



Secured Data Sharing In Dynamic Public Data Integrity for Auditing by Efficient Use of Vector Commitment

T. PremchanderAssociate Prof., Department of CSE TKRCET,
Hyderabad, India**A. Rajani**PG Scholar in Department of CSE TKRCET,
Hyderabad, India

Abstract: *The old scheme for secure and efficient public dynamic public data integrity auditing for shared dynamic data is not still secure. Here in this paper we present collusion attack in existing scheme. Also it provides efficient public integrity auditing scheme. There is efficient use of vector commitment and verifier local revocation group signature. We implement concrete scheme for group signature. The scheme support public checking, efficient user revocation, and properties like confidently, efficiency, count ability and traceability. Finally we compare our scheme with old which shows good result in security.*

Keywords: *Public integrity auditing, dynamic data, vector commitment, group signature, cloud computing.*

I. INTRODUCTION

Cloud Storage service are such as simple storage services in online data backup services of Amazon, and practical cloud based software Google drive, drop box, mozy, bitcasa and memo pal have been built for cloud application. There is invalid result in cloud server such as server hardware, software failure, human maintenance and malicious attack. Rabin data dispersion scheme implemented for practical application and overcome above challenges. Author in [10], [11], [12], [18] provide solutions to integrity and availability of remote cloud store. This document is a template. An electronic copy can be downloaded from the conference website. For questions on paper guidelines, please contact the conference publications committee as indicated on the conference website. Information about final paper submission is available from the conference website. . Dynamic scheme means when scheme support data modification only data owner cloud modify data. The limited dynamic scheme cloud only efficiently supports special field operation (eg. Append). The static scheme not support data modification. In publicly verifiable, data integrity check can be performed by data owner and by any third party auditor. Multiple user in group need to share source code they need to access, modify compile and run the shared source code at any time and place. Remote data auditing is only data owner can update its data. Ring signature supports multiple user data operation. The proxy re-signature is private and authenticated channels exist between each pair of entities. Till today is no solution for above problem in public integrity auditing with group user modification.

Real Time Example:

In an Group file sharing environment if an user wishes to revoke from a group then the complexity added to the files shared by that user where someone else in the group need to take authority over their files by downloading and reassigning key to that file. In order to overcome that we appoint an third person where his work is to monitor the files of the revoked user and reassign it to someone else in the group based on owners priority without any overhead of download. Here we generate private and public key based on the prime no. The main aim of this paper is to search for private and public files. In case of public files users can modify their files and update to it.

II. RELATED WORK

Group signatures without revocation. The provably coalition-resistant scalable group signature was described by Ateniese, Cameliash, Joye and Tsudik in 2000 [7]. At that time, the security of group signatures was not totally understood and proper security definitions were given later on by Bellare, Micciancio and Warinschi [9] (BMW) whose model captures all the requirements of group signatures in three properties. In (a relaxation of) this model, Boneh, Boyen and Shacham [16] obtained a construction in the random oracle model [10] with signatures shorter than 200 bytes [13]. In the BMW model, the population of users is frozen after the setup phase beyond which no new member can be added. Dynamic group signatures were independently formalized by Kiayias and Yung [4] and Bellare-Shi-Zhang [11]. In these models, pairing-based schemes with relatively short signatures were put forth in [5]. Ateniese et al. [6] also gave a construction without random oracles using interactive assumptions. In the BMW model [9], Boyen and Waters independently came up with a different standard model proposal [19] using more classical assumptions and they subsequently refined their scheme [21] to obtain constant-size signatures. In the dynamic model [11], Growth [8] described a system with constant size signatures without random oracles but this scheme was rather a feasibility result than an ancient construction. Later on, Growth gave [9] a fairly ancient realization {with signatures consisting of about 50 group elements { in the standard model with the strongest anonymity level. Revocation. In group signatures, membership revocation has received much attention in the last decade [2, 8, 9, 18] since

revocation is central to digital signature schemes. One simple solution is to generate a new group public key and deliver a new signing key to each unrevoked member. However, in large groups, it may be inconvenient to change the public key and send a new secret to signers after they joined the group. An alternative approach taken by Bresson and Stern [22] is to have the signer prove that his membership certificate does not appear in a public list or revoked certificates. Unfortunately, the signer's workload and the size of signatures grow with the number of expelled users. Song [5] presented an approach handling revocation in forward-secure group signatures. However, verification takes linear time in the number of excluded users. Using accumulators [12], Camenisch and Lysyanskaya [9] proposed a method (notably followed by [6]) to revoke users in the ACJT group signature [7] while keeping $O(1)$ costs for signing and verifying. While elegant, this approach is history-dependent and requires users to keep track of all changes in the population of the group: at each modification of the accumulator value, unrevoked users need to update their membership certificates before signing new messages, which may require $O(r)$ exponentiations (if r is the number of revoked users) in the worst case. Brickell [3] suggested the notion of verifier-local revocation group signatures, which was formalized by Boneh and Shacham [1] and further studied in [5]. In their systems, revocation messages are only sent to verifiers (making the signing algorithm independent of the number of revocations). The group manager maintains a revocation list (RL) which is used by verifiers to make sure that signatures were not generated by a revoked member. The RL contains a token for each revoked user and the verification algorithm has to verify signatures w.r.t. each token (a similar revocation mechanism is used in [4]). As a result, the verification cost is inevitably linear in the number of expelled users. More recently, Nakanishi, Fuji, Hira and Funabiki [9] described a construction with constant complexities for signing/verifying and where group members never have to update their credentials. On the other hand, their proposal has the disadvantage of linear-size group public keys (in the maximal number N of users), although a tweak allows reducing the size to $O(N^2)$. In the context of anonymous credentials, Tsang et al. [8,9] showed how to blacklist users without compromising their anonymity or involving a trusted third party. Their protocols either have linear proving complexity in the number of revocations or rely on accumulators (which may be problematic for our purposes). Camenisch, Kohlweiss and Soriente [2] suggested handling revocations by periodically updating user's credentials in which a specific attribute indicates a validity period. While useful in certain applications of anonymous credentials, in group signatures, their technique would place quite a burden on the group manager who would have to generate updates for each unrevoked individual credential. Paragraphs must be indented. All paragraphs must be justified, i.e. both left-justified and right-justified.

A. Text Font of Entire Document: The entire document should be in Times New Roman or Times font. Type 3 fonts must not be used. Other font types may be used if needed for special purposes.

B. Title and Author Details: Title must be in 20 pt Regular font. Author name must be in 10 pt Regular font. Author affiliation must be in 10 pt Italic.

III. ARCHITECTURE DIAGRAM

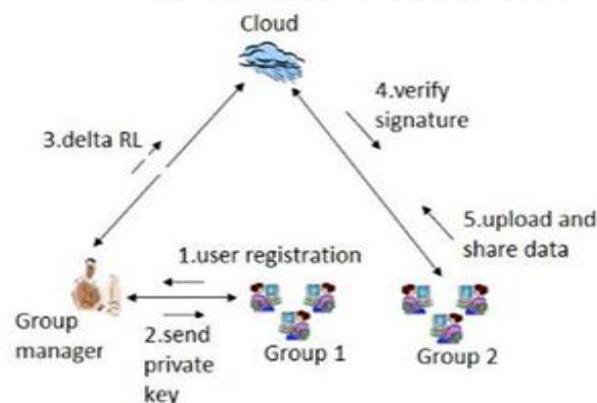


Figure 1. The cloud storage model

File Upload:

File owner allowed uploading data on the cloud either for their private or public use. They act as a Group Manager for the file they upload in cloud. Both the original user and group users are able to access, download and modify shared data. Shared data is divided into a number of blocks. A user in the group can modify a block in shared data by performing an insert, delete or update operation on the block.

File Auditing:

If a user edited a data then the auditor will monitor the user and report to the owner about the edited data. The group manager will monitor the changes in the file and if he finds any discrepancy auditor has full rights to revoke from his particular group. The public verifier can audit the integrity of shared data without retrieving the entire data from the cloud, even if some blocks in shared data have. On one hand, once a user is revoked from the group, the blocks signed by the revoked user can be efficiently resigned. More specifically, the proxy is able to convert a signature of Alice into a signature of Bob on the same block. Mean while, 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.

Group Sharing:

Data owner will store their data in the cloud and share the data among the group members. Who upload the data have rights to modify and download their data in the cloud. He can also set rights to other users in his group to edit or download data.

User Revocation:

If a user wishes to revoke from a group their request regarding revocation will be forwarded to the auditor where auditor will check to it and revoke the user from group. The user revocation is secure because only existing users are able to sign the blocks in shared data. Even with are-signing key, the cloud cannot generate a valid signature for an arbitrary block on behalf of an existing user. In addition, after being revoked from the group, a revoked user is no longer in the user list, and can no longer generate valid signatures on shared data.

IV. SYSTEM MODEL

A system model for the cloud storage architecture, which includes three main network entities: users, a cloud server, and a trusted third party.

- User: an individual or group entity, which owns its data stored in the cloud for online data storage and computing.

Different users may be affiliated with a common organization, and are assigned with independent authorities on certain data fields.

- Cloud server: an entity, which is managed by a particular cloud service provider or cloud application operator to provide data storage and computing services. The cloud server is regarded as an entity with unrestricted storage and computational resources.

- Trusted third party: an optional and neutral entity, which has advanced capabilities on behalf of the users, to perform data public auditing and dispute arbitration. In the cloud storage, a user remotely stores its data via online infrastructures, platforms, or software for cloud services, which are operated in the distributed, parallel, and cooperative modes. During cloud data accessing, the user autonomously interacts with the cloud server without external interferences, and is assigned with the full and independent authority on its own data fields. It is necessary to guarantee that the users' outsourced data cannot be unauthorized accessed by other users.

Admin login into system first

- Then create the groups
- Number of users register himself
- Admin add them in different user then admin upload the file
- Select the group to which file will share then key for group user get generated
- File uploaded then user login, using signature decrypted data then revoke user try to access file but he cannot for integrity admin send verification request to TPA then TPA verify data from Cloud Service provider.

An easy way to comply with formatting requirements is to use this document as a template and simply type your text into it.

V. CIPHER TEXT DATABASE

In cloud storage outsourcing environment, the outsourced data is usually encrypted database, which is usually implicitly assumed in the exiting academic research. Actually, our scheme could support the auditing of database of both plaintext and cipher text database. However, it is not straightforward to extend a scheme to support encrypted database. In order to achieve the confidentiality of the data record mx , the client can use his/her secret key to encrypt each mx using a encryption scheme. When there is only one user (data owner) in the group, the user only needs to choose a random secret key and encrypt the data using a secure symmetric encryption scheme. However, when the scheme needs to support multi- user data modification, while at the same time keeping the shared data encrypted, a shared secret key among group users will result in single point failure problem. It means that any group user (revoked or leave) leak the shared secret key will break the confidentiality guarantee of the data. To overcome the above problem, we need to adopt a scheme, which could support group users data modification. Luckily, Wu et al. [26] designed an Asymmetric Group Key Agreement scheme (ASGKA).The scheme has a nice property that, instead of a common secret key, only a shared encryption key is negotiated in an ASGKA protocol. Also, in the scheme, the public key can be simultaneously used to verify signatures and encrypt messages while any signature can be used to decrypt cipher text under this public key. Using the bilinear pairings, the authors instantiate a one-round ASGKA protocol tightly reduced to the decision Bilinear Diffie-Hellman Exponentiation (BDHE) assumption in the standard model. Thus, according to the ASGKA protocol, we consider the case of encrypted database (x, cx) , where x is an index and cx is the corresponding cipher value. We provide the detailed changes upon our scheme to support encrypted database.

1) In the Setup phase, the scheme has to run the key agreement of ASGKA for the group users. Then the database $DB = (i, mi)$ is encrypted by the group key gpk of data owner. Finally, the stored database is a cipher text database $DB = (i, ci)$.

2) In the second step of the Update phase, a group user firstly decrypts the record ci using the ASGKA secret key $gsk[*]$ to get plaintext database $DB = (i, mi)$. Then, update the data to $m'i$, and later encrypt the data with the public key gpk of ASGKA scheme to get the new encrypted database $DB = (i, c' i)$.

VI. CONCLUSION

In this paper, securely share the data file among the dynamic groups. Without revealing their identity members in the same group can share the data efficiently. Cryptography is used for over all security. When compared to other algorithm key size is very small, it is not able to hack easily. It is used for efficient revocation without updating private keys of remaining users. In future, concentrate on key management, how to revoke the private keys from the group members.

REFERENCES

- [1] C. Gentry and S. Halevi, "Implementing gentrys fully homomorphic encryption scheme," in Proc. of EUROCRYPT 2011, Tallinn, Estonia, May 2011, pp. 129–148.
- [2] S. Benabbas, R. Gennaro, and Y. Vahlis, "Verifiable delegation of computation over large datasets," in Proc. of CRYPTO 2011, CA, USA, Aug. 2011, pp. 111–131.
- [3] M. Backes, D. Fiore, and R. M. Reischuk, "Verifiable delegation of computation on outsourced data," in Proc. of ACM CCS 2013, Berlin, Germany, Nov. 2013, pp. 863–874.
- [4] D. Chaum and E. van Heyst, "Group signatures," in Proc. Of EUROCRYPT 1991, Brighton, UK, Apr. 1991, pp. 257–265.
- [5] E. Bresson and J. Stern, "Efficient revocation in group signatures," in Public-Key Cryptography - PKC 2001, Cheju Island, Korea, Feb. 2001, pp. 190–206.
- [6] J. Camenisch and A. Lysyanskaya, "Dynamic accumulators and application to efficient revocation of anonymous credentials," in Proc. of CRYPTO 2002, CA, USA, Aug. 2002, pp. 61–76.
- [7] G. Ateniese, D. Song, and G. Tsudik, "Quasi-efficient revocation in group signatures," in Proc. of FC 2002, Soughampton, Bermuda, Mar. 2002, pp. 183–197.
- [8] B. Wang, L. Baochun, and L. Hui, "Public auditing for shared data with efficient user revocation in the cloud," in Proc. Of IEEE INFOCOM 2013, Turin, Italy, Apr. 2013, pp. 2904–2912.
- [9] J. Yuan and S. Yu, "Efficient public integrity checking for cloud data sharing with multi-user modification," in Proc. of IEEE INFOCOM 2014, Toronto, Canada, Apr. 2014, pp. 2121–2129.
- [10] D. Catalano and D. Fiore, "Vector commitments and their applications," in Public-Key Cryptography - PKC 2013, Nara, Japan, Mar. 2013, pp. 55–72.