International Journal of Engineering Research and General Science Volume 2, Issue 6, October-November, 2014
ISSN 2091-2730

Homomorphic Authenticable Ring Signature (HARS) mechanism for Public Auditing on Shared Data in the cloud (Oruta)

Ms.Sonam M. Kamble[#1], Prof.A.C.Lomte[*2]

[#] *Studen , Department Of Computer Engineering, University of Pune, BSIOTR Wagholi, Pune, Maharashtra, India*

[*] *Professor, Department Of Computer Engineering, University of Pune, BSIOTR Wagholi, Pune, Maharashtra, India*

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, privacy-preserving, shared data, Digital Signature, cloud computing,

INTRODUCTION

With cloud computing and storage, users are able to access and to share resources offered by cloud service providers at a lower marginal cost. It is routine for users to leverage cloud storage services to share data with others in a group, as data sharing becomes standard feature in most cloud storage offerings, including Dropbox, iCloud and Google Drive. The integrity of data in cloud storage, however, is subject to skepticism and scrutiny, as data stored in the cloud can easily be lost or corrupted due to the inevitable hardware/software failures and human errors.

The traditional approach for checking data correctness is to retrieve the entire data from the cloud, and then verify data integrity by checking the correctness of signatures (e.g., RSA) or hash values (e.g., MD5) of the entire data. Certainly, this conventional approach able to successfully check the correctness of cloud data. However, the efficiency of using this traditional approach on cloud data is in doubt.

The main reason is that the size of cloud data is large in general. Downloading the entire cloud data to verify data integrity will cost or even waste user's amounts of computation and communication resources, especially when data have been corrupted in the cloud. Recently, many mechanisms have been proposed to allow not only a data owner itself but also a public verifier to efficiently perform integrity checking without downloading the entire data from the cloud, which is referred to as public auditing.

In these mechanisms, data is divided into many small blocks, where each block is independently signed by the owner; and a random combination of all the blocks instead of the whole data is retrieved during integrity checking .A public verifier could be a data user (e.g. researcher) who would like to utilize the owners data via the cloud or a third-party auditor (TPA) who can provide expert integrity checking services. Existing public auditing mechanisms can actually be extended to verify shared data integrity and data freshness. However, a new significant privacy issue introduced in the case of shared data with the use of existing mechanisms is the leakage of identity privacy to public verifiers .To protect the confidential information, it is essential and critical top reserve identity privacy from public verifiers during public auditing.

In our model, privacy is accomplished by allowing the parties to upload their data in multi clouds and data is split into multiple parts so it gives more protection.The critical reasons due to which our above system is beneficial as:

1. Current working scenario involves paper based work for Data analysis and verification.
2. Data Storage is one way to mitigate the privacy concern.
3. Unauthorized users can leak or misuse the data, this problem still remains due to the paper based work.

These are the above reasons which compel us to propose Oruta, a novel privacy preserving public auditing mechanism. More specifically, we utilize ring signatures to construct homomorphic authenticators in Oruta, so that a public verifier is able to verify the

integrity of shared data without retrieving the entire data while the identity of the signer on each block in shared data is kept private from the public verifier. In addition, extend this mechanism to support batch auditing, which can perform multiple auditing tasks simultaneously and improve the efficiency of verification for multiple auditing tasks. Mean while, Oruta is compatible with random masking; *Oruta* stands for *One Ring to Rule Them All*.

For the first time data inserting the Encryption service generate encryption key and this key is stored separately on Key Storage area, and encrypted data is stored on the cloud storage area.In decryption process when the user request for the data, then key and data are collected at the Decryption service but the service will not immediately decrypt the data, until and unless user insert the OTP sent on his mail. When user will enter this OTP correctly then the data is decrypted by Decryption service and data is provided to the user. Some researchers have suggested that user data stored on a Service- provider's equipment must be encrypted. Encrypting data prior to storage is a common method of data protection, and service providers may be able to build firewalls to ensure that the decryption keys associated with encrypted user data are not disclosed to outsiders. However, if the decryption key and the encrypted data are held by the same service provider, it raises the possibility that high-level administrators within the service provider would have access to both the decryption key and the encrypted data, thus presenting a risk for the unauthorized disclosure of the user data.Existing methods for protecting data stored in cloud environment are user authentication, building secure channel for transmission of data. For this procedure they use various Cryptographic as well as Security based algorithm such as AES (Advance Encryption Standard), DES (Data Encryption Algorithm), Triple DES, RSA algorithm with digital signature.

We know that there is a method to build trusted computing environment for cloud computing system by integrating a trusted computing platform into the security of cloud computing system.System in which a cloud computing system is combined with trusted computing platform with trusted platform module that consists of some important security services, including authentication, confidentiality and integrity, are provided in cloud computing system. This tends to provide extra level of security. In their proposed model the Encryption-Decryption service will only the encrypt the data and this encrypted data is send to cloud storage and then original data is deleted from Encryption-Decryption service.The need of our system to overcome the flaws and include some of the above advantages has led to the implementation of our proposed algorithm.

LITERATURE SURVEY

For some years, tools for defending against hackers have been in the form of software to be installed on each device being protected or appliances deployed on-premise. However, to be effective, such protection needs to be constantly updated. Common methods for ensuringsecurity of data in cloud consist of data encryption (cryptographic process) before storage, authentication process before storage or retrieval and constructing secure channels for data transmission. The protection methods find their routes in cryptographic algorithms and digital signature techniques.

The cryptographic algorithms are classified into two categories: symmetric and asymmetric algorithms. Symmetric algorithm uses a single key known as secret key both for encryption and decryption process whereas asymmetric algorithm uses two keys; one is the public key made available publically and the other one is the private key, which is kept secret used to decrypt the data. Breaking the private key is rarely possible even if the corresponding public key is known well in advance. Examples of symmetric algorithm comprise of Data encryption standard (DES), International data encryption algorithm (IDEA), advanced encryption standard (AES) on the other hand asymmetric key algorithm include RSA algorithm.Asymmetric algorithms are best suited for real world use and provides undeniable advantages in terms of functionality whereas symmetric algorithms is ideally suited for security applications like remote authentication for restricted websites which do not require full-fledged asymmetric set up.The use of passwords for authentication process is popular among the users but the transmission of messages containing password may be vulnerable to illegal recording by the hackers hence posing a security breach in the system. Some more advanced authentication techniques may employ the concept of single-usage-password where the system may generate challenge token expecting the user to respond with an encrypted message using his secret key which converts the password to some derived value enabling.

While using the cryptographic techniques for ensuring data security care should be taken for storing encryption and decryption keys. Rigorous methods should be adopted to prevent insiders and privileged user from gaining access to the encrypted data and decryption key simultaneously. Thus, the importance of SLAs is recognized in this context. The policies responsible for user data protection must be clearly mentioned in the provider's contract. After reviewing the data security requirements following recommendations have been included in multiparty SLA suggested at the end to ensure data security in cloud:

1. Encrypted data and decryption key must not be stored at the same place

2. Access control techniques should be applicable for malicious insiders and privileged users

3. Independent audits must be conducted to access the effectiveness of techniques employed for data storage

4. Service providers must abide the ethics and legal laws and should be responsible for discrepancies if any

5. Backup and reset methods against system crash and failures.

In many applications, it is desirable to work with signatures that are both short and yet where many messages from different signers are verified very quickly. RSA signatures satisfy the latter condition, but are generally thousands of bits in length. Recent developments in pairing based cryptography produced a number of short signatures which provide equivalent security in a fraction of the space. Unfortunately, verifying these signatures is computationally intensive due to the expensive pairing operation. In an attempt to simultaneously achieve short and fast signatures, it was proved how to batch verify two pairing-based schemes so that the total number of pairings was independent of the number of signatures to verify. On the theoretical side, we introduce new batch verifiers for a wide variety of regular, identity based, group, ring and aggregate signature schemes. Our goal is to test whether batching is practical; that is, whether the benefits of removing pairings significantly outweigh the cost of the additional operations required for batching, such as group membership testing, randomness generation, and additional modular exponentiations and multiplications.

MATERIAL AND METHODS

Users are able to access and to share resources offered by cloud service providers at a lower marginal cost. It is routine for users to leverage cloud storage services to share data with others in a group, as data sharing becomes standard feature in most cloud storage offerings, including Dropbox, iCloud and Google Drive. The integrity of data in cloud storage, however, is subject to skepticism and scrutiny, as data stored in the cloud can easily be lost or Corrupted due to the inevitable hardware/software failures and human errors. The traditional approach for checking data correctness is to retrieve the entire data from the cloud, and then verify data integrity by checking the correctness of signatures (e.g., RSA) or hash values (e.g., MD5) of the entire data. Certainly, this conventional approach ableto successfully check the correctness of cloud data. However, the efficiency of using this traditional approach on cloud data is in doubt.

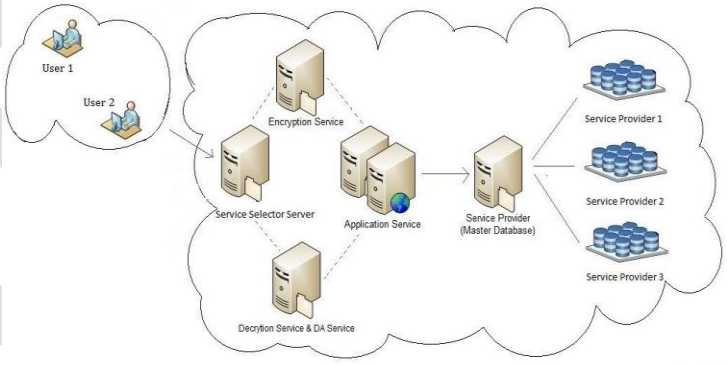

Figure 1: System Arichitecture

The main reason is that the size of cloud data is large in general. Downloading the entire cloud data to verify data integrity will cost or even waste users' amounts of computation and communication resources, especially when data have been corrupted in the cloud. Recently, many mechanisms have been proposed to allow not only a data owner itself but also a public verifier to efficiently perform integrity checking without downloading the entire data from the cloud, which is referred to as public auditing . In these mechanisms, data is divided into many small blocks, where each block is independently signed by the owner; and a random combination of all the blocks instead of the whole data is retrieved during integrity checking .A public verifier could be a data user (e.g. researcher) who would like to utilize the owners data via the cloud or a third-party auditor (TPA) who can provide expert integrity checking services. Existing public auditing mechanisms can actually be extended to verify shared data integrity and data freshness. However, a new significant privacy issue introduced in the case of shared data with the use of existing mechanisms is the leakage of identity privacy to public verifiers .To protect the confidential information, it is essential and critical top reserve identity privacy from public verifiers during public auditing.

To solve the above privacy issue on shared data, we propose Oruta, a novel privacy preserving public auditing mechanism. More specifically, we utilize ring signatures to construct homomorphic authenticators in Oruta, so that a public verifier is able to verify the integrity of shared data without retrieving the entire data while the identity of the signer on each block in shared data is kept private from the public verifier. In addition, extend this mechanism to support batch auditing, which can perform multiple auditing tasks simultaneously and improve the efficiency of verification for multiple auditing tasks.

In the idea of cloud computing the user of cloud outsources its data on to the cloud, and then the third party auditor is going to check authorization of that user to access the cloud.In cloud if it is found that the unauthorized user is trying to access data of any other authorized user then the third party comes in picture, the third party auditor gives the notification to the authorized user that some unauthorized user is trying to access its private data. The concept of cloud computing represents a shift in thought, in those end users need not know the details of a specific technology. The service is fully managed by the provider. This on demand service can be provided at cloud service providers are making a substantial effort to secure their systems, in order to minimize the threats ofinsider attacks, and reinforce the confidence of customers. In the cloud scenario if third party auditor itself get hacked then the authorized will not receive any notification of unauthorized access of its data. So in the propose method the service will eliminates the third party auditor.

DESIGN PROCESS

Some of the features those are included in our design feature are as follows:

1. Ring Signatures:

The concept of ring signatures is first proposed by Rivest et al. in 2001. With ring signatures, a verifier is convinced that a signature is computed using one of group members private keys, but the verifier is not able to determine which one. This property can be used to preserve the identity of the signer from a verifier. The ring signature scheme introduced by Boneh et al. (Referred to as BGLS in this paper) is constructed on bilinear maps.We will extend this ring signature scheme to construct our public auditing mechanism.

2. Integrity Threats:

Two kinds of threats related to the integrity of shared data are possible. First, an adversary may try to corrupt the integrity of shared data and prevent users from using data correctly. Second, the cloud service provider may inadvertently corrupt (or even remove) data in its storage due to hardware failures and human errors. Making matters worse, in order to avoid jeopardizing its reputation, the cloud server provider may be reluctant to inform users about such corruption of data.

3. Privacy Threats:

The identity of the signer on each block in shared data is private and confidential to the group. During the process of auditing, a semi-trusted TPA, who is only responsible for auditing the integrity of shared data, may try to reveal the identity of the signer on each block in shared data based on verification information. Once the TPA reveals the identity of the signer on each block, it can easily distinguish a high-value target (a particular user in the group or a special block in shared data).

OBSERVATIONS

The obserations of our proposed system led to the happening of the results mentioned below:

1. Encrypted data and key are stored separately on different storage media.

2. Before decrypting the data the user have to enter OTP which is sent on his mail and combination of OTP, key and encrypted data are used to generate original data.

3. For accessing the data the user is restricted in read only mode and for insert, modify and delete the notification is sent to admin.

4. After encryption or decryption the original data is deleted.

5. For securing the Account and Service Hijacking, we are eliminating the TPA. The work of TPA will be done by admin and our proposed system.

ACKNOWLEDGMENT

We would like to sincerely thank Prof.A.C.Lomte, our mentor (Professor, at Computr Department of BSIOTR, Wagholi, Pune), for her support and encouragement**.**

CONCLUSION`

The security aspect in cloud is major concern thus we have proposed novel system which can process the request in grouping or batch manner which can enhance performance and efficiency of data transfer/system.The algorithm clearly shows improvements to its predecessor in various fashions like security, transfer of data, scalability and other perspective.

REFERENCES

[1] M. Armbrust, A. Fox, R. Griffith, A. D.Joseph, R. H.Katz, A. Konwinski,G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A View of Cloud Computing, Communications of the ACM", vol. 53, no. 4, pp. 5058, April 2010

[2] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song," Provable Data Possession at Untrusted Stores, in Proc. ACM Conference on Computer and Communications Security (CCS)", 2007, pp. 598610.

[3] C.Wang, Q.Wang, K. Ren, andW. Lou, "Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing, in Proc. IEEE International Conference on Computer Communications (INFOCOM)", 2010, pp.525533.

[4] R. L. Rivest, A. Shamir, and Y. Tauman,"How to Leak a Secret, in Proc. International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT)". SpringerVerlag, 2001, pp. 552565.

[5] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and Verifiably Encrypted Signatures from Bilinear Maps, in Proc. International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT)". Springer-Verlag, 2003, pp. 416432.

[6] H. Shacham and B. Waters, "Compact Proofs of Retrievability,in Proc. International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT)". SpringerVerlag, 2008, pp. 90107.

[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] Y. Zhu, H. Wang, Z. Hu, G.-J. Ahn, H. Hu, and S. S.Yau, "Dynamic Audit Services for Integrity Verification of Outsourced Storage in Clouds, in Proc. ACM Symposium on Applied Computing (SAC)", 2011, pp.15501557