with A New Fair Multi-Party Non-Repudiation Protocol

[†]Jun Feng, [†]Yu Chen*, [†]Douglas Summerville, [‡]Wei-Shinn Ku, [§]Zhou Su
[†]Dept. of Electrical and Computer Engineering, SUNY - Binghamton, Binghamton, NY 13902, USA
[‡]Dept. of Computer Science and Software Engineering, Auburn University, Auburn, AL 36849, USA
[§]Dept. of Computer Science, Waseda University, Ohkubo 3-4-1, Shinjyuku, Tokyo 169-8555, Japan

***Abstract* - Along with variant advantages, cloud storage also poses new security challenges. Potential users are reluctant to move important and sensitive data to cloud unless security challenges have been well addressed. This paper reports our on-going efforts to address three data security issues in cloud storage: repudiation, fairness, and roll-back attacks. We proposed a novel fair multi-party non-repudiation (MPNR) protocol, which provide a fair non-repudiation storage cloud and is capable of preventing roll-back attacks.**

Keywords: Cloud Storage, Non-repudiation, Roll-back Attack.

1. Introduction and Security Problems

Cloud storage systems can meet basic requirements of mass storage and low expense. However, users are reluctant to move important and sensitive data to cloud unless security challenges have been well addressed. Recently, secure cloud storage architectures have been proposed [5], [6]. In these architectures, confidentiality is achieved by encryption, and data integrity is protected using message digests. In addition, non-repudiation is supported by signed message-digests, freshness is achieved by periodic audit, and write-serializability is guaranteed by chain hash. SUNDR can be used to defend "fork consistency attack". Broadcast encryption and key rotation are used to improve scalability. However, the most crucial aspect of cloud storage is that none of the peers is trustful. This leads to integrity vulnerabilities [3] that can potentially cause more security problems, such as fairness, disputation, and roll-back attacks.

Although signed digests can be used as non-repudiation evidence in cloud storages [3], [6], they did not consider "Fairness" [7]. Reference [6] only considered data should be certified with non-repudiation. It is the same as what current cloud platforms does [3]. Reference [3] considered both peers should have non-repudiation evidence, but it is not "fair" since any party can refuse to send its own certification after he receives the sender's certification. "Roll-back" is a specific case of "fork consistency attack". Although SUNDR [4] can solve such problem, the essential prerequisite is not satisfied in storage cloud since peers do not trust each other.

Suppose there are three entities involved in cloud storage. The data owners store important or sensitive data to cloud and pay for the service. Service providers provide secure storage services and obtain profit. Data users fetch data from cloud storage and pay for the service. Only owners can decide and change access control polices. Since none of the peers is trustworthy in cloud, there are several important security concerns arise.

1. Disputation: If a user gets data through cloud service provider and the user claims that the data is tampered. The innocent entity needs evidence to defend against false accusations. And it is desired to find the peer who is responsible for the fault.
2. Fairness: During the data transmission procedure, in order to gain certain advantages, malicious party may refuse to response after receiving the evidence from other peers.
3. Roll-back attacks: When the data owner has updated the data set with a newer version, the malicious service provider still provide older version to the users who download the data. It is difficult for users to detect it.

Extended from our two-party non-repudiation protocol [2], this paper proposed a new fair multi-party non-repudiation (MPNR) protocol to address these problems.

2. A New Fair MPNR Protocol

This section presents a new fair MPNR protocol. The MPNR protocol solves the problems of fair non-repudiation and roll-back attacks. Each message consists of specified data transmission information as evidence: NRR (Non-Repudiation of Receipt) or NRO (Non-Repudiation of Origin). Furthermore, the data can be divided into blocks and encrypted with different keys. Each individual user only has keys for authorized cipher text blocks.

2.1 Non-Repudiation Process

The MPNR needs three rounds. The rounds are between data owner and service provider, data owner and users, users and service provider. Each round has two steps. Originator starts a task with NRO. Receiver returns a NRR. Owner or users can be the originator. Provider or users can be the receiver. Each round has two modes: Normal mode and Resolve mode. Normal mode is similar to that in references [1] . It supposes the two peers in one round are willing to exchange messages and non-repudiation evidence. When the originator fails to obtain the non-repudiation evidence, TTP (Trusted Third Party) will be invoked in Resolve mode.

2.1.1 Normal session

In a normal data transmission, it takes six steps as follows shown in Figure 1.

* Manuscript submitted on Oct. 1, 2010 to *the 8th IEEE Consumer Communications & Networking Conference, Work In Progress Track* (CCNC 2011 Short Position Paper), Las Vegas, NV., USA, Jan. 9 – 12, 2011. Corresponding author: Yu Chen, SUNY – Binghamton, Binghamton, NY 13902. E-mail: ychen@binghamton.edu, Tel.: (607) 777-6133.

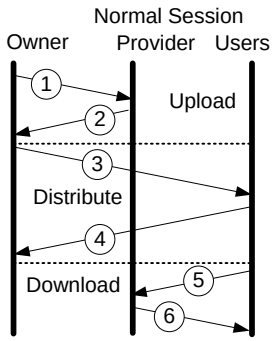

Figure 1. Normal Session working flow analysis.

1. Owner uploads data to storage with NRO_{OP} message. OP stands for sending from owner to service provider.

2. Service provider checks the message. If it is valid, provider responds with NRR_{PO}, here PO stands for sending from service provider to owner. Otherwise, the provider responds with an "ERROR" message.

3. On receiving the NRR_{PO}, the owner sends both NRO_{OP} and NRR_{PO} to users by group encryption [1] with NRO_{OU}.

4. If the message is valid, users respond with NRR_{UO}. Otherwise, users send back an "ERROR" message.

5. Any user can send "download" request to Cloud provider with NRR_{UP}

6. If received message is valid, service provider responds with NRR_{PU} and data. Otherwise, an "ERROR" message will be sent.

2.1.2 Resolve Session

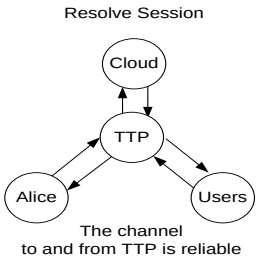

Figure 2. Abnormal Session working flow analysis.

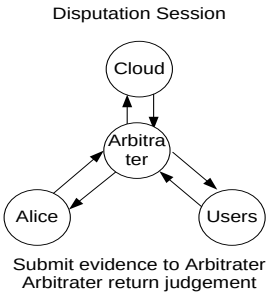

Figure 3. Disputation Session working flow analysis.

After steps 1, 3 or 5, if originator has not received response when time out, a Resolve session can be started with TTP through reliable channels. Then receivers cannot deny the reception of messages through TTP. There are two possibilities. One is that the receiver responds with NRR. The other is that if the receiver refuses to respond, TTP will generate a NRR and send back to the originator. Therefore, originator can always get a NRR. The Resolve session is shown in Figure 2.

2.1.3 Disputation Session

If there is any disputation, as shown in Figure 3, any of the entities can provide non-repudiation evidences to Arbitrator to prove their innocence.

2.2 Discussions

Let's consider the security concerns listed in Section 1 again. Regarding the problem of disputation, each peer can have NRO or NRR at end of a round. It is different from that in reference [6], where only receiver can have the non-repudiation evidence. Fairness is achieved in this MPNR protocol with the help of TTP. Both parties can get NRO or NRR as evidence at the end of a round. In this structure, owner distributes the up-to-date signed root hash to users directly. For higher efficiency, a Merkle hash tree can be integrated to generate a root hash of the data, and then sign the root hash as the evidence of data integrity. Each time when owner updates the data, the root hash is updated and distributed to users in step 3. When a user downloads the updated data from cloud, the consistency can be easily verified by checking the root hash from provider and root hash from owner. Therefore the roll-back attacks can be detected if cloud gives an old version. Thus the integrity evidence prevents the service provider from obtaining any benefit by launching a roll-back attack.

3. Conclusions

Data storage has been considered as one of the major profitable applications in cloud computing. However, it cannot be accepted widely if the security is not guaranteed. In this paper, a MPNR protocol has been proposed that has the capability to address three important concerns in cloud storage platforms. Our solution is based on the non-repudiation according to the cloud conditions and can enhance the security of cloud storage.

References

[1] G. Chiou and W. Chen, "Secure broadcasting using the secure lock," *IEEE Transactions on Software Engineering*, vol. 15, no. 8, pp.929-934, Aug. 1989.

[2] J. Feng, Y. Chen, W.-S. Ku, and P. Liu, "Analysis of Integrity Vulnerabilities and a Non-repudiation Protocol for Cloud Data Storage Platforms," the *2nd International Workshop on Security in Cloud Computing* (SCC 2010), in conjunction with ICPP 2010, San Diego, California, USA, Sept. 14, 2010.

[3] J. Feng, Y. Chen, and P. Liu, "Bridging the Missing Link of Cloud Data Storage Security in AWS", the *7th IEEE Consumer Communications and Networking Conference - Security for CE Communications* (CCNC'10), Las Vegas, Nevada, USA, January 9 - 12, 2010.

[4] J. Li, M. Krohn, D. Mazieres, and D. Shasha. "SUNDR: Secure untrusted data repository", In *OSDI*, 2004.

[5] S. Kamara and K. Lauter. "Cryptographic cloud storage", In *ACM Workshop on Cloud Security*, 2009.

[6] R.A. Popa, J. Lorch, D. Molnar, H.J.Wang, and L. Zhuang, "Enabling Security in Cloud Storage SLAs with CloudProof", *Microsoft TechReport MSR-TR-2010-46*, May, 2010.

[7] J. Zhou and D. Gollmann, "A Fair Non-repudiation Protocol," Proceedings of *1996 IEEE Symposium on Security and Privacy*, pp 55--61, Oakland, USA, May 1996.