Integrity Verification for Shared Data in Group with User Revocation

M. Suguna, S. Mercy Shalinie and R. Sivaranjani

Abstract Cloud computing provides storage for the multiple users to store and share their data anywhere at anytime basis. There were some security issues faced by the cloud users such as data correctness, data theft, data leakage, privacy on user level because of the third-party data control. One of the major issues in cloud storage is ensuring data integrity when data are shared by multiple users in the cloud and the data owner accesses data locally. To overcome this issue, many public integrity auditing schemes have been proposed where the computation overhead is huge for the data owner. Hence, efficient auditing with minimum overhead at client side is in need. In the proposed method, we have multi-user modification model with user revocation where the auditing work is delegated to a trusted third-party auditor (TPA) on a secure model, thereby reducing the overhead faced by client.

Keywords Public auditing · Cloud storage · Third-party auditor
Block less verification · User revocation · Data integrity

1 Introduction

Cloud computing provides storage for the users to access the data on their own computer's. Cloud is used to connect multiple computers via the digital network through one computer. Some cloud memory such as cloud-based software Dropbox [1] constructs the cloud application. CloudMe [2] has been built as a cloud application. There are two components in cloud architecture; they are front end and back end. The front end is only accessed by client or user, and back end is full of cloud architecture; here cloud controls the storage devices and servers. Cloud storage is a model to store data on multiple virtual servers hosted by TPA rather

M. Suguna (✉) · S. Mercy Shalinie · R. Sivaranjani
Department of Computer Science and Engineering,
Thiagarajar College of Engineering (TCE), Madurai, India
e-mail: mscse@tce.edu

A. M. Zungeru et al. (eds.), *Wireless Communication Networks and Internet of Things*, Lecture Notes in Electrical Engineering 493,
https://doi.org/10.1007/978-981-10-8663-2_5

than being hosted on dedicated servers. There are some types of cloud for user flexibility; they are public cloud, private cloud, community cloud, hybrid cloud.

Cloud computing are of three types: Infrastructure as a Service (IaaS): The IaaS is the base for the cloud. By using the IaaS, the CSP can ensure that the data are secure and the data can be accessed via, firewalls, routers, storage, and other network equipment in the cloud; Platform as a Service (PaaS): In PaaS, a client can create own appliance which runs on contributor infrastructure; and Software as a Service (SaaS): In SaaS, there is no requirement of client side expenditure for servers or software licenses. There are some of the security issues faced by cloud computing which are data integrity, data theft, security on vendor and user level, information and physical security, third-party data control, operational security. Two kinds of threats are prevalent in shared data storage in cloud. First, the client can try to corrupt the data in the shared pool. Second, the CSP can accidently remove or change the data in its memory due to hardware/software crash. The major issue in cloud storage is data integrity. To solve the problem, many mechanisms [3–6] have been proposed and allowed multiple users to conduct integrity checking beyond downloading the whole data from the cloud. In existing, data owner who carry the secret key can only change the data and share in the cloud. To allow group user modification with integrity [7], the data owner needs to stay online, collect the changes made on data from the other clients, and update the verification tags [7] for each modified user with integrity assurance.

In cloud, to support multi-user modification, Wang proposes data integrity upon ring signature [5]. In this scheme, auditing cost is maintained with fixed size in the group. The cloud node is responsible for updating signature in the cloud storage to prevent impersonation attack in the cloud [5]. We need to overcome the following challenges to get efficient user revocation:

(1) Allowing group user to modify or share information in the cloud without the help of data owner and creating individual aggregate tags for each user becomes a problem. This is because the authentication tags must be generated with client's secret key, which is kept secret from all. Without verification tags, user cannot provide integrity verification in cloud. To solve the problem, let users can share the same secret key. By this, all verification tags are in the similar format, and it can be easily coagulated.
(2) Efficient user revocation. All users authenticated tags are needed to be updated in the cloud, and all revoked users authentication tags are also updated and maintained in the cloud so that we can easily remove the secret key of revoked user from the cloud. If any user revoked from the group, then public key of the group is needed to be updated with authentication tag in the cloud.
(3) Public integrity verification. Public auditing is handled by the data owner and also by any clients who hold a public key. In this scheme, we propose a novel integrity auditing scheme for cloud environment to support multiple-user modification which addresses the above challenges. This scheme supports polynomial-based verification tags from multiple clients into one and transfers the information to the auditor. In this scheme, auditing cost is maintained with

fixed size in the group to support group user revocation [4, 8]. The cloud node is responsible for updating the signature in the cloud storage to prevent impersonation attack in the cloud [7, 9]. By using Shamir's Secret Sharing [10], secret divides into *N* polynomial shares. The design of public integrity auditing scheme supports group user modification with blockless verification.

2 Models

2.1 System Model

In system model, cloud consists of three systems: cloud server, public verifier (TPA), and group users. Cloud server maintains storage services to the group users. Group user consists of number of clients, and original user shares data in the cloud. All clients in the group can change and access the data in the cloud. TPA can check the integrity of data using proof information from the cloud. Once user revoked, then user cannot access the information in the cloud. As our proposed scheme allows public integrity auditing, any user who holds public key can act as a TPA in the cloud. The acquired information are stored as structure of files and each file splits into number of blocks with the authentication key that is created by the own user. When client modifies or updates the block, client updates the corresponding verification tag with his/her own secret key without contacting the user. If any user revoked from group, the user cannot access the data in the cloud because TPA verifies and recomputes the public key (for more detail refer Fig. 1) for the group users.

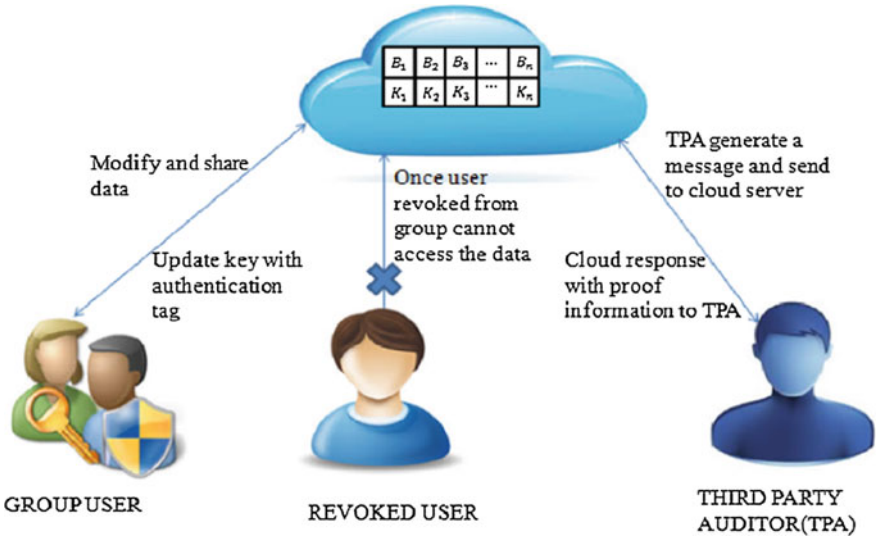

Fig. 1 System model

2.2 Threat Model

In threat model, integrity can be disputed in the following ways: cloud service provider (CSP) can also crash the data, hardware or software failure and operational errors of system administrator, revoked users can also try to access the information stored in the cloud.

We analyze the problem of constructing a public integrity auditing for dynamic data shared in a group with user revocation.

(1) Public auditing: The TPA verifies the data block stored in the cloud without downloading the information from the cloud.
(2) Data correctness: The TPA checks the integrity of data shared in the cloud.
(3) Unforgeability: Group user can only generate valid key or signature on shared data.
(4) Efficient user revocation: If any user is revoked from the group, then the user key and the signed blocks are taken by the original user. Then revoked user accesses are removed from the cloud.
(5) Scalability: Multiple users shared their data in the cloud publicly, and the public verifier is able to handle the multiple auditing tasks simultaneously in efficient manner.

3 Proposed Methodology

Setup: In setup step, original user runs key generation part and generates the public key (P_k), private key (U_k), secret key (S_k) of each user. In this design, each user has unique secret key for modification. To audit the file in the data block, each user needs authentication tag to upload and maintain the log in a cloud.

Update: In update step, all users in the group can change or modify the data in a cloud. After modifying or updating the data, the user needs to compute the tag with their own secret key. While updating the modified data block with the tag in the cloud, it simultaneously updates the data block in the log file.

Challenge: Third-party auditor (TPA) evaluates the integrity verification of data. TPA audits the log file and generates a message and sends to the cloud server.

Prove: In proving step, the cloud waits for the message from challenge step; then, the cloud generates the proof information. Finally, the cloud responds to third-party auditor (TPA) with proof information.

Verify: By using the proof information, the verifier checks the file integrity and analyzes the data integrity.

User revocation: The original user and the cloud check if any user is revoked from the group; then, the authentication tag generated by revoked user and a secret key of revoked user are removed from the tag. Then original user checks the number of tags modified by the revoked users which becomes a potential burden for

the user'. To control the burden from the original user, all tag update operations are handled by cloud because the cloud can support parallel processing. After receiving the message, cloud updates the authentication tag of each block. The verifier and the group users then remove the public information from revoked users. Public verifier (TPA) audits the files last accessed by the revoked users and sends the message to cloud. The cloud checks the message and log file and sends the proof information to the TPA. TPA checks the integrity of the data and analyzes the report as accept or reject.

Step 1. In the initial setup, an original user s_0 evaluates key algorithm and creates the public key (P_k), secret key (U_k), private key (S_k) for every user. Using file processing algorithm, each file F splits into n blocks of data and each block then divides into s elements. For every user tags σ_i are generated for the files to be uploaded and these tags are stored at third-party verifier.

Step 2. In update step, multiple users can share or modify the data in the cloud simultaneously and a new authentication tag σ_i is computed for each modification or updation done by the user. During download, TPA generates a challenge message

$$\delta_i = e\left(\sigma_i, g^{\frac{\varepsilon_0 R}{\varepsilon_k}}\right) e\left(\left(u^{B_i}.g^{\overrightarrow{f}(\gamma)}_{\beta_i}\right), g\right)^{\varepsilon_0 R} \quad \text{and}$$

$$\delta_i e\left(\sigma'_{i,}\left(g^{\frac{\varepsilon_0}{\varepsilon_0+\tau}}\right)^R\right) e\left(\left(u^{B_i}.g^{\overrightarrow{f}(\gamma)}_{\beta_i}\right), g\right)^{\varepsilon_0 R}$$

sends it to the cloud, and cloud acquires the challenge message from the auditor (TPA) and creates proof information and sends it to the auditor.

Step 3. By utilizing the proof information, verifier checks data correctness verification on download. The original user s_0 runs a Shamir's Secret Sharing scheme and generates N points. Each cloud node needs to update a piece of the tag. If any client is revoked from the group, then the group key is updated and the updated key is circulated amongst all group users.

Step 4. By this, public auditing and user revocation are achieved securely using a trusted verifier. By using dynamic auditing scheme, any client in the group can easily modify and update blocks in the data in single block using dynamic operation.

Multi-file auditing: In cloud, group users often make changes in blocks to ensure data integrity TPA audit the data in blocks frequently. So the computational cost is inefficient. To control batch auditing operation performed in the cloud. To audit N number of information blocks in file batch, challenge converts N number of data blocks into one message and one verification step to reduce cost. By this multi-file, auditing enables the verifier to perform integrity auditing for N number of files as single file cost.

4 Support Dynamic Operations

Any client in the group can easily change the information in the cloud using dynamic operation. Dynamic operation supports insert, delete, update on single block. By using index hash table, all users can efficiently perform dynamic operation on shared data. Client can modify the single data block in shared data by using insert and delete operations. The modified blocks, are all changed and if users share the data, then the signature of the block has been recomputed the signature of the block even though the content has not been changed. Here, I denotes Index and B denotes block in the table.

By using hash table [5], user can perform dynamic operation efficiently. In our appliance, the identifier is described as $id_j = \{V_j, r_j\}$ where v_j is denoted as the virtual values of blocks a_j and r_j is a value created by a hash basis H_2. The value of r is generated by the H_2; it shows that each block has a solitary identifier and the virtual indices are able to ensure that all shared data are in right order in index table. (Figs. 2 and 3 show the multiple dynamic operations with our index hash table). Here, ρ supports sufficient number of blocks for the group, so that there is no way to have the same virtual indexes in the table.

I	B	V	R
1	a_1	ρ	r_1
2	a_2	2ρ	r_2
3	a_3	3	r_3
4	a_4	4ρ	r_4
⋮	⋮	⋮	⋮
N	a_n	$n\rho$	r_n

Insert ⟶

I	B	V	R
1	a_1	ρ	r_1
2	a_2'	$\lvert 3\rho/2 \rvert$	r_2'
3	a_2	2ρ	r_2
4	a_3	3	r_3
5	a_4	4ρ	r_4
⋮	⋮	⋮	⋮
n	a_n	$n\rho$	r_n

Fig. 2 Insert block into dynamic data operation using hash table as identifier

I	B	V	R
1	a_1	ρ	r_1
2	a_2	2ρ	r_2
3	a_3	3	r_3
4	a_4	4ρ	r_4
⋮	⋮	⋮	⋮
N	a_n	$n\rho$	r_n

Update ⟶
Delete ⟶

I	B	V	R
1	a_1'	ρ	r_1'
2	a_2	2ρ	r_2
3	a_3	3	r_3
4	a_4	4ρ	r_4
⋮	⋮	⋮	⋮
n	a_n	$n\rho$	r_n

Fig. 3 Update blocks and delete blocks in dynamic data operation using a hash table as identifier

5 Performance Analysis

In this mechanism, we evaluate the performance of the proposed method by storing the files on CloudMe and implement the algorithms using Java. On CloudMe, we deploy different text files accessed by the users and modify file with the authentication tag. The computational cost is calculated for the user and verifier by varying the file size. The communication cost is analyzed through the challenge message and proof information. To check the verification tag generation time, we increase the number of blocks in the file. Our result depicts the analysis of verification tag generation. To verify the file size in the auditing, we alter the number of blocks from 1000 to 100,000. As depicted in Fig. 4, the tag generation time is propositional to the number of blocks from 10 to 100 s.

Figure 5 shows that to revoke a user, the advanced user revocation algorithm consumes minimal storage overhead for tag updation for each user which leads to increase in communication cost.

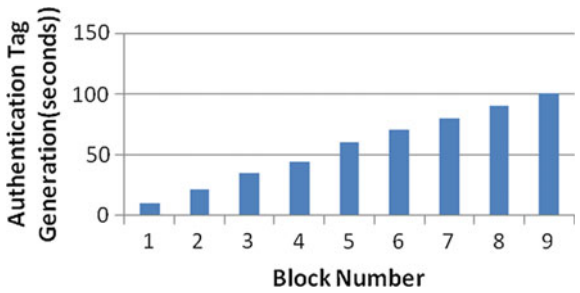

Fig. 4 Authentication tag generation time

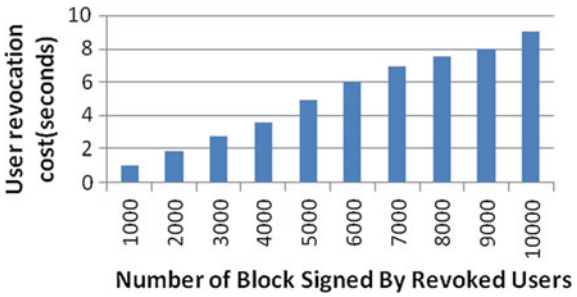

Fig. 5 User revocation cost on cloud

6 Conclusion

Public integrity auditing mechanism checks the data correctness in cloud-sharing resources. To support group user modification and dynamic auditing, user generates an authentication tag to insert, delete, and update the data in the block. TPA can verify the data integrity with blockless verification. Authentication tag generation is performed by user revocation algorithm. Although the advanced user revocation algorithm requires more cost and cloud-sharing resources, it achieves better reliability for the system. In this scheme, we extend our mechanism to support batch auditing but there are some issues that will be continued as a future work. One of them is traceability, which means ability to reveal the identity of the signer based on verification meta data. Another issue is the cloud reciprocity problem (although original user back up his/her data in multiple CSPs, CSPs might exercise mutual aid to avoid the huge cost of data lost). Thus, we can achieve data correctness for multiple tasks through batch auditing technique.

References

1. Dropbox (2007) A file-storage and sharing service. Dropbox [Online]. Available: http://www.dropbox.com/
2. CloudMe. A file-storage and sharing service in cloud. CloudMe [Online]. Available: http://www.CloudMe.com/
3. Wang C, Wang Q, Ren K, Lou W (2010) Privacy-preserving public auditing for data storage security in cloud computing. In: Proceedings of the 29th IEEE international conference on computer communications (INFOCOM), San Diego, CA, USA, Mar 2010, pp 1–9
4. Zhu Y, Wang H, Hu Z, Ahn GJ, Hu H, Yau SS (2011) Dynamic audit services for integrity verification of outsourced storages in clouds. In: Proceedings of the ACM symposium on applied computing (SAC), pp 1550–1557
5. Wang B, Li B, Li H (2012) Oruta: privacy-preserving public auditing for shared data in the cloud. In: Proceedings of the IEEE 5th international conference on cloud computing (CLOUD), Washington, DC, USA, Jun 2012, pp 295–302
6. Jiang T, Chen X, Ma J (2016) Public integrity auditing for shared dynamic cloud data with group user revocation. IEEE Trans on Comput. Citation information: https://doi.org/10.1109/tc.2015.2389955

7. Yuan J, Yu S (2015) Public integrity auditing for dynamic data sharing with multiuser modification. IEEE Trans Inf Forensics Secur 10(8):1717
8. Wang B, Li B, Li H (2013) Public auditing for shared data with efficient user revocation in the cloud. In: Proceedings of the 32nd IEEE international conference on computer communications (INFOCOM), Turin, Italy, Apr 2013, pp 2904–2912
9. Wang B, Li B, Li H (2015) Panda: public auditing for shared data with efficient user revocation in the cloud. IEEE Trans Serv Comput 8(1)
10. Shamir A (1979) How to share a secret. Commun ACM 22(11):612–613