Enabling Public Verifiability for Shared Data with Efficient User Revocation in the Cloud

Deeksha.V[1], Madankumar.G[2], Saravana Perumal.V.M[3]
[1, 2,] UG students, department of computer science, Rajarajeswari college of engineering
India
[3] Professor, department of computer science, Rajarajeswari college of engineering
India

***Abstract*- with data storage and sharing services provided by the cloud, people work together as a group by sharing data with each other. Once the shared data is created in the cloud, users in the group are able to access and modify and also share the latest updated data with the rest of the group. When users put their data (of large size) on the cloud, the user will be having a security issues like data integrity. To safeguard the shared data integrity, that can be verified publicly. Users in the group need to compute signatures on all the blocks in shared data. Different blocks in shared data are generally signed by different users due to data modifications performed by different users. For security reasons, once a user is repudiated from the group, the blocks which were previously signed by this revoked user must be re-signed by an existing user. The straightforward method that allows an existing user to download the corresponding part of shared data and re-sign it during user repudiation, is not efficient due to the large size of shared data in the cloud. In this paper, we propose an auditing mechanism for the integrity of shared data with efficient user revocation. By utilizing the idea of proxy re-signatures, we allow the cloud to re-sign blocks on behalf of existing users during user revocation, so that existing users need not to download and re-sign blocks. Additionally, a public verifier is often ready to audit the integrity of shared information. And Moreover, our mechanism is also able to support batch auditing by verifying multiple auditing tasks at the same time.**

I. INTRODUCTION

With data storage and sharing services provided by the cloud, people work together as a group by sharing the data. Once a user creates shared data within the cloud, the each user in the group is not only able to access and modify shared information, but also share the latest version of the shared information with the rest of the group. As cloud providers promise a more secure and reliable environment to the users, the integrity of data in the cloud may still be compromised, due to the existence of hardware/software failures and human errors. To protect the integrity, various numbers of mechanisms have been proposed. We discuss a mechanism which include, a signature that is attached to each block in data and the integrity of data relies on the correctness of all the signatures. One amongst the foremost significant and common options of these mechanisms is to permit a public verifier to efficiently check data integrity within the cloud while not downloading the complete data, remarked as public auditing.

This public verifier could be 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 is able to provide verification services on data integrity to users. Most of the previous mechanisms proposed not consider the efficiency of user revocation when auditing the correctness of shared data in the cloud. With shared data, once a user modifies a block, she also needs to compute a new signature for the modified block. Due to the modifications by the different users, different blocks are signed by different users. For security reasons, when a user leaves the group or misbehaves, this user must be repudiated from the group. As a result, this revoked user should no longer be able to access and modify shared data, and the signatures generated by this revoked user are no longer valid to the group.

Since shared data is outsourced to the cloud and users no longer store it on local devices, a straightforward method to re-compute these signatures during user repudiation (as shown in Figure1) is to ask an existing user (i.e., Alice) to first download the blocks previously signed by the repudiated user (i.e., Bob), verify the correctness of these blocks, then re-sign these blocks, and finally upload the new signatures to the cloud. However, this straight forward method may cost the existing user a huge amount of communication and computation resources by downloading and verifying blocks, and by re-computing and uploading signatures, especially when the number of re-signed blocks is quite large or the membership of the group is frequently changing.

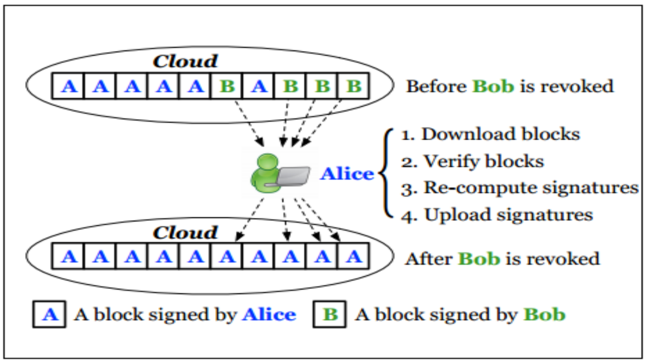


Fig.1. Alice and Bob share data in the cloud. When Bob is revoked, Alice re-signs the blocks that were previously signed by Bob with her private key.

Clearly, if the cloud might possess every user's personal key, it will simply end the resigning task for existing users while not asking them to transfer and re-sign blocks. However, since the cloud isn't within the same trusted domain with every user within the group, outsourcing every user's personal key to the cloud would introduce significant security problems. Another vital downside we need to contemplate is that the re computation of any signature throughout user revocation ought to not have an effect on the most attractive property of public auditing — auditing data integrity in public while not retrieving the entire data. Therefore, a way to expeditiously scale back the numerous burdens to existing users introduced by user revocation, and still enable a public champion to see the integrity of shared information while not downloading the complete information from the cloud, is a difficult task.

In this paper, we tend to propose Panda, a unique public auditing mechanism for the integrity of shared knowledge with efficient user revocation within the cloud. In our mechanism, by utilizing the thought of proxy re-signatures, once a user within the cluster is revoked, the cloud is in a position to resign the blocks, that were signed by the revoked user, with a re-signing key [Figure-2].As a result, the potency of user revocation may be considerably improved, and computation and communication resources of existing users may be simply saved. Meanwhile, the cloud, who isn't within the same sure domain with every user, is merely ready to convert a signature of the revoked user into a signature of associate in nursing existing user on identical block, however it cannot sign quirky blocks on behalf of either the revoked user or associate in nursing existing user.

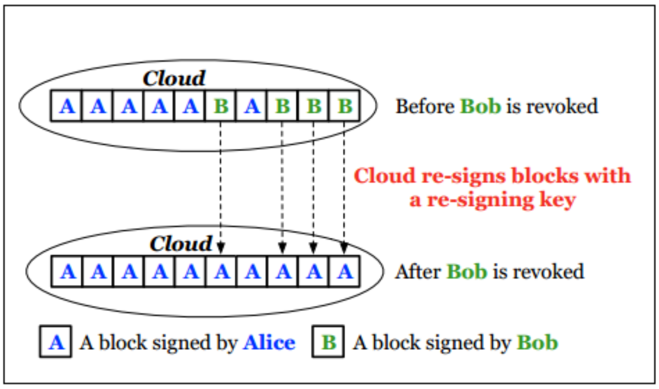


Fig.2. When Bob is revoked, the cloud re-signs the blocks that were previously signed by Bob with a re-signing key.

By planning new proxy resignature themes with nice properties, that ancient proxy resignatures do not have, our mechanism is oftenready to check the integrity of shared knowledge while not retrieving the entire knowledge from the cloud. Additionally, by taking benefits of Shamir Secret Sharing, we will additionally extend our mechanism into the multi-proxy model to reduce the prospect of the misuse on re-signing keys within the cloud and improve the responsibility of the whole mechanism.

II. RELATED WORK

A. Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing

Cloud Computing has been envisioned as the next-generation architecture of IT Enterprise. It moves the application software and databases to the centralized large data centers, where the management of the data and services may not be fully trustworthy. This unique paradigm brings about many new security challenges, which have not been well understood. This work studies the problem of ensuring the integrity of data storage in Cloud Computing. In particular, we consider the task of allowing a third party auditor (TPA), on behalf of the cloud client, to verify the integrity of the dynamic data stored in the cloud. The introduction of TPA eliminates the involvement of client through the auditing of whether his data stored in the cloud is indeed intact, which can be important in achieving economies of scale for Cloud Computing. The support for data dynamics via the most general forms of data operation, such as block modification, insertion and deletion, is also a significant step toward practicality, since services in Cloud Computing are not limited to archive or backup data only. While prior works on ensuring remote data integrity often lacks the support of either public verifiability or dynamic data operations, this paper achieves both. We first identify the difficulties and potential security problems of direct extensions with fully dynamic data updates from prior works and then show how to construct an elegant verification scheme for seamless integration of these two salient features in our protocol design. In particular, to achieve efficient data dynamics, we improve the Proof of Retrievability model by manipulating the classic Merkle Hash Tree (MHT) construction for block tag authentication. Extensive security and performance analysis show that the proposed scheme is highly efficient and provably secure

B. Provable Data Possession at Untrusted Stores

We introduce a model for provable data possession (PDP) that allows a client that has stored data at an untrusted server to verify that the server possesses the original data without retrieving it. The model generates probabilistic proofs of possession by sampling random sets of blocks from the server, which drastically reduces I/O costs. The client maintains a constant amount of metadata to verify the proof. The challenge/response protocol transmits a small, constant amount of data, which minimizes network communication. Thus, the PDP model for remote data checking supports large data sets in widely-distributed storage systems. We present two provably-secure PDP schemes that are more efficient than previous solutions, even when compared with schemes that achieve weaker guarantees. In particular, the overhead at the server is low (or even constant), as opposed to linear in the size of the data. Experiments using our implementation verify the practicality of PDP and reveal that the performance of PDP is bounded by disk I/O and not by cryptographic computation.

C. Privacy-Preserving Public Auditing for Secure Cloud Storage

Using Cloud Storage, users can remotely store their data and enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources, without the burden of local data storage and maintenance. However, the fact that users no longer have physical possession of the outsourced data makes the data integrity protection in Cloud Computing a formidable task, especially for users with constrained computing resources. Moreover, users should be able to just use the cloud storage as if it is local, without worrying about the need to verify its integrity. Thus, enabling public auditability for cloud storage is of critical importance so that users can resort to a third party auditor (TPA) to check the integrity of outsourced data and be worry-free. To securely introduce an effective TPA, the auditing process should bring in no new vulnerabilities towards user data privacy, and introduce no additional online burden to user. In this paper, we propose a secure cloud storage system supporting privacy-preserving public auditing. We further extend our result to enable the TPA to perform audits for multiple users simultaneously and efficiently. Extensive security and performance analysis show the proposed schemes are provably secure and highly efficient. Our preliminary experiment conducted on Amazon EC2 instance further demonstrates the fast performance of the design.

D. Proxy R e-Signatures : New Definitions, Algorithms, and Applications

In 1998, Blaze , B leumer, and Strauss (BB S) proposed *proxy re-signatures*, in which a semitrusted proxy acts as a *translator* between Alice and Bob. To translate, the proxy converts a signature from Alice into a signature from Bob on the same message. The proxy, however, does not learn any signing key and cannot sign arbitrary messages on behalf of either Alice or Bob.Since the BBS proposal, the proxy re-signature primitive has been largely ignored, but we show that it is a very useful tool for sharing web certificates, forming weak group signatures, and authenticating a network path. We begin our results by formalizing the definition of security for a proxy re-signature.

We next substantiate the need for improved schemes by pointing out certain weaknesses of theoriginal BBS proxy re-signature scheme which make it unfit for most practical applications. We then present two secure proxy re-signature schemes based on bilinear maps. Our first scheme relies on the Computational Diffie-Hellman (CDH) assumption; here the proxy cantranslate from Alice to Bob and vice-versa. Our second scheme relies on the CDH and 2-Discrete Logarithm (2-DL) assumptions and achieves a stronger security guarantee – the proxyis only able to translate in one direction. Constructing such a scheme has been an open problem since proposed by BBS in 1998. Furthermore in this second scheme, even if the delegator and the proxy collude, they cannot sign on behalf of the delegates. Both schemes are efficient and secure in the random oracle model.

E. An Efficient Auditing Protocol for Secure Data Storage in Cloud Computing

Cloud Computing is a type of distributed computing whereby resources and applications are shared over the internet. These applications are stored in one location and can be accessed in different location by any authorized users where the user does not need any infrastructure. In cloud storage, while outsourcing trust worthiness of the data is a scary task in cloud. To ensure the integrity of dynamic data stored in the cloud, external Third Party Auditor (TPA) is acquainted in a cloud infrastructure. For enabling public auditing in cloud data storage security, users can resort to an external auditor to check integrity of an outsourced data. The third party auditor (TPA) should met the following fundamental requirements: 1) TPA should be able to efficiently audit the cloud data without revealing the original data, and it should not add burden to the cloud user; 2) Auditing process should not bring no new vulnerabilities towards the user data. 3) Integrity of the data is protected against TPA by invoking some cryptographic techniques to ensure the storage correctness in cloud. In particular, this scheme achieves batch auditing where multiple delegated auditing tasks from different users, can be performed by the TPA and further enables TPA to perform data dynamics operations. Thus, the performance analysis depicts that the proposed schemes are more sheltered and highly competent.

F. Dynamic Audit Services for Outsourced Storages in Clouds

In this paper, we propose a dynamic audit service for verifying the integrity of an untrusted and outsourced storage. Our audit service is constructed based on the techniques, fragment structure, random sampling and index-hash table, supporting provable updates to outsourced data and timely anomaly detection. In addition, we propose a method based on probabilistic query and periodic verification for improving the performance of audit services. Our experimental results not only validate the effectiveness of our approaches, but also show our audit system verifies the integrity with lower computation overhead and requiring less extra storage for audit metadata.

III. EXISTING SYSTEM

Since shared data is outsourced to the cloud and users no longer store it on local devices, a straightforward method to re-compute these signatures during user repudiation (as shown in Figure. 1) is to ask an existing user (i.e., Alice) to first download the blocks previously signed by the repudiated user (i.e., Bob), verify the correctness of these blocks, then re-sign these blocks, and finally upload the new signatures to the cloud. However, this straightforward method may cost the existing user a huge amount of communication and computation resources by downloading and verifying blocks, and by re-computing and uploading signatures, especially when the number of re-signed blocks is quite large or the membership of the group is frequently changing. To make this matter even worse, existing users may access their data sharing services provided by the cloud with resource limited devices, such as mobile phones, which further prevents existing users from maintaining the correctness of shared data efficiently during user repudiation. Clearly, if the cloud could possess each user’s private key, it can easily finish the re-signing task for existing users without asking them to download and re-sign blocks. However, since the cloud is not in the same trusted domain with each user in the group, outsourcing every user’s private key to the cloud would introduce significant security issues. Another important problem we need to consider is that the re-computation of any signature during user repudiation should not affect the most attractive property of diverge auditing — auditing data integrity publicly without retrieving the entire data. Therefore, how to efficiently reduce the significant burden to existing users introduced by user repudiation, and still allow a public verifier to check the integrity of shared data without downloading the entire data from the cloud, is a challenging task.

IV. SYSTEM DESIGN

Block diagram includes three entities: the cloud, the public verifier, and two or many 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 or a third-party auditor (TPA) who can provide verification services on data integrity, aims to check the integrity of shared data via a challenge-and response protocol with the cloud. In the group, there is one original user and a number 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.

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 users 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 repudiate this user. Generally, as the creator of shared data, the original user acts as the group manager and is able to repudiate users on behalf of the group. Once a user is repudiated, the signatures computed by this repudiated user become invalid to the group, and the blocks that were previously signed by this repudiated user should be re-signed by an existing user’s private key, so that the correctness of the entire data can still be verified with the public keys of existing users only.

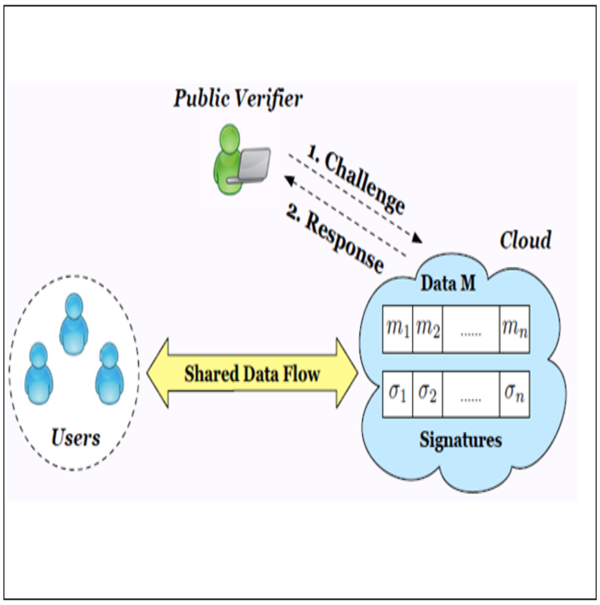


Fig 3 Architecture diagram of PANDA

V. CONCLUSION

In this paper, we did analysis of proposed work and have a tendency to propose a completely unique public auditing mechanism for the integrity of shared data with economical user revocation in mind. By utilizing the thought of proxy re-signatures, we have a tendency to enable the cloud to re-sign blocks on behalf of existing users throughout user revocation, in order that existing users don't have to right transfer and re-sign blocks by themselves. Additionally, a public verifier is often able to audit the integrity of shared knowledge while not retrieving the complete data from the cloud, although some a part of shared knowledge has been re-signed by the cloud. Moreover, our mechanism is in a position to support batch auditing by verifying multiple auditing tasks at the same time. We proposed a new public auditing mechanism for shared data with efficient user revocation in the cloud. When 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.

VI. .REFERENCES

[1]. B. Wang, B. Li, Member, H. Li, —Panda: Public Auditing for Shared Data with Efficient User Revocation in the Cloud — in the Proceedings of IEEE INFOCOM 2014, 2014

[2]. B. Wang, B. Li, and H. Li, —Public Auditing for Shared Data with Efficient User Revoation in the Cloud,‖ in the Proceedings of IEEE INFOCOM 2013, 2013, pp. 2904–2912.

[3]. M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, an M. Zaharia, —A View of Cloud Computing,‖ Communications of the ACM, vol. 53, no. 4, pp. 50–58, Apirl 2010.

[4]. G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, —Provable Data Possession at Untrusted Stores,‖ in the Proceedings of ACM CCS 2007, 2007, pp.598–610.

[5]. H. Shacham and B. Waters, —Compact Proofs of Retrievability,‖ in the Proceedings of ASIACRYPT 2008. S pringe Verlag,2008,pp. 90–107.

[6]. C. Wang, Q. Wang, K. Ren, and W. Lou, —Ensuring Data Storage Security in Cloud Computing,‖ in the Proceedings of ACM/IEEEIWQoS 2009, 2009, pp. 1 –9.

[7]. C. Wang, Q. Wang, K. Ren, and W. Lou, —Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing,‖ in the Proceedings of IEEE INFOCOM 2010, 2010, pp.525–533.

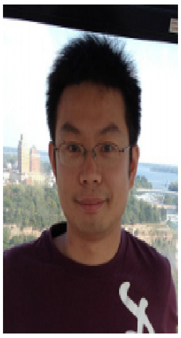
[8]. H. Wang, —Proxy Provable Data Possession in Public Clouds,‖ IEEE Transactions on Services Computing, accepted.

[9]. B. Wang, B. Li, and H. Li, —Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud,‖ in the Proceedings of IEEE Cloud 2012, 2012, pp. 295–302.

[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.

[11]. B.Wang, B. Li, and H. Li, —Certificateless Public Auditing for Data Integrity in the Cloud,‖ in Proceedings of IEEE CNS 2013, 2013, pp. 276–284.

[12] 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.



Boyang Wang is a Ph.D. student from the School of Telecommunications Engineering, Xidian University, Xi'an, China. He was a visiting Ph.D. student at the Department of Electrical and Computer Engineering, University of Toronto, from Sep. 2010 to Aug. 2012. He obtained his B.S. in information security from Xidian University in 2007. His current research interests focus on security and privacy issues in cloud computing, big data, and applied cryptography. He is a student member of IEEE.

Baochun Li is a Professor at the Department of Electrical and Computer Engineering at the University of Toronto, and holds the Bell University Laboratories Endowed Chair in Computer Engineering. His research interests include large-scale multimedia systems, cloud computing, peer-to-peer networks, applications of network coding, and wireless networks. He is a member of ACM and a senior member of IEEE.

Hui Li is a Professor at the School of Telecommunications Engineering, Xidian University, Xi'an, China. He received B.Sc. degree from Fudan University in 1990, M.Sc. and Ph.D. degrees from Xidian University in 1993 and 1998. In 2009, he was with Department of ECE, University of Waterloo as a visiting scholar. His research interests are in the areas of cryptography, security of cloud computing, wireless network security, information theory. He is the co-author of two books. He served as TPC co-chair of ISPEC 2009 and IAS 2009, general co-chair of E-Forensic 2010, ProvSec 2011 and ISC 2011.