Secure Multi-Owner Data Sharing for Dynamic Groups using Proxy-Signature in the Cloud

M. Siva Kishore
M.Tech(CSE)
KHIT,Guntur
India

P. Naga Lakshmi
Assistant Professor(Dept.of CSE)
KHIT,Guntur
India

B. Tarakeswara Rao, PhD
Professor(Dept.of CSE)
KHIT,Guntur
India

ABSTRACT

Cloud computing offers an economical and efficient solution for sharing data among the cloud users with low support without the weight of neighborhood data storing and upkeep. In any case, the administration of the data and services may not be completely dependable on the cloud, as users no longer have physical possession of the outsourced personal data so data integrity protection turns into a troublesome assignment. Keeping up the integrity of shared data services where data is shared among various cloud users is likewise a testing undertaking. This paper gives effective user revocation on multi-owner dynamic group sharing and for that it utilizes Homomorphic straight authenticator with random veiling procedure. Homomorphic authenticable proxy signature scheme with public auditing mechanism checks imparted data integrity along to efficient user revocation. Moreover, these systems can bolster clump inspecting by checking multiple evaluating errands at the same time.

Keywords

Proxy Server, privacy-preserving, public auditing, shared data, user revocation, cloud computing.

1. INTRODUCTION

Cloud computing gives qualities as on-demand self-service, ubiquitous system access, area free asset pooling, quick asset versatility, and utilization based evaluating and transference of danger, this trademark makes Cloud computing suitable for undertakings. One crucial part of this outlook changing is that data is being incorporated or outsourced to the Cloud. From client's viewpoint, including both people and IT endeavors, remotely putting away data to the cloud give favorable circumstances as a help of the weight for capacity service, general data access with autonomous geological areas, and evasion of capital use on equipment, software, and staff support, and so forth. As the client doesn't have control over data subsequent to putting away it in a cloud so the rightness of the data in the cloud is being put at danger because of the accompanying reasons. Most importantly, in spite of the fact that the foundations under the cloud are considerably more capable and solid than individualized computing gadgets however there is the danger of data integrity. Furthermore, Cloud service Provider (CSP) may by dispose of data that has not been or is infrequently gotten to, or even shroud data misfortune occurrences in order to keep up notoriety. To address these issues, public key based Homomorphism straight authenticator (HLA) system can be utilized for integrating so as to inspect and the

HLA with random veiling, convention ensure that the TPA couldn't realize any information about the data substance put away in the cloud server amid the proficient reviewing procedure. The conglomeration and logarithmic properties of the authenticator further advantage the outline for clump auditing. In offer data services, as data is altered by diverse clients that is the reason distinctive blocks in shared data are marked by diverse users. Each block is joined with a mark and integrityof data depends on the rightness of the considerable number of marks. Once a client is denied from a gathering, around then the block marked by the repudiated client must be surrendered by the current client for security reasons. In fundamental system, first data blocks are downloaded by existing client and then transfer procedure is done in the wake of existing so as to check the rightness and leaving of block client, which brings about huge measure of communication and computation cost because of substantial size of shared data in cloud

2. RELATED WORKS

The concept of public auditability was given by Ateniese et al. [8]. They have described this concept in their defined provable data possession (PDP) model for making sure the ownership of data files on no trustworthy storage and used Rivest Shamir Adleman based Homomorphic linear authenticators for auditing of outsourced data. Provable data possession model allows client (who has stored data on untrusted server) to verify, that the server possesses the original data without retrieving it. PDP model creates probabilistic proofs of possession by sampling random sets of blocks from the server. This significantly minimizes I/O costs. The client maintains a constant amount of metadata to verify the proof. The response protocol sends a modest, constant quantity of information, which reduces network communication. Hence, the PDP model for distant information inspection supports large data sets in widely-Cloud storage systems. Authors have presented two provably-secure PDP schemes that are more capable than prior 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 by execution confirm the practicality of PDP and tell that the performance of PDP is restricted by disk Input output and not by cryptographic computation. For auditors who are external, linear combination of sample blocks were required and when directly used, their protocol did not provided privacy preserving and thus may leak the user data to auditors.

Shacham et al. [7] built proof of irretrievability (PoR) model and constructed a random linear function based Homomorphic authenticator which enables limitless number of inquiry and requires minimal communication overhead. Shacham et al.s first methods, built from BLS signatures and secure in the random oracle model, characteristics of a proof-of retrievability protocol in which the clients inquiry and servers response are both very short. This method allows public verifiability: anyone can act as a verifier, not only the file owner. Second method, which builds on pseudorandom functions (PRFs) and is protected in the regular model, allows only secret confirmation. It features a proof-of-retrievability protocol with a yet shorter servers response than the first method proposed, but the clients query is very long. Both methods depend on Homomorphic characteristics to comprehensive evidence into one small authenticator value.

Wang et al [6] projected a theory to combine BLS-based HLA with MHT to sustain equally publicauditability and full data dynamics. Considered a like support for incomplete dynamic data storage in a disseminated situation with added quality of data error localization. To efficiently carry public auditability without having to recovering the data blocks themselves, resort to the homomorphic authenticator system. Homomorphic authenticators are unforgeable metadata generated from individual data blocks, which can be strongly aggregated in such a way to reassure a verifier that a linear combination of data blocks is appropriately computed by verifying only the aggregated authenticator. In this design, here proposal is to use PKC based homomorphic authenticator (e.g. BLS signature or RSA signature based authenticator) to implement the verification protocol with public auditability. In the following explanation, there is present the BLSbased method to illustrate the design with data dynamics support. As will be shown, the schemes designed under BLS construction can also be implemented in RSA construction.

K.Ren et al [5] proposed privacy preserving system where public key based homomorphic authenticator is combined with random masking which fulfill the requirement of efficient audit without demanding the local copy of data and user data privacy. Explored the technique of bilinear aggregate signature for multi user setting which allow third party auditor execute multiple number of auditing task together.

C.Wang et al [4] proposed privacy-preserving public auditing system for data storage security in Cloud Computing. Homomorphic linear authenticator and random masking have been used to guarantee that the 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. Considering TPA may concurrently

handle multiple audit sessions from different users for their outsourced data files, privacy-preserving public auditing protocol further extended into a multi-user setting, where the TPA can perform multiple auditing tasks in a batch manner for better efficiency. Extensive analysis shows that discussed schemes are provably secure and highly efficient.

G.Wang et al [3] Proposed proxy provable data possession protocol for remote data checking as PPDP is major concern in public cloud when client cannot perform the remote data possession checking

This proposed protocol is based on bilinear pairing technique and through security analysis and performance analysis author has proved that the protocol is provable secure and efficient. Li et al [2] has proposed a privacy preserving mechanism that supports public auditing on shared data stored in the cloud. He has used ring signature to compute verification metadata and identity of signer is kept private from public verifier, who are able to efficiently verify shared data integrity without retrieving the entire file. Additionally this mechanism is able to perform multiple auditing tasks simultaneously instead of verifying them one by one and experimental results demonstrate the effectiveness and efficiency of this mechanism when auditing shared data integrity’s. Wang et al [1] proposed public auditing mechanism for shared data using Homomorphism authenticator and efficient user revocation in cloud. Here semi trusted cloud re-signs the blocks which were signed by revoked user, using proxy re-signature and save a significant amount of computation and communication resources during user revocation.

3. SYSTEM METHOD

3.1 Problem definition

While utilizing cloud services as data storage and data sharing in a group, Trustworthiness of individual and shared data on cloud and client denial are significant concerns. This paper utilizes the idea of Homomorphism direct authenticator with the arbitrary concealing strategy for individual data and Homomorphic authenticable intermediary mark plan with open examining system for shared data and client contradiction.

3.2 Methodology

Diagram shows the architecture of system flow and methodology is explained in detail below:

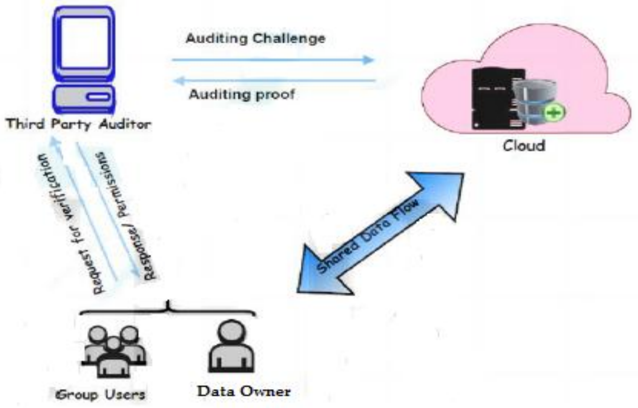

Fig. 1. System Flow

The cloud server, a group of users and a public verifier as

Shown in fig 1. There are two types of users in a group: the Data Owner and a number of group users. The Data Owner initially creates shared data in the cloud, and shares it with group users. Both the Data Owner and group users are members of the group. Every member of the group is allowed to access and modify shared data. Shared data and its verification metadata are both stored in the cloud server. A public verifier, such as a third party auditor providing expert data auditing services or a data user outside the group intending to utilize shared data, is able to publicly verify the integrity of shared data stored in the cloud server. When a public verifier wishes to check the integrity of shared data, it first sends an auditing challenge to the cloud server. After receiving the auditing challenge, the cloud server responds to the public verifier with an auditing proof of the possession of shared data. Then, this public verifier checks the correctness of the entire data by verifying the correctness of the auditing Proof. Essentially, the process of public auditing is a challenge and- response protocol between a public verifier and the cloud server.

3.3 Privacy-preserving public auditing for secure data storage [4]:

Homomorphic linear authenticator with random masking technique is used when there is a need of public auditability without retrieving the data blocks. HLAs are unforgeable verification metadata which are used to authenticate the integrity of a data block. HLAs can be aggregated. It is possible to compute an aggregated HLA which authenticates a linear combination of the individual data blocks. This scheme uses below algorithms:

•KeyGen: KeyGen is a key generation algorithm that is executed by the user to setup the scheme.

•SigGen: SigGen is executed by the user to produce verification metadata, which may consist of signatures, or other linked information that will be used for executing audit.

•GenProof: GenProof is executed by the CS to produce a verification of data storage rightness.

•VerifyProof: is executed by the TPA to audit the verification from the CS.

This Public auditing technique works in two phases:

Setup: Setup phase works with two algorithms, Key-Gen and SigGen. By running KeyGen algorithm, user initializes the public and secret parameters of the system and verification metadata for data file is generated using SigGen algorithm. Data file F and the verification metadata is stored on cloud server and user deletes its local copy. User may alter the data file F by expanding it or including additional metadata to be stored at server as a part of pre-processing.

Audit: The Audit phase works with two algorithms, GenProof and VerifyProof. Whenever TPA wants to verify that the cloud server has retained the data file F properly or not, at that time TPA is sending audit message or challenge to cloud server. By running GenProof, cloud server will derive a response message from a function of the stored data file F and its verification metadata. Then TPA verifies the response by running algorithm VerifyProof. Flow of scheme: First third party auditor (TPA) retrieves file and verifies its signature, if signature verification occurs successfully then next step is being performed, else the process is terminated. In next step TPA generates a random challenge request and send is to server. After receiving the challenge request, server computers. Here is linear combination of sampled blocks, σ is aggregated authenticator and R is calculated for inserting the random masking so that by evaluating the linear equations, TPA cannot predict the data. Server finally computes μ by using and R and send the calculated values and R to TPA as a storage correctness proof. Then TPA verifies the response by running algorithm VerifyProof.

3.4 public auditing scheme for shared data and user data revocation[1]:

Homomorphic authenticable proxy resignature scheme with Panda public auditing mechanism is used for public auditing of shared data with efficient user revocation in cloud. Here semi trusted cloud re-signs the blocks which were signed by revoked user, using proxy re-signature and save a significant amount of computation and communication resources during user revocation. Additionally it support dynamic data and batch auditing for handling number of task simultaneously.

Scheme Details: Let G1 and G2 be two groups of order p, g and w be the generator of G1.e:G1 X G1 → G2 be a bilinear map.(e,p,G1,G2,g,w,H) are the global parameters where H is the hash function. Total number of blocks in shared data is n, shared data is described as M = m1,… ,mn) and total number of users in a group is d. Flow of this mechanism is described below with the help of algorithms.

KeyGen: This is key generation algorithm and here user generates their public and private key. Here Data Owner creates a user list which contains ids of all the users in the group. This user list (UL) is public and signed by the original user.

ReKey: Through this algorithm cloud computes resigning key for each pair of user in group and it is assumed that private channels as SSL exist between each pair of entities and there is no collusion. For this cloud generates a random r and send it to user A, user A calculates some value and send it to user B then user B do same calculation and pass the value to cloud and by this value cloud recovers the Rekey.

Sign: This algorithm is used for signing the block by Data Owner i.e. creator of data and if a user in the group modifies a block in shared data, the signature on the modified block is also computed as in Sign. Given private key as

ski = πi, block and its block identifier idk. User ui outputs the signature on block.

ReSign: This algorithm is used for re-signing the blocks by cloud which were previously signed by revoked users. Re-signing key, Public key, signature, block, block identifier, cloud checks that. If the verification result is 0, the cloud outputs 1; otherwise, it outputs.

ProofGen: In ProofGen algorithm, cloud is able to generate proof of possession of shared data under the challenge of public verifier and this works in two parts. In first part public verifier generates audit message and send it to cloud and in second part cloud generates a proof of possession of shared data M, after receiving the auditing message.

Proof Verify:
By using Proof Verify algorithm public verifier is able to check the correctness responded by cloud. Here verification of shared data is done by using challenge and response protocol between the cloud and public verifier. Given an auditing message auditing proof and all existing users public key and public verifier checks the correctness of this auditing proof as below, if the result is 1, verifier believes that integrity in all the blocks in shared data M is correct otherwise public verifier outputs 0. In ReSign algorithm, Cloud always translates revoked users signature into signature of data creator (original user) because Data Owner acts as group manager and assumed to be secure in this mechanism. Another way to decide which resigning key should be used when a user is revoked from the group is to ask the Data Owner to create a priority list (PL). Every existing user's id is in the PL and listed in the order of re-signing priority. When the cloud needs to decide which existing user the signatures should be converted into, the first user shown in the PL is selected. To ensure the correctness of the PL, it should be signed with the private key of the original user. Based on the properties of bilinear maps; the correctness of this mechanism in Proof Verify can be explained.

A. Proxy Re-signatures

In our Proposed framework may deceive verifiers about the incorrectness of shared data with a specific end goal to spare the reputation of its data services and abstain from losing money on its data services. In addition, we additionally expect there is no collusion between the cloud and any user amid the outline of our instrument. By and large, the incorrectness of share data under the above semi-trusted model can be presented by hardware/software failures or human blunders happened in the cloud. Considering these elements, users don't completely believe the cloud with the integrity of shared data. In our system, by using the thought of proxy re-signatures, once a user in the group is revoked, the cloud can resign the squares, which were marked by the revoked user, with a re-marking key. In the meantime, the cloud, which is not in the same trusted area with every user, is only ready to convert a signature of the revoked user into a signature of a current user on the same square. Two-stage authentication technique used to give more security Easily Revocable of signatures for the current users. The general population verifier can review the integrity of shared data without retrieving the entire data from the cloud.. Blocking users account. Login with secret key each time.

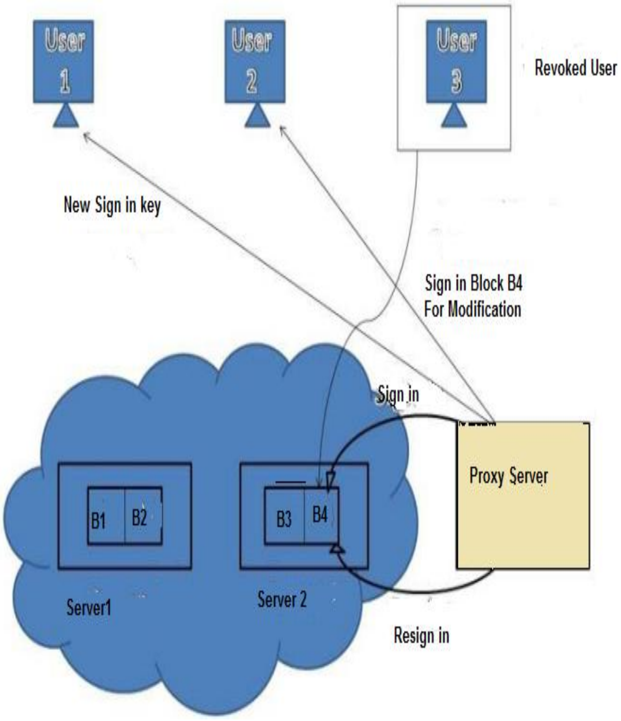

Fig 2. Architecture for Proxy re-signature.

Proxy re-signatures, first proposed by Blazeetal. Allow a semi-trusted proxy to act as a translator of signatures between two users. More specifically, the proxy is able to convert a signature of Alice into a signature of Bob on the same block. Meanwhile, the proxy is not able to learn any private keys of the two users, which means it cannot sign any block on behalf of either Alice or Bob. In this paper, to improve the efficiency of user revocation, we propose to let the cloud to act as the proxy and convert signatures for users during user revocation [6][7].

4. PROPOSEDSYSTEM ARCHITECTURE

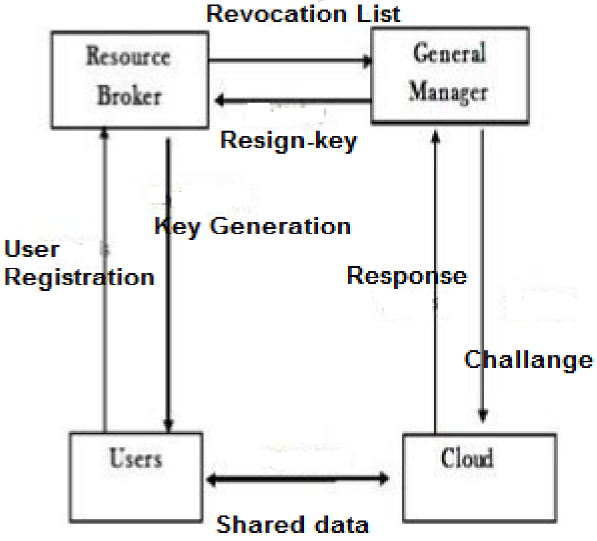

Fig 3. Proposed System Architecture

4.1.User Registration

User registered with their details such as identity (user name, password and email-id).For registered users they will obtain private key, that private key is used for group signature and file decryption. The Resource Broker adds the user identity (ID) to the group user list that will be used in traceability phase.

4.2.File Generation

Group members will store their data in real cloud. The groups members will request with group id and based on the revocation list the TTP allow the data owner to upload the data in the cloud, if their signature is true. If it’s a revoked user, he is not allowing for generating the data and signature verification status false. When generating the data, hash id will be generated that will be used for deleting the data.

4.3.File Access

To access the data that are stored in the cloud, group member will give request as group id, data id. Resource Broker will verify their signature, if the group member in the same group then allow to access file. Group member have rights to access data, but not having rights to delete or modify the data that are stored in the cloud. If any request from revoked user, cloud server won’t allow accessing the data.

4.4.File Deletion

File that are stored in the cloud can be deleted by either group member (i.e., the member who uploaded the file into the server) or by Resource broker. It allows data owners to delete their own files that are stored in the cloud. If any delete request from the group member, cloud server will verify the signature and delete the data file that are stored in the cloud.

4.5.Traceability

Resource Broker will reveal their real identity in case of any dispute occurs. If any malpractice happened inside the organization it can be easily traceable. If any group members are modify or delete the data file of other groups, it can easily identify which member doing such activities.

4.6.User Revocation

User Revocation is performed by the TTP (General Manager).Revocation List is generated by Resource Broker, group members are allowed to encrypt the data and make that data confident against revoked users. Revocation list is bounded by signature to declare its validity.

5. CONCLUSION

In this paper, we propose, a privacy-preserving and public auditing mechanism for shared data in the cloud. We utilize proxysignature schemes to construct Homomorphic authenticators, so that a public verifier is able to audit shared data integrity without retrieving the entire data, yet it cannot distinguish who is the signer on each block. To improve the efficiency of verifying multiple auditing tasks, we further extend our mechanism to support batch auditing. By utilizing Aggregate Signature Schemes the verification and thereby privacy preservation of the data owner is being done and it is seen that the owner could efficiently upload the files and a user could download it using the key which is being sent to his/her mail. The performance and efficiency of the work is valuated.

6. REFERENCES

[1] Boyang Wang, Baochun Li and Hui Li, “Public Auditing for Shared Data with Efficient User Revocation in the Cloud”, IEEE Transactions on services computing, vol. 8, no. 1, January/February 2015.

[2] B. Wang, B. Li, and H. Li, “Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud”, Proc. IEEE CLOUD, pp. 295-302, 2014.

[3] H. Wang, “Proxy Provable Data Possession in Public Clouds”, IEEE Trans. Services Computing, vol. 6, no. 4, pp. 551-559, Oct.-Dec. 2013.

[4] C. Wang, Q. Wang, K. Ren ,“Privacy-Preserving Public Auditing for Secure Cloud Storage Auditing”, IEEE transaction on computer, 2013.

[5] C. Wang, Q. Wang, K. Ren, and W. Lou, “PrivacyPreserving Public Auditing for Data Storage Security in Cloud Computing”, Proc. IEEE INFOCOM, pp. 525-533, 2010.

[6] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, “Enabling Public Verifiability and Data Dynamic for Storage Security in Cloud Computing”,Proc. 14th European Conf. Research in Computer Security (ESORICS09), pp. 355- 370, 2009.

[7] H. Shacham and B. Waters, “Compact Proofs of Retrievability”, Proc. 14th Intl Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT08), pp. 90-107, 2008.

[8] G. Ateniese, R. Burns, R. Curtmola, J. Herring, LKissner, Z. Peterson, and D. Song, “Provable Data Possession at Untrusted Stores”, Proc. 14th ACM Conf. Computer and Comm. Security (CCS07), pp. 598-610, 2007.

[9] Shamir, “How to Share a Secret”, Comm. ACM, vol. 22, no. 11,pp.612-613,No v. 1979.