International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 03 Issue: 06 | June-2016 www.irjet.net p-ISSN: 2395-0072

An Efficient Auditing Protocol for Dynamic Data Sharing in Cloud Computing

RAVIRAJ A N[1], PRABODH C P[2]

[1]*M.Tech, Dept. Of Computer Science and Engineering, Siddaganga Institute of Technology, Karnataka, India*
[2]*Assistant Professor, Dept. Of Computer Science and Engineering, Siddaganga Institute of Technology, Karnataka, India*

Abstract - *Cloud Computing has many benefits such as using applications over the internet, configuring the applications anytime. The Public Cloud allows systems and services to be easily accessible to the general public. Now a days cloud storage is not only used for storage purpose but also users try to change the data anytime. Data integrity and security are major concerns for users of cloud computing. Due to the trustworthiness of data in cloud storage services, wants to check the data stored in the cloud so that they can know the data stored without any error. However, for the majority of the earlier schemes, only the data owner who has the secret keys can edit the data and other users can only read . Recently, attempts are made to allow multiple cloud users to edit data. In this paper, a efficient integrity auditing scheme for data sharing with multiple user editing, public auditing in cloud is implemented.*

***Key Words*: Cloud storage, public verification, Integrity auditing, dynamic data, batch auditing.**

1.INTRODUCTION

In recent times cloud computing is used everywhere. Cloud Computing has many benefits such as using applications over the internet, configuring the applications anytime. The Public Cloud allows systems and services to be easily accessible to the general public. Now a days cloud storage is not only used for storage purpose but also users try to change the data anytime. In public cloud model, resources are shared publicly, therefore does not ensure high level of security. The major issue in cloud computing is security for the data. In several repository like dropbox where several users work in group, where they not only store the data but also do lot of modifications. Even the well known cloud service providers may loose data due to some technical failure, which makes users to think that stored is unharmed or not .

The major issue in cloud storage is data integrity. Due to this issue users wants to check the data stored in the cloud so that they can know the data stored without any error. To check the integrity of data many auditing protocols are proposed but all those schemes provides modification only done by the data owner and all other users only can see the data.

The major challenges in auditing protocol for dynamic data sharing which supports multiple user editing and also user revocation is the aggregation of generated signature tags and efficient user revocation and data auditing. In this paper, a efficient integrity auditing scheme for data sharing with multiple user editing, public auditing in cloud is proposed, based on polynomial authentication tags, cloud will consolidate authentication tags from multiple users into one tag when sending the integrity proof information to the verifier. The trustworthiness of our design during the user revocation procedure is improved by using Shamir's Secret Sharing scheme [6].

The rest of the paper contains related works, system model , threat model ,construction and application, conclusion.

2. RELATED WORK

Considering the issue on data integrity many schemes has been proposed, in 2009 Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing [4]by Q.Wang, C. Wang, J. Li ,K. Ren and W. Lou. This work studies the problem of ensuring the integrity of data storage in Cloud Computing. It considers 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. The disadvantage here is only the data owner has the ability to edit the data. In 2013 Secure and Constant Cost Public Cloud Storage Auditing with Deduplication by J. Yuan and Shucheng Yu[9]. This design allows deduplication of both files and their corresponding authentication tags. Here the disadvantage is it concentrates only on file uploading and User revocation is not considered.

All the above attempts considers the integrity of data where only the data owner has the ability to modify the data.

3. MODELS

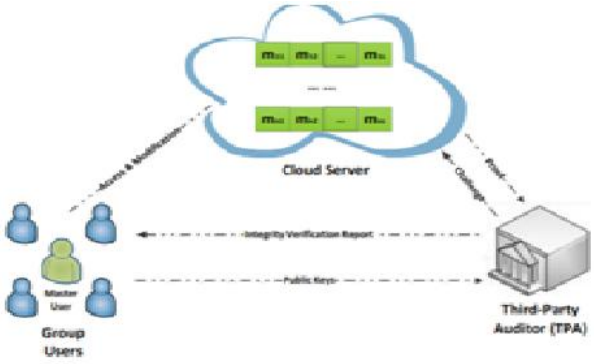

Fig -1: System Model

A. System Models

Our system consists of three parts. Cloud server, group users and third-party auditor .

1. **Cloud server**: The cloud server provides storage maintained by the cloud service providers. The cloud is untrusted because it stores the data remotely data may be lost due to technical failure.

2. **Group users:** Group users consists of master user and users. Master user is the data owner who creates the group and shares the file with other users. Master user creates the public key and secret key and shares with group users. All other user with the public key generates their own secret key and modifies the data without the help of maser user.

3. **Third Party Auditor:** When any user wants to check the integrity of data user sends the challenge request to TPA the TPA then checks the data using challenge algorithm and prove algorithm and finds the data is tampered or not.

The stored data is divided into many blocks. For integrity verification, each block is linked with an signature tag that is initially created by the master user. When a user edits block, then user update the corresponding signature tag with own secret key.

2.Threat models

1.Technical failure of the cloud service provider.

2. Any attacker who tries to illegally to corrupt the data.

3. Revoked user who tries to assume identity of other valid user.

Problem Statement:

Cloud-based storage synchronization platforms where several users work in group for accessing and editing files on cloud servers anytime. For right execution of combined applications one major concern is integrity of data. That is each data modification is done by approved group user and the data should be undamaged. This issue is vital specified by the piece of evidence that cloud storage platform even recognized cloud platforms, may occur some failures, and external malicious attacks . Further we seen that there have been many users reported large differentiation between the numbers of data altered and those agreed by service providers which makes users to think about their data safety.

4. SHAMIR SECRET ALGORITHM

A Shamir secret algorithm is used to share a secret. Shamir secret sharing technique divides the data M into b parts. The user can get the data M with b parts. However if we have the full information of b-1 parts does not discloses any idea of M. This idea allows us to create strong key managing techniques for cryptographic system that can do job of strongly and accurate even when someone destroys the parts. This scheme is based on polynomial interpolation. The main method of an Shamir Secret Sharing scheme is that, let b points uniquely defines a b-1 degree polynomial. For the following b-1 degree polynomial we have $f(x)=a_{b-1}x^{b-1}+.....+a_1x+a_0$ where $a_{b-1}.....a_1 \leftarrow Z_q^*$ then, the secret is $M=a_0$ and each part of this secret is a point of polynomial f(x). With any b points of b-1 degree polynomial f(x) we can get secret M with polynomial interpolation. For more information about Shamir Secret Sharing refer (15).

DEFINITIONS

Key Generation Procedure: The master user uses this key generation algorithm to create public keys and secret keys for the group.

Setup Procedure: The master user uploads the file to cloud for storage and generates corresponding signature tags.

Update algorithm: Any group user who wishes to edit the stored data in cloud runs the update algorithm by not taking the help of the master user.

Challenge and Prove algorithm: Challenge algorithm to challenge and prove algorithm to confirm the proof that it really stores the data accurately by running the Prove algorithm.

Verify algorithm: By proof message and public keys, the TPA can check the stored data integrity in cloud using the Verify algorithm.

User Revocation algorithm: The master user runs this algorithm to revoke the user. When a user exits the group or disobedience of a user is found.

5. DETAIL CONSTRUCTION

$H(\cdot)$	One-way hash function [25]
G	A multiplicative cyclic group
q	The prime order of Group G
$e: G \times G \rightarrow G_1$	A bilinear map
g, u	Random generators of group G
λ	Security parameter
F	Data file that will be split into n blocks
m_i	A data block of file F and will be split into s elements
m_{ij}	A block element of data block m_i
σ_i	Authentication tag generated for data block m_i
$f_{\vec{\alpha}(x)}$	a polynomial with coefficient vector $\vec{\alpha} = (\alpha_0, \alpha_1, \cdots, \alpha_{s-1}), \alpha_j \in Z_q^*$
$\varepsilon_k, \alpha, \mu, R$	Random numbers, where $0 \leq k \leq K-1$

Fig -2: Preliminaries

Table shows the notation used in the scheme. In this scheme, there are *W* users a_w in a group and a_0 is the master user and shares the data with the group. So, a_0 can withdraw any other users when required. All users of group have the privilege to access and modify the data.TPA performs the data integrity auditing procedure. The algorithms used are: Key Generation, Setup, Update, Challenge, Prove, Verify and User Revocation.

Key Generation Procedure: The master user uses this key generation algorithm to create public keys and secret keys for the group. Each user a_w , $0 \leq w \leq W-1$ in the group first randomly selects $\varepsilon_w \leftarrow Z_q^*$ and generates $v=g^{\alpha\varepsilon_o}$, $W_w = g^{\alpha\varepsilon_w}$. Master user a_0 also randomly selects $\alpha \leftarrow Z_q^*$and generates $\{g^{\alpha_j}\}$, $0 \leq j \leq s+1$.The public keys ,master keys of the system and secret keys of users are:

$$P = \left\{ g, a, q, v, \{g^{\alpha^j}\} 0 \leq j \leq s+1, \left\{ w_w, g^{\frac{\varepsilon_0}{\varepsilon_w}} \right\} \right\} \quad 0 \leq w \leq W-1$$

M=$\{\varepsilon_0, \alpha\}$

S= $\{\varepsilon_w\}$ $1 \leq w \leq W-1$

Setup Procedure: The master user uploads the file to cloud for storage and creates equivalent signature tags. To store the file into cloud , the master user a_0 divides the block into b blocks and each one block has r elements: $\{b_{ij}\}$, $1 \leq i \leq b$, $0 \leq j \leq r-1$.Then, a_0 creates signature tag σ_i for each block as

$$\sigma_i = (a^{B_i} \cdot g^{f_{\beta_i}(\alpha)})^{\varepsilon_0}$$

Update algorithm: Any group user who wishes to edit the data stored in cloud runs the update algorithm by not taking the help of the master user. If a group user a_w $w \neq 0$ modifies a data block b_i to $b`_i$ and generates the tag for $b`_i$ from his own secret key ε_w as

$$\sigma_i' := (a^{B_i'} \cdot g^{f_{\beta_i'}(\alpha)})^{\varepsilon_w}$$

Challenge algorithm: : The TPA uses the Challenge algorithm to challenge and prove algorithm to confirm the proof that it really stores the data accurately by running the Prove algorithm. For integrity checking, the TPA first randomly selects b blocks as a set B. If the selected b blocks are edited by D user, $0 \leq |D| \leq W-1$. The TPA generates two random numbers R, μ and produces $X = (g^{\frac{\varepsilon_0}{\varepsilon_w}})^R$ where $w \in D$ .Finally, the TPA generates the challenge message C = $\{D, X, g^R, \mu\}$and sends it to the cloud.

Prove algorithm: By receiving the challenge message C =$\{D, X, g^R, \mu\}$, then for blocks in challenge set D edited by user a_w, $w \in D$, the cloud computes

$$\Pi_i = e(\sigma_i, g^R)$$

For those blocks not modified or modified by a_0

$$\Pi_i = e(\sigma_i, g^{\frac{\varepsilon_0 R}{\varepsilon_w}})$$

At last the cloud reacts to TPA with proof information Proof = $\{\Pi, \psi, y\}$.

Verify algorithm: By proof message and public keys, the TPA can verify the integrity of data stored on cloud using the Verify algorithm. $e(H(m),pk) = e(\sigma,g_2)$ If Equation holds good the TPA says the result as Accept, if not says result as discard.

User Revocation algorithm: When user exits the group or disobedience of a user is found, the master user runs the User Revocation algorithm to revoke the user. If a user that needs to be revoked, say a_w, $w \neq 0$, the master user a_0 will inform the cloud, valid group users and the TPA that a_w is revoked. On receiving the user revocation request, the cloud updates the signature tags of blocks that were last modified by a_w.

6. BATCH AUDITING

When several users request their integrity checking at the same time, it becomes difficult for TPA to perform these requests one after the other. If P auditing requests for P encrypted data files from w users. To process this request TPA uses batch auditing to decrease both communication cost and computational cost. Precisely, for a set of *P* different

files $F_p = \{m_{pi,j}\} 1 \le p \le P, 1 \le i \le n_p, 0 \le j \le s_p - 1$, The TPA has to consolidate the integrity auditing requests of P data files into one request and single verification to lessen the cost, where n_p defines amount of blocks in file F_p and s_p states the amount of elements in each block. We propose batch algorithm based on particular file auditing. For batch auditing, Setup and Update algorithms are similar to one file auditing. The challenge procedure for auditing *P* files is similar as the one file. The challenging message $CM = \{D, X, g^R, \mu\}$ consists the information of data blocks of all the *P* files. By getting the challenging message $CM = \{D, X, g^R, \mu\}$, the cloud uses the prove algorithm for each single file and generates ψ_p, π_p , $1 \le p \le P$ after that cloud gathers the proof information into two elements as $\pi = \pi_{p=1}{}^{P}\pi_p$ and $\psi = \pi^{P}{}_{p=1}\psi_t$.The cloud also calculates $y = f_A(\mu) \bmod q$ and $A = \{0, 0, \Sigma^{P}p=1\Sigma i \in Dpi * mti,0,\ldots \Sigma^{T}t=1\Sigma i \in Dpi * mpi,s_p - 1\}$.At last the cloud reacts with TPA with proof $Proof = \{\pi, \psi, y\}$.By receiving the proof information *Proof* , the TPA first computes $\omega_p = \_i \in DBp\ i$ for every file and generates $\eta = u^{\Sigma Pp=1}{}^{\omega p}$. Then it verifies the integrity of *P* files as

$e(\eta, \kappa^{R}{}_{0}) \cdot e(\psi^{R}, \nu \cdot \kappa^{-\mu}{}_{0}) = \pi \cdot e(\kappa^{-y}{}_{0}, g^R)$

If Equaton holds good, the TPA says the verification result *VerifyResult* as *Accept* for these *P* files, if not says *VerifyResult* as *discar*d The correctness of Batch-auditing can be verified by verify result and batch verify result.

7. APPLICATIONS

A possible application of this design is Version Control Systems which are widely used in managing of source code, documents and configuration files for software development. Many developers work as group in VCS, who works on shared source codes. To confirm the changes of all changes of modified files from version to version, VCS store up the first version of a file completely and its following version are stored with the distinct from the earlier version. To use our scheme in VCS, we see the developers as group and who creates the initial file as the master user a_0. Master user a_0 creates the public key and secret key and shares with group users. All other user with the public key generates their own secret key and modifies the data by not taking the help of maser user. When any user wants to check the integrity of data user sends the challenge request to TPA the TPA then checks the data using challenge algorithm and prove algorithm and finds the data is tampered or not. The data stored is divided into of blocks. For integrity verification, each data block is linked with an signature tag that is initially created by the master user. When a user edits block, user update the corresponding signature tag with own secret key. Our scheme can be useful to existing VCS to capably maintain integrity assurance by not changing their original design.

8. CONCLUSIONS

In this paper, we consider the problem of data integrity in cloud storage, which is essential Cloud-based storage synchronization platforms. It involves the consolidation of signature tags to accomplish the precision of data over cloud server. Then proposed an effective dynamic data sharing scheme which supports multiple user to edit the shared file. To support effective several auditing tasks, further explored the technique of bilinear aggregate signature to extend that into a multi-user setting, where TPA can perform multiple auditing tasks simultaneously.

REFERENCES

[1] J. Yuan and S. Yu, "Efficient public integrity checking for cloud data sharing with multi-user modification," in *Proc. 33rd Annu. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Toronto, ON, Canada, Apr./May 2014, pp. 2121–2129.

[2] *Dropbox for Business*. [Online]. Available: https://www.dropbox.com/business, accessed Apr. 24, 2015.

[3] A. Juels and B. S. Kaliski, Jr., "PORs: Proofs of retrievability for large files," in *Proc. 14th ACM Conf. Comput. Commun. Secur. (CCS)*, 2007, pp. 584–597.

[4] 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. 14th Eur. Conf. Res. Comput. Secur.*, Saint-Malo, France, 2009, pp. 355–370.

[5] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, San Diego, CA, UMar. 2010, pp. 1–9.

[6] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. pp. 612–613, Nov. 1979.

[7] D. Eastlake and P. Jones. *US Secure Hash Algorithm 1 (SHA*[Online]. Available: http://www.ietf.org/rfc/rfc3174.txt?number=3accessed Apr. 24, 2015.

[8] A. De Caro and V. Iovino, "jPBC: Java pairing based cryptographyin *Proc. 16th IEEE Symp. Comput. Commun. (ISCC)*, Corfu, GrJun./Jul. 2011, pp. 850–855.

[9] J. Yuan and S. Yu, "Secure and constant cost public cloud storage auditing with deduplication," in *Proc. 1st IEEE Conf. Commun. NetwSecur. (CNS)*, Washington, DC, USA, Oct. 2013, pp. 145–153.