Security Preserving Access Control Mechanism In Public Clouds Using PANDA Security Mechanism

Mrunali Pingale
Department of Information Technology
Siddhant College of Engineering, Sudumbare.
Pune, Maharashtra
mrunali_pingale@yahoo.com

Prof. Jyoti Pingalkar
Department of Information Technology
Siddhant College of Engineering, Sudumbare.
Pune, Maharashtra
jyoti_pingalkar@rediffmail.com

Abstract— Users in a particular group need to compute signatures on the blocks in shared data, so that the shared data integrity can be confirmed publicly. Various blocks in shared data are usually signed by various vast number of users due to data alterations performed by different users. Once a user is revoked from the group, an existing user must resign the data blocks of the revoked user in order to ensure the security of data. Due to the massive size of shared data in the cloud, the usual process, which permits an existing user to download the corresponding part of shared data and re-sign it during user revocation, is inefficient. The new public auditing scheme for shared data with efficient user revocation in the cloud is proposed so that the semi-trusted cloud can re-sign the blocks that were previously signed by the revoked user with the valid proxy re-signatures, when a user in the group is revoked.

Keywords— Public auditing, shared data, user revocation, cloud computing

I. INTRODUCTION

Users in a particular group need to compute signatures on the blocks in shared data, so that the shared data integrity can be confirmed publicly. Various blocks in shared data are usually signed by various vast number of users due to data alterations performed by different users. Once a user is revoked from the group, an existing user must resign the data blocks of the revoked user in order to ensure the security of data. Due to the massive size of shared data in the cloud, the usual process, which permits an existing user to download the corresponding part of shared data and resign it during user revocation, is inefficient. The new public auditing scheme for shared data with efficient user revocation in the cloud is proposed so that the semi-trusted cloud can re-sign the blocks that were previously signed by the revoked user with the valid proxy re-signatures, when a user in the group is revoked.

With cloud computing and storage, users are able to access and to share resources offered by cloud service providers at a lower marginal cost. Data is stored in the cloud, and shared among a group of users in a collaborative manner. It is natural for users to wonder whether their data remain intact over a prolonged period of time: due to hardware failures and human errors in an untrusted cloud environment, the integrity of data stored in the cloud can become compromised. To protect the integrity of data in the cloud and to offer peace of mind to users, it is best to introduce a third party auditor (TPA) to perform auditing tasks on behalf of users. Such a third party auditor enjoys amply computation/communication resources that users may not possess. Provable data possession (PDP), first proposed by, allows a verifier to perform public auditing on the integrity of data stored in an untrusted server without retrieving the entire data. Subsequent work focused on how dynamic data and data privacy can be supported during the public auditing process. However, most of previous works only focus on auditing the integrity of personal data. A privacy-preserving public auditing mechanism for shared data in an untrusted cloud, so that the identity of the signer on each block in shared data is not disclosed to the third party auditor (TPA) during an auditing task. By preserving identity privacy, the TPA cannot figure out which user in the group or which block in shared data is a higher valuable target than others. In Paper, information used for verification are computed with ring signatures; as a result, the size of verification information, as well as the time it takes to audit with it, are linearly increasing with the number of users in a group. To make matters worse, when adding new users to a group, all the existing verification information will need to be re-computed if ring signatures are used, introducing a significant computation burden to all users. In addition, the identities of signers are unconditional protected by ring signatures, which prevent the group manager to trace the identity when someone in the group is misbehaved. In this paper, we propose a new privacy-preserving mechanism to audit data stored in an untrusted cloud and shared among a large number of users in a group. We take advantage of group signatures to construct homomorphic authenticators, so that the third party auditor is able to verify the integrity of shared data without retrieving the entire data, but cannot reveal the identities of signers on all blocks in shared data. Meanwhile, the size of verification information, as well as the time it takes to audit with it, are not affected when the number of users sharing the data increases.
The original user, who creates and shares the data in the cloud, is able to add new users into a group without re-computing any verification information. In addition, the original user (acts as the group manager) can trace group signatures on shared data, and reveal the identities of signers when it is necessary. We also utilize homomorphic MACs to effectively reduce the amount of storage space needed to store verification

information. As a necessary trade-off, we allow the third party auditor to share a secret key pair with users, which we refer to as authorized auditing. Although we allow an authorized TPA to possess the secret key pair, the TPA cannot compute valid group signatures as group users because this secret key pair is only a part of a group users private key. To our best knowledge, we present the first mechanism designed with scalability in mind when it comes to support auditing data shared among a large number of users in a privacy-preserving fashion.

II. LITERATURE SURVEY

Statistical Comparisons of Classifiers over Multiple Data Sets (2013) In this method introduce some new pre - or post processing step has been proposed, and the implicit hypoth-esis is made that such an enhancement yields an improved performance over the existing classification algorithm. Alternatively, various solutions to a problem are proposed and the goal is to tell the successful from the failed. A number of test data sets is selected for testing, the algorithms are run and the quality of the resulting models is evaluated using an appropriate measure, most commonly classification accuracy. The remaining step, and the topic of this paper, is to statistically verify the hypothesis of improved performance. Various researchers have addressed the problem of comparing two classifiers on a single data set and proposed several solutions. The core of the paper is the study of the statistical tests that could be (or already are) used for comparing two or more classifiers on multiple data sets.Learning algorithms is used for the Classification purpose. The main disadvantage of this process is the problems with the multiple data set tests are quite different, even in a sense complementary [3].

It used six real world dataset from the UCI repository have been used. Three of them have classification Problem with discrete features, the next two classifications with discrete and continuous features, and the last one is approximation problem. The learning algorithm is used to check the quality of feature selected are a classification and regression tree layer with pruning. This process and algorithms is implemented by the orange data mining System. Overall, the non parametric tests, namely the Wilcox on and Friedman test are suitable for our problems. They are appropriate since they assume some, but limited commensurability. They are safer than parametric tests since they do not assume normal distributions or homogeneity of variance. There is an alternative opinion among statisticians that significance tests should not be per - formed at all since they are often misused, either due to misinterpretation or by putting too such stress on their results The main disadvantage of the system is it measure to low accuracy of the search process [3].

The introduction of TPA reduces the involvement of the client through the auditing of the validity of data stored in the cloud. Block modification, insertion, and deletion, which is significant step toward reality verifies the integrity of the data stored in the cloud, because services in Cloud Computing are not confined to backup data only(2013). Previous works on ensuring remote data integrity misses the support of public auditability or dynamic data operations and thus our paper addresses both the difficulties. We find out the difficulties and security hassles of extensions with dynamic data updates from previous works and then show how to construct an efficient verification schema for the flawless integration of these two important features in protocol design. We enhance the prevelant proof of storage models by editing the classic Merkle Hash Tree construction for data authentication to gain efficient data dynamics[5],. To support efficient handling of multiple auditing tasks, we also find out the technique of bilinear aggregate signature to extend our outcome into a multiuser setting, where TPA can perform simultaneous auditing tasks.The proposed schemes are highly efficient and also shielded[2]. Through a proof-of-retrievability system, a data storage center confirms that he is actually storing all of a client's data.

The central ultimate test is to build systems that are both client and shielded. We give the proof-of-retrievability schemes with full proofs of security against arbitrary adversaries in the strongest model, that of Juels and Kaliski. scheme, built from BLS signatures and shielded in the oracle model, having the shortest query and response of any proof-of-retrievability with public very ability. Our second plan, which exploits efficiently on pseudorandom functions (PRFs) and is shielded in the standard model, has the least response of any proof-of-retrievability scheme with private efficiency (but a longer query)[2]. Both schemes rely on homomorphic properties to sort a proof into a single small authenticator value. The collusion-resistant proxy re-signature schemes usually have two levels of signatures, are in different forms and need to be confirmed differently, achieving blockless verifiability on both of the two levels of signatures and verifying them together in a public auditing mechanism is contributed(2012).

In this paper, we propose a novel public auditing mechanism for the integrity of shared data with efficient user revocation in an untrusted cloud. In our mechanism, by utilizing the idea of proxy re-signatures [11], once a user in the group is revoked , the cloud is able to re-sign the blocks, which were signed by the revoked user, with a re-signing key. As a result, the efficiency of user revocation can be significantly improved, and computation and communication resources of existing users can be easily saved. Meanwhile, the cloud, who is not in the same trusted domain with each user, is only able to convert a signature of the revoked user into a signature of an existing user on the same block, but it cannot sign arbitrary blocks on behalf of either the revoked user or an existing user. By designing a new proxy re-signature scheme with nice properties, which traditional proxy re-signatures do no have, even after the cloud re-signs any block, a public verifier is always able to check the integrity of shared data without retrieving the entire data from the cloud. Generally, the integrity of shared data is threatened by three factors.First, the cloud service provider may inadvertently pollute shared data due to hardware/software failures and human errors. Second, an external adversary may try to corrupt shared data in the cloud, and prevent users from using shared data correctly. Third, a revoked user, who no longer has the right as existing users, may try to illegally modify shared data.Considering these threats, users do not fully trust the cloud with the integrity of shared data. To protect the integrity of shared data, each block in shared data is attached with a

signature, which is computed by one of the users in the group. When shared data is initially created by the original user in the cloud, all the signatures on shared data are computed by the original user. After that, once a user modifies a block, this user also needs to sign the modified block with his/her own private key. By sharing data among a group of users, different blocks may be signed by different users due to modifications from different users[4] and various other database parameters.

III. PROBLEM DEFINITION AND MOTIVATION

A. Problem Definition

Expected support as the measurement of pattern frequentness, which has inherent weak-nesses with respect to the underlying probability model, and is therefore ineffective for mining high-quality sequential patterns from uncertain sequence databases.

B. Motivation

To quantify design recurrence focused around the conceivable world semantics. This paper secure two indeterminate sequence information models dreamy from numerous genuine applications involving dubious arrangement information, and plan the issue of mining probabilistically visit successive examples (or p-Fsps) from information that adjust to our models. Taking into account the prex-projection methodology of the well known Prexspan calculation, [4] this paper create two new calculations, all things considered called U-Prexspan, for p-FSP mining. U- Prexspan adequately stays away from the issue of ”conceivable world explosion”, and when joined with our three pruning methods and one accepting system, accomplishes great execution. The efficiency and viability of U-Prexspan are varied through extensive probes. [4][5][6]

C. Mathematical Model

Assumption:

Given, two groups are there

G1 = A,BC,D,E

G2=F,G,H,I,K

Hence Bilinear Map .....

e(5,5): G2 = A,B,C,D,E,F,G,H,,I,K

e(5,5)!= 1

1. q- Strong SPEKE Assumption : (1/ (1+1))*3

where x=1 belongs to any integer no in (1 to 10).

Why (1 to 10) .....Because total elements in G2 = 10.

where q = modulo

2. DL Assumption : a+b=c (5+5)=10 3. SPEKE ASSUMPTION :

A Real No..

IV SYSTEM DESCRIPTION AND IMPLEMENTATION

A. System Design

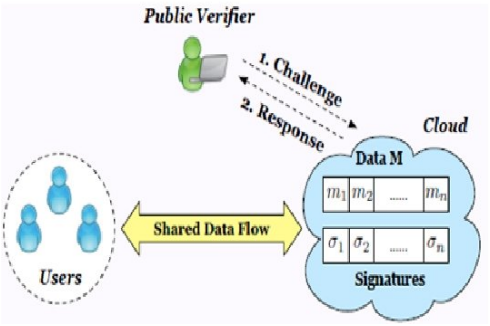

Figure 4.1: System Diagram

Three entities are involved in project : the cloud, the public verifier, and users (who share data as a group). The cloud offers data storage and sharing services to the group. The public verifier, such as a client who would like to utilize cloud data for particular purposes (e.g., search, computation, data mining, etc.) or a third-party auditor (TPA) who can provide verification services on data integrity, aims to check the integrity of shared data via a ultimate test-and response protocol with the cloud. In the group, there is one original user and anumber of group users. The original user is the original owner of data. This original user creates and shares data with other users in the group through the cloud. Both the original user and group users are able to access, download and modify shared data. Shared data is divided into a number of blocks. A user in the group can modify a block in shared data by performing an insert, delete or update operation on the block. In this paper, we assume the cloud itself is semi-trusted, which means it follows protocols and does not pollute data integrity actively as a malicious adversary, but it may lie to verifiers about the incorrectness of shared data in order to save the reputation of its data services and avoid losing money on its data services. In addition, we also assume there is no collusion between the cloud and any user during the design of mechanism. Usually, the incorrectness of share data under the above semitrusted model can be introduced by hardware/software failures or human errors happened in the cloud. Considering these factors, users do not fully trust the cloud with the integrity of shared data. To protect the integrity of shared data, each block in shared data is attached with a signature, which is computed by one of the users in the group. Specifically, when shared data is initially created by the original user in the cloud, all the signatures on shared data are computed by the original user. After that, once a user modifies a block, this user also needs to sign the modified block with his/her own private key. By sharing data among a group of users, different blocks may be signed by different users due to modifications from different users. When a user in the group leaves or misbehaves, the group needs to revoke this user. Usually, as the creator of shared data, the original user acts as the group manager and is able to revoke users on behalf of the group. Once a user is revoked, the signatures computed by this revoked user become invalid to the group, and the blocks that were previously signed by this revoked user should be re-signed by an existing users private key, so that the correctness

of the entire data can still be confirmed with the public keys of existing users only.

B. System Implementation

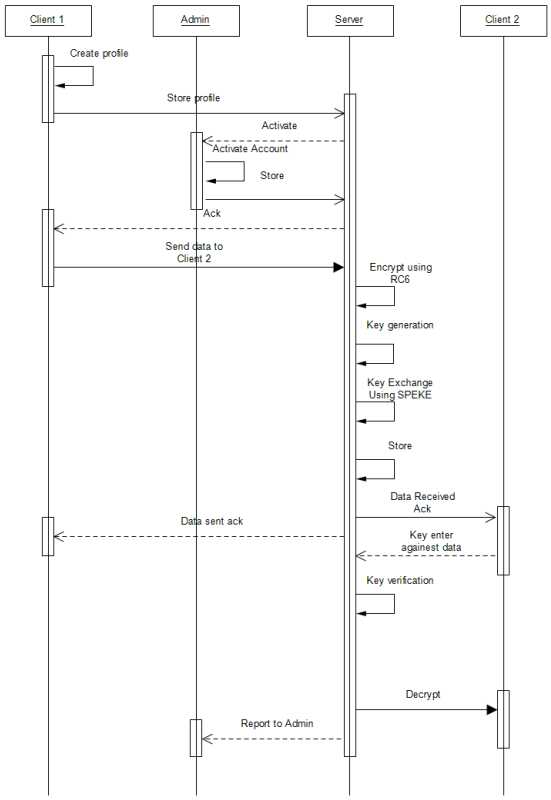

Figure 4.2: Sequence diagram for System workflow

V PROPOSED ALGORITHM

A. SPEKE

SPEKE is one of the older and well-known protocols in the relatively new field of password-authenticated key exchange. It was first described by David Jablon in 1996.[1] In this publication Jablon also suggested a variant where, in step 2 of the protocol, g is calculated as g =gqS with a constant gq. However, this construction turned out to be insecure against dictionary attacks and was therefore not recommended anymore in a revised version of the paper.

In 1997 Jablon refined and enhanced SPEKE with additional variations, including an augmented password-authenticated key agreement method called B-SPEKE.[2] Since 1997 no flaws have been published for SPEKE. A paper published by MacKenzie in 2001 presents a proof in the random oracle model that SPEKE is a secure PAKE protocol (using a some what relaxed definition) based on a variation of the Decision Diffie-Hellman assumption.

VI CONCLUSION

In this project, we proposed an innovative public auditing system for shared data with efficient user revocation in the cloud. Whenever a user in the group is revoked, we allow the semi-trusted cloud to re-sign blocks that were signed by the revoked user with proxy resignatures.The outcomes show that the cloud can enhance the efficiency of user revocation, and existing users in the group can save a significant amount of computation and communication resources during user revocation. Thus our project focuses on achieving efficiency and save users time by allowing the cloud to sign those data blocks.

VII REFERENCES

[1] N. Cao, S. Yu, Z. Yang, W. Lou, and Y. T. Hou, LT Codes-based Secure and Reliable Cloud Storage Service, inthe Proceedings of IEEE INFO-COM 2012,2012,pp.693701.

[2] 2. J. Yuan and S. Yu, Proofs of Retrievability with Public Verifiability and Constant Communication Cost in Cloud, in Proceedings of ACM ASIACCS-SCC13,2013.

[3] 3. H. Wang, Proxy Provable Data Possession in Public Clouds,IEEE Transactions on Services Computing,accepted.

[4] B. Wang, B. Li, and H. Li, Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud, inthe Proceedings of IEEE Cloud 2012,2012,pp.295302.

[5] 5. S. R. Tate, R. Vishwanathan, and L. Everhart, Multi-user Dynamic Proofs of Data Possession Using Trusted Hardware, in Proceedings of ACM CODASPY13,2013,pp.353364.

[6] 6. B. Wang, B. Li, and H. Li, Knox: Privacy-Preserving Auditing for Shared Data with Large Groups in the Cloud, inthe Proceedings of ACNS 2012, June 2012, pp. 507525.

[7] 7. M. Blaze, G. Bleumer, and M. Strauss, Divertible Protocolsand Atomic Proxy Cryptography, inthe Proceedings of EUROCRYPT 98.Springer Verlag,1998,pp.127144.

[8] 8. A. Shamir, How to share a secret, in Communication of

ACM, vol. 22, no. 11, 1979, pp. 612613.

[9] 9. B. Wang, H. Li, and M. Li, Privacy-Preserving Public Auditing for Shared Cloud Data Supporting Group Dynamics, in the Proceedings of IEEE ICC 2013,2013.

[10] B. Wang, S. S. Chow, M. Li, and H. Li, Storing Shared Data on the Cloud via Security-Mediator, in Proceedings of IEEE ICDCS 2013,2013.