A Proposal for Implementing Mobile Applications Using Untrusted Servers

Meng LI[1] and Yasushi SHINJO[1]

1. Introduction

Conventional mobile applications are built based on a client-server mode. This requires central servers for storing shared data. The users of such mobile applications must fully trust central servers and their service providers. Once a server is compromised by hackers, user information may be revealed because data is often stored on the server in cleartext. Users may lose their data when service providers shut down their services.

We are implementing a mobile application that is not relied on trusted central servers. Concretely, we are developing a group finance manager application, Grouper, using multiple untrusted servers for data transfer. Data is divided into several pieces and uploaded to diverse servers. Each server can only keep a piece of data temporarily. A piece will be deleted after a period of time in order to protect user data. In addition, all devices of group members keep a complete data set, and data can be recovered even untrusted servers shut down.

2. Design

We design Grouper running on mobile devices. This does not rely on trusted central servers.

2.1 Group Finance Manager

In Grouper, users can create financial records including income and expenditure in their own devices. With data synchronization, they can also share their records with other group members through untrusted servers, so group income and expenditure information are analyzed and shown to all members. To create a group in Grouper, the owner of this group registers and sets access keys in untrusted servers and passes them to group members by a face-to-face way.

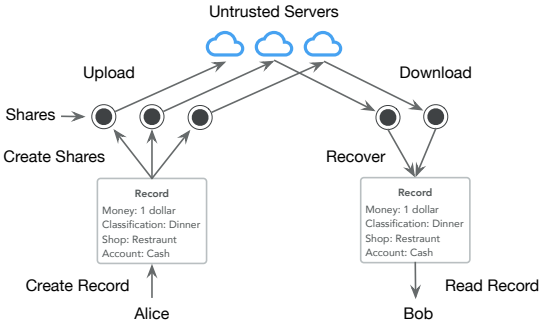

Figure 1. Flow of synchronization

2.2 Shamir's Secret Sharing

Grouper uses a secret sharing scheme to protect data in untrusted servers [1]. In a secret sharing scheme, a dealer securely shares a secret with a group of participants by generating n shares using a cryptographic function. At least k or more shares can recover the secret, but $k-1$ or fewer shares can obtain nothing about the secret. We use Shamir's secret sharing that is a popular technique to implement such a threshold scheme.

2.3 Data Synchronization Using Multiple Untrusted Servers

We design our application Grouper based on data synchronization through multiple untrusted servers rather than a single server. There are three principles in our design.

Firstly, a server transfers data as similar to a router, but does not keep data permanently. Grouper uses untrusted servers as a bridge for transferring data. Secondly, a server keeps data temporarily. We define a period of time in which data can be kept in a server. Thirdly, servers do not know the content of data. Keeping data temporarily cannot ensure data security, because servers know the cleartext of data in this temporary period. In Grouper, we use secret sharing to protect data security.

Figure 1 describes the flow of synchronization

[1] University of Tsukuba

in Grouper. At first, Alice adds a record and creates three shares by a secret sharing scheme where n is three and k is two. Next, Grouper uploads those shares to untrusted servers. When Bob is online, Grouper in Bob's device downloads two shares from servers and recovers the new record created by Alice.

2.4 Reliable Synchronization

Grouper provides a reliable synchronization service. For example, a user in a group creates a new record in her device, all of other members in the group should synchronize this record, even if this record may be deleted by untrusted servers after a period of time. We call this problem *reliable synchronization*.

To realize reliable synchronization, a record creator should upload his new record until all of other members download this new record from untrusted servers successfully. However, if we use a naive protocol, a member can attack this group by being lazy. This forces the record creator to upload shares to untrusted servers indefinitely. We are addressing this problem now by using a reliable multicast technique.

3. Implementation

Grouper consists of clients that run an iOS application and multiple untrusted servers that run a Web service.

In clients, Grouper uses Core Data [2], a native iOS framework to manage model layer objects. Sync [3] is a modern JSON synchronization framework for Core Data, and performs data synchronization by generating and parsing JSON strings. c-SSS [4] is an implementation of Shamir's secret sharing in the C language. We use it to generate shares from a JSON string.

In servers, the Web service of Grouper provides three functions. Firstly, the Web service supports reliable synchronization. Secondly, the Web service ensures that shares are deleted after a prescriptive time. Thirdly, the Web service allows only group members who have access keys to download shares.

4. Related Work

DepSky [5] is a system that stores encrypted data on servers and runs application logic on the logic server. DepSky keeps encrypted data in some commercial storage services and performs application logic in individual servers. In Grouper, untrusted servers undertake the responsibility of temporarily data storage and message delivery with server-side computation.

Compared with conventional applications and frameworks that use untrusted network and servers, the key feature of Grouper is the use of secret sharing and temporary data storage. This is more convenient and faster than using data encryption because clients need not to distribute decryption keys and perform heavy encryption computation.

5. Conclusion

This paper describes Grouper, a group finance manager that synchronizes data among mobile devices with multiple untrusted servers. Grouper uses secret sharing and temporary data storage services. Grouper consists of clients that run an iOS application and multiple untrusted servers that run a Web service. Each server of Grouper does not know the others and keeps one piece of shares generated by a secret sharing scheme temporarily to protect data. We are designing a robust protocol of reliable synchronization using untrusted servers.

References

[1] Liao-Jun Pang and Yu-Min Wang. A new (t, n) multi-secret sharing scheme based on Shamir's Secret Sharing. Applied Mathematics and Computation, 167(2):840–848, 2005.

[2] Apple Inc. Core Data Programming Guide. https://developer.apple.com/library/content/documentation/Cocoa/Conceptual/CoreData.

[3] Elvis Nunez. Sync. https://github. com/SyncDB/Sync.

[4] Fletcher T. P. c-SSS. https://github.com/fletcher/c-sss.

[5] Alysson Bessani, Miguel Correia, Bruno Quaresma, Fernando Andre, and Paulo Sousa. DepSky: dependable and secure storage in a cloud-of-clouds. ACM Transactions on Storage (TOS), 9(4):12, 2013.