ISSN 2320 - 2602
Volume 5 No.4, April 2016
International Journal of Advances in Computer Science and Technology
Available Online at http://www.warse.org/IJACST/static/pdf/file/ijacst01542016.pdf

Privacy Enhancing using HLA scheme while data auditing in cloud

P.Ravinder Rao
Dept.of CSE,
Anurag Group of Instiitutions
Hyderabad,India
ravinderraocse@cvsr.ac.in

Amarnadh
Dept.of CSE,
Anurag Group of Instiitutions
Hyderabad,India
amarnadhcse@cvsr.ac.in

P.Raja Sekhar Reddy
Dept.of CSE,
Anurag Group of Instiitutions
Hyderabad,India
rajasekharreddycse@cvsr.ac.in

ABSTRACT
Cloud users can remotely store their data and access the on-demand quality applications and services from a shared pool of computing resources. Cloud computing provides an economical and efficient solution for sharing group resource among cloud users. It is one of the common places for data storage and shared across multiple users. With our proposed mechanism the identity of the signer is kept private from third party auditor who is still able to publicly verify the integrity of shared data without retrieving the entire file while preserving data and identity by using HLA scheme in the cloud.

Keywords—Cloud Server ,Third Party Auditor, Homomorphic Linear Authenticator.

1. INTRODUCTION

Cloud service providers manage an enterprise-class infrastructure that offers a scalable, secure and reliable environment for users, at a much lower marginal cost due to the sharing nature of resources. The major mis conceptions of using cloud security is still absolutely an issue. In fact, it's a growing issue. Arbor Networks 9th annual Worldwide Infrastructure Security Report illustrates this point very clearly with the largest reported DDoS attack in 2013 clocking in at 309 Gbps. As cloud computing becomes more popular, it will become the target of more malicious attacks. No single environment is safe and every infrastructure must be controlled with set policies in place. Heart bleed is a perfect example where a number of massive cloud organizations can be impacted by standardized security structure Dealing with data loss**.** Allowing users to get into the cloud is one thing. Accessing applications through a cloud model is a powerful way to allow end-users to work remotely. However, what happens when users start uploading files to the cloud? Healthcare is a great example where data loss can be extremely costly. A recent report from the Health Information Trust Alliance (HITRUST) really paints the picture around the ramifications of a data breach. Over the recent years, the numbers around healthcare data breaches can be quit sobering. Many organizations often times don't have a Data Loss Prevention (DLP) system plan in place. This means that a user, even non-maliciously, might post some information or upload a file which can contain sensitive information. It is routine for users to use cloud storage services to share data with others in a team, as data sharing becomes a standard feature in most cloud storage offerings. The integrity of data in cloud storage, however, is subject to uncertainty and inspection, as data stored in un-trusted cloud can easily be lost or corrupted, due to hardware failures and human errors [1]. To protect the integrity of cloud data, it is best to perform public auditing by introducing a third party auditor (TPA), who offers its auditing service with more powerful computation and communication abilities than regular users. The first provable data possession (PDP) mechanism [2] to perform public auditing is designed to check the correctness of data stored in un-trusted server, without retrieving the entire data. Moving a step forward, Wang et al. [3] is designed to construct a public auditing mechanism for cloud data, so that during public auditing, the content of private data belonging to a personal user is not disclosed to the third party auditor. We believe that sharing data among multiple users is perhaps one of the most engaging features that motivate cloud storage. A unique problem introduced during the process of public auditing for shared data in the cloud is how to preserve privacy from the TPA, because the identities of signers on shared data may indicate that a particular user in the group or a special block in shared data is a higher valuable target than others besides this problem . Several security systems for data sharing on un trusted servers have been proposed[4],[5],[6].in these approaches ,data owners store the encrypted data files in

un trusted storage and distributed the corresponding decryption keys only to authorized users. Thus unauthorized users as well as storage servers cannot learn the content of the data files because they do not have the knowledge of decryption keys.

The main contributions of this paper include:

1. We provide secure and privacy-preserving access control to users, which guarantees any member in a group to anonymously utilize the cloud resource.
2. We suggested a model of Third party public auditing protocol for privacy preserving.

2. RELATED WORK

In [5], files stored on the un-trusted server include two parts: file metadata and file data. The file metadata implies the access control information including a series of encrypted key blocks, each of which is encrypted under the public key of authorized users. Thus, the size of the file metadata is proportional to the number of authorized users. The user revocation in the scheme is an intractable issue especially for large-scale sharing, since the file metadata needs to be updated. In their extension version, the NNL construction [9] is used for efficient key revocation. However, when a new user joins the group, the private key of each user in an NNL system needs to be recomputed, which may limit the application for dynamic groups. Another concern is that the computation overhead of encryption linearly increases with the sharing scale.

Ateniese et al. [6] leveraged proxy re encryptions to secure distributed storage. Specifically, the data owner encrypts blocks of content with unique and symmetric content keys, which are further encrypted under a master public key. For access control, the server uses proxy cryptography to directly re encrypt the appropriate content key(s) from the master public key to a granted user's public key. Unfortunately, a collusion attack between the un-trusted server and any revoked malicious user can be launched, which enables them to learn the decryption keys of all the encrypted blocks.

Yu et al [7] presented a scalable and fine-grained data access control scheme in cloud computing based on the KPABE technique. The data owner uses a random key to encrypt a file, where the random key is further encrypted with a set of attributes using KP-ABE. Then, the group manager assigns an access structure and the corresponding secret key to authorized users, such that a user can only decrypt a cipher text if and only if the data file attributes satisfy the access structure. To achieve user revocation, the manager delegate's tasks of data file re encryption and user secret key update to cloud servers. However, the single owner manner may hinder the implementation of applications with the scenario, where any member in a group should be allowed to store and share data files with others. Lu et al. [8] proposed a secure provenance scheme, which is built upon group signatures and cipher text-policy attribute-based encryption techniques. Particularly, the system in their scheme is set with a single attribute. Each user obtains two keys after the registration: a group signature key and an attribute key. Thus, any user is able to encrypt a data file using attribute-based encryption and others in the group can decrypt the encrypted data using their attribute keys. Meanwhile, the user signs encrypted data with her group signature key for privacy preserving and traceability. However, user revocation is not supported in their scheme.

3. PRELIMINARIES

3.1 Bi-linear Maps

Let G1 and G2 be an additive cyclic group and a multiplicative cyclic group of the same prime order q, respectively [10].

Let e: G1 X G1 -> G2 denote a bilinear map constructed with the following properties:

1. Bilinear: For all a, b $\in Z_q^*$ and P, Q $\in$ G1, e(aP, bQ)=e(P,Q)^{ab}.
2. Non degenerate: $\exists$P such that e(P,P)$\neq$1.
3. Computable: There is an efficient algorithm to compute e(P,Q) for any P,Q $\in$G1.

3.2 Complexity Assumptions

Definition 1: q-strong Diffie-Hellman (q-SDH) Assumption [10]

Definition 2: Decision linear (DL) Assumption [10]

Definition 3: Weak Bilinear Diffie-Hellman Exponent (WBDHE) Assumption [11].

Definition 4: ((t,n)-general Diffie-Hellman Exponent (GDHE) Assumption [12]).

3.3 MAC-based Solution

Use of MAC to authenticate the data is a trivial way to upload the data blocks with their MACs to the server, and sends the corresponding secret key *sk* to the TPA. Later, the TPA can randomly retrieve blocks with their MACs and check the correctness via *sk*. Apart from the high (linear in the sampled data size) communication and computation complexities, the TPA requires the knowledge of the data blocks for verification. To circumvent the requirement of the data in TPA verification, one may restrict the verification to just consist of equality checking. The idea is as follows. Before data outsourcing, the cloud user chooses s random message authentication code keys $\{sk\tau\}_{1\leq\tau\leq s}$, pre-computes *s* (deterministic) MACs, $\{MAC_{sk\tau}(F)\}_{1\leq\tau\leq s}$ for the whole data file F, and publishes these verification metadata (the keys and the MACs) to TPA. The TPA can reveal a secret key sk_τ to the cloud server and ask for a fresh keyed MAC for comparison in each audit. This is privacy preserving as long as it is impossible to recover F in full given $MAC_{sk\tau}$ (F) and sk_τ. However, it suffers from the following severe drawbacks:

1) The number of times a particular data file can be audited is limited by the number of secret keys that must be fixed a priori. Once all possible secret keys are exhausted, the user then has to retrieve data in full to re-compute and re-publish new MACs to TPA.

2) The TPA also has to maintain and update state between audits, i.e., keep track on the revealed MAC keys. Considering the potentially large number of audit delegations from multiple users, maintaining such states for TPA can be difficult and error prone.

3) It can only support static data, and cannot efficiently deal with dynamic data at all. However, supporting data dynamics is also of critical importance for cloud storage systems. For the reason of brevity and clarity

4. SYSTEM MODEL

As illustrated in the figure this paper involves three parties: the cloud server, User and Public Verifier.

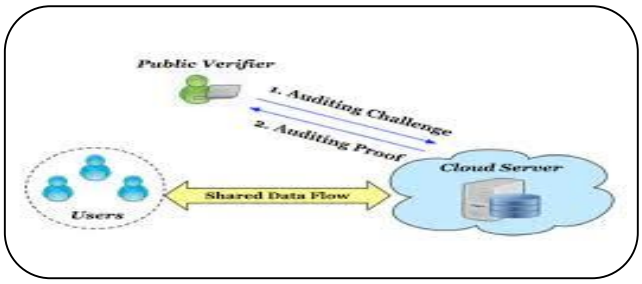

Figure.1. public Audit process in cloud for storage correctness[14]

The cloud is operated by CSP's and provide huge storage services however cloud is not fully trusted by users since CSP's are very likely to be outside of cloud users trusted domain. The group manager is responsible for user registration, user revocation.

The group members are set of registered users that will store their private data into the cloud server and share them with others in the group.

The Third party auditor (TPA) is responsible for auditing the stored data to check its integrity, in this user wishes to check the integrity of shared data, he will send an auditing request to the TPA, after receiving the request, TPA generated the auditing message, to the clouds server and retrieves an auditing proof of shared data from the cloud server, then the TPA verifies the auditing proof, finally the TPA sends an auditing report to the user based on the result of the verification.

5. THE PROPOSED METHOD

5.1. Overview

To implement the effective public auditing in the cloud rather than MAC based solution as a better option HLA scheme can be used for effective check of storage correctness of data in the cloud ,we expect this technique allows data users can verify the storage correctness.

5.2 HLA-based Solution:

To effectively support public auditability without having to retrieve the data blocks themselves, the HLA technique [10], [11], [12] can be used. HLAs, like MACs, are also some un-forgeable verification metadata that authenticate the integrity of a data block. The difference is that HLAs can be aggregated. It is possible to compute an aggregated HLA which authenticates a linear combination of the individual data blocks. At a high level, an HLA-based proof of storage system works as follow. The user still authenticates each element of F = ($m_1, \cdots, m_n$) by a set of HLAs Φ. The cloud server stores {*F, Φ*}. The TPA verifies the cloud storage by sending a random set of challenge {vi}. (More precisely, F, Φ and {vi} are all vectors, so {v_i} is an ordered set or {*i,* v_i} should be sent). The cloud server then returns $\mu = \sum_i v_i \cdot m_i$ and an aggregated authenticator σ (both are computed from *F, Φ* and {v_i}) that is supposed to authenticate μ. Though allowing efficient data auditing and consuming only constant bandwidth, the direct adoption of these HLA-based techniques is still not suitable for our purposes. This is because the linear combination of blocks, $\mu = \sum_i v_i \cdot m_i$, may potentially reveal user data information to TPA, and violates the privacy preserving guarantee. Specifically, if an enough number of the linear combinations of the same blocks are collected, the TPA can simply derive the user's data content by solving a system of linear equations.

5.3 Privacy-preserving public auditing system

Once if the group user needs to check correctness of the data stored in the cloud, the public auditing mechanism can be effectively implemented to achieve this task. Various public auditing protocols were implemented to do these tasks. To achieve privacy preserving we suggest the model which uses homo-morphic linear authenticator. This scheme follow the implementation process with the

Setup Phase: The cloud user runs Key Gen to generate the public and secret parameters. Specifically, the user chooses a random signing key pair (*spk, ssk*), a random $x \leftarrow Z_p$, a random element $u \leftarrow G_1$, and computes $v \leftarrow g^x$. The secret parameter is *sk* = (*x, ssk*) and the public parameters are *pk* = (*spk, v, g, u, e(u, v)*).

Given a data file $F = (m_1, \ldots, m_n)$, the user runs SigGen to compute authenticator σ_I for each block m_i : $\sigma_i \leftarrow (H(W_i) \cdot u^{m_i})^x \in G1$. Here W_i = *name||i* and *name* is chosen by the user uniformly at random from from Z_p as the identifier of file F. Denote the set of authenticators by $\Phi = \{\sigma i\}_{1 \le i \le n}$. The last part of the sigGen is for ensuring the integrity of the unique file identifier *name.*

For data storage correctness. Specifically, the server chooses a random element $r \leftarrow Z_p$, and calculates $R = e(u, v)^r \in G_T$. Let μ ′ denote the linear combination of sampled blocks specified in chal: $\mu' = \sum_{i \in I} v_i m_i$. To blind μ ′ with r, the server computes: $\mu = r + \gamma\mu' \bmod p$, where $\gamma = h(R) \in Z_p$. Meanwhile, the server also calculates an aggregated

authenticator $\sigma = \pi_{i\in I} \sigma_i^{vi} \in G_1$. It then sends {μ, σ, R} as the response proof of storage correctness to the TPA.

The sequence of operations carried out at TPA

1. Retrieve file tag *t*, verify its Signature, and quit if fail;
2. Generate a random challenge v_i m_i, and also $\sigma=\pi_{i\in I} \sigma_i^{vi}$
3. {μ, σ ,R}- >Storage correctness proof
4. Compute γ = h(R), and then verify {μ, σ,R}.

Operations in the middle

{(i,v_i)}i∈ I - challenge request *chal*

The sequence of operations carried out at cloud server

1. Compute $\mu' = \sum_{i\in I}$.
   This operation is carried after executing step 1and 2 operations of TPA
2. Randomly pick r ← Zp, and h(R);
3. Compute $\mu = r + \gamma \mu'$ mod p

Then CS sends {μ, σ ,R} Storage correctness proof to TPA
Finally the last step of operation in TPA gets executed for verification of storage correctness proof sent by cloud server..

Public Auditing process

Audit Phase: The TPA first retrieves the file tag *t.* With respect to the mechanism we described in the setup phase, the TPA verifies the signature $SSig_{ssk}$ *with Spk*, and quits by emitting FALSE if the verification fails .otherwise TPA recovers *name.* To generate the challenge message for the audit "chal", the TPA picks a random c-element subset I = {s_1, . . . , s_c} of set [1, *n*]. For each element i ∈ I, the TPA also chooses a random value v_i (of bit length that can be shorter than |p|, as explained in [11]). The message "chal" specifies the positions of the blocks that are required to be checked. The TPA sends chal = $\{(i, v_i)\}_{i\in I}$ to the server.

Upon receiving challenge chal = $\{(i, v_i)\}_{i\in I}$, the server runs Gen Proof to generate a response proof and new users can directly decrypt files stored in the cloud before their participation and by using homo-morphic linear authenticator TPA would not learn any knowledge about the data content stored on the cloud server during the efficient auditing process, which not only eliminates the burden of cloud user from the tedious and possibly expensive auditing task, but also alleviates the users' fear of their outsourced data leakage.

6. TEST ANALYSIS

To evaluate the performance of the cloud, testing its computation cost to respond different operations issued by client auditing requests. It can be assumed that the computation cost of the cloud is acceptable. In addition, it is worth noting that the computation cost is independent with the size of the requested audit for file to access and verify operations.

7. CONCLUSION

In the current paper, we suggest a privacy preserving public auditing method. The cloud user's wishes to check correctness of the data stored in the cloud, the public auditing mechanism can be effectively implemented to achieve this task. The HLA based scheme will be having fewer burdens on the server there by reducing the traffic and computation complexity by aggregating the HLAs.

ACKNOWLEDGEMENT

This work was carried out with the help of reference papers , I thank all the authors of reference papers without which this work would not be carried out.

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. 50–58, Apirl 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. 598–610.

[3] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing," in *Proc. IEEE International Conference on Computer Communications (INFOCOM)*, 2010, pp. 525–533.

[4] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K. Fu,"Plutus: Scalable Secure File Sharing on Untrusted Storage," Proc. USENIX Conf. File and Storage Technologies, pp. 29-42, 2003.

[5] E. Goh, H. Shacham, N. Modadugu, and D. Boneh, "Sirius: Securing Remote Untrusted Storage," Proc. Network and Distributed Systems Security Symp. (NDSS), pp. 131-145, 2003.

[6] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage," Proc. Network and Distributed Systems Security Symp. (NDSS), pp. 29-43, 2005.

[7] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving Secure, Scalable, and Fine-Grained Data Access Control in Cloud Computing,"Proc. IEEE INFOCOM, pp. 534-542, 2010.

[8] R. Lu, X. Lin, X. Liang, and X. Shen, "Secure Provenance: The Essential of Bread and Butter of Data Forensics in Cloud Computing," Proc. ACM Symp. Information, Computer and Comm. Security, pp. 282-292, 2010.

[9] D. Naor, M. Naor, and J.B. Lotspiech, "Revocation and Tracing Schemes for Stateless Receivers," Proc. Ann. Int'l Cryptology Conf. Advances in Cryptology (CRYPTO), pp. 41-62, 2001.

[10] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrustedstores," in *Proc. of CCS'07*, Alexandria, VA, October2007, pp. 598–609.

[11] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public Verifiability and data dynamics for storage security in cloud computing," in *Proc. of ESORICS'09, volume 5789 of LNCS*.Springer-Verlag, Sep. 2009, pp. 355–370.

[12] H. Shacham and B. Waters, "Compact proofs of retrievability," in *Proc. of Asiacrypt 2008*, vol. 5350, Dec 2008, pp. 90–107.

[13] B. Sheng and Q. Li, "Verifiable Privacy-Preserving Range Query in Two-Tiered Sensor Networks," Proc. IEEE INFOCOM, pp. 46-50, 2008.

[14] Boyang Wang, Baochun Li, Member, IEEE, and Hui Li, Member, IEEE "Privacy-Preserving Public Auditing for Shared Data in the Cloud" IEEE transactions on Parallel and Distributed Computing

c
o
m
p
u
e

R

=

e
(
u
,

v
)
r

a
n
d

h