Group User Revocation and Integrity Auditing of Shared Data in Cloud Environment

Pushkar Zagade
Department of Computer Engineering
JSPM's JSCOE
Pune,
Maharashtra, India

Shruti Yadav
Department of Computer Engineering
JSPM's JSCOE
Pune,
Maharashtra, India

Aishwarya Shah
Departmentof Computer Engineering
JSPM's JSCOE
Pune,
Maharashtra, India

Ravindra Bachate
Department of Computer Engineering
JSPM's JSCOE
Pune,
Maharashtra, India

ABSTRACT
The enhancement of cloud computing make storage outsourcing becomes an exceeding trend, which result a secure data auditing a cool topic that emerge in research literature. Recently some researches consider the problem of efficient and secure public data authentication inspection for shared dynamic data. However, these schemes are still not secure against the collusion and leakage of cloud storage server from unauthorized attacker and revoked group users during user revocation in cloud storage system. In this paper, there will be auditing the integrity of shared data with dynamic groups in cloud. A new user can be added into the group and an existing group member can be revoked by preserving privacy including data backup based on vector commitment and verifier-local revocation group signature. This scheme supports the public validation and efficient user revocation and also some nice properties such as traceability, efficiency, confidently, countability. Finally, the security and experimental analysis show that our scheme is also secure and efficient.

General Terms
Vector commitment, deduplication, Asymmetric Group Key Agreement scheme (ASGKA)

Keywords
Public integrity auditing, dynamic data, cloud computing.

1. INTRODUCTION
The improvements and enhancements in cloud computing motivates organization as well as enterprises to outsource their data to third party cloud service providers (CSP's) which will result in improvements the data storage limitation of resource constrain local devices. In market, already some cloud storage services are available like simple storage service (S3) [1] on-line data backup services of Amazon and software like Google Drive, [2] Dropbox, [3] Mozy, [4] Bitcasa and [5] Memopal built for cloud application. In some cases cloud server sometime returns invalid results such as hardware/software failure, malicious attack and human maintenance. Security and privacy of cloud user's data should be protected by data integrity and accessibility. To overcome the security issues of today's cloud storage services, simple replication and protocols like Rabin's data dispersion scheme are not sufficient for practical application. For achieving the integrity and availability of remote cloud storage, some various solutions and their different variants have been proposed. In these solutions, when a scheme supports modification of data, it is known as dynamic scheme, otherwise static one. A scheme is *publicly verifiable* that means the integrity check of data can be performed not only by data owners, but also by the third party auditor (TPA). However, the focus of the dynamic scheme is on the cases where only and only data owner could modify the data of cloud.Recently, the development of cloud computing emerged some applications where the services of cloud can be used as a collaboration platform. In these software development environments, one or more than one (multiple) users in a group need to share source code as well as they needs to access, compile, modify and run the source code share by user at any time. The new model of cooperation network in cloud provides the infeasibility of data for auditing the remote data, where only the data owner can update its data. It will result in terrific communication and computation to the data owner which causes the single point of data owner. To achieve multiple data operation, Wang et al. put forth data integrity based on ring signature. In the scheme, it does not consider the user revocation problem and the cost of auditing is linear to the data size and group size. To further raise up the previous scheme and support group user revocation, Wang et al. proposed a scheme based on proxy re-signatures. However, this scheme assumes that authenticated and private channels exist between the pair of entities and there is no collusion among them. Also, cost of auditing the scheme is linear to the size of the group. Another attempt to improve the previous scheme and make the scheme scalable, efficient and collusion resistant, Yuan and Yu designed a dynamic public integrity auditing scheme with group user revocation. However, in their scheme, the authors do not consider the secrecy of data among the group users. That means, their scheme could efficiently support plain text of data update and integrity auditing, while not ciphertext data. In their scheme, if data owner shares group key among the users of group, revocation of any group user allow the group users to update their shared key. Also, the owner of the data does not take part in the user revocation phase, where the user revocation phase is itself conducted by the cloud. In this case, the malicious cloud server will result in collusion of revoked user and the cloud server where the cloud server could update data number of times as designed and provide a legal data finally. Due to above mentioned deficiency; we propose a construction which includes data encryption and decryption during the data modification processing, secure and efficient user revocation and also removal of redundant data. Here, vector commitment scheme will be applied over the database. Then we apply the Asymmetric Group Key Agreement (AGKA) and group signatures to support ciphertext database update among group users and efficient group user revocation respectively. The user in the group will be able to encrypt or decrypt a message from any other group users when the group users use the AGKA protocol to encrypt or decrypt the share

database. The collusion of the cloud and revoked group users will be prevented by the group signature.

2. LITERATURE SURVEY

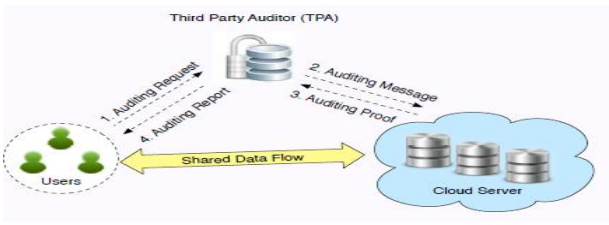
Boyang Wang, Baochun Li and Hui Li are the members of IEEE explore the concept of "Oruta: Privacy-Preserving Public Auditing for Shared data in the Cloud" in 2014. It shows that services of cloud provide not only data storage in commonplace, but also data sharing across multiple users. However, it remains an open challenge to audit the shared data by preserving identity privacy. This system proposed public auditing of shared data stored in cloud by using privacy preserving mechanism.

In particular, this paper exploits the concept of group signature which computes the verification information required for integrity auditing of shared data. With this mechanism, the signer identity of each block in shared data remains private from a third party auditor (TPA) which can publicly verify shared data integrity without accessing entire data. In extend this mechanism support batch auditing. This mechanism is responsible for auditing multiple shared data in just single auditing task. The high level comparison between Oruta and its relevant existing systems are shown in following *Table 1*. This paper represent first attempt towards designing effective public auditing of shared data in the cloud storage by preserving privacy.

TABLE 1
Comparison with Existing Mechanisms

	PDP [2]	WWRL [3]	Oruta
Public auditing	Yes	Yes	Yes
Data privacy	No	Yes	Yes
Identity privacy	No	No	Yes

As illustrated in Fig., the work in this paper involves three parties: the cloud server, the third party auditor (TPA) and users. There are two types of users in a group: the original user and a number of group users. The original user and group users are both members of the group. Group members are allowed to access and modify shared data created by the original user based on access control polices.



But this paper only shows that how to audit shared data integrity in a cloud with **static** group. It means groups of users already defined in cloud before shared data and membership of user is not changed during data sharing. The original user is responsible for deciding who is able to share its data before uploading data to the cloud. Another interesting problem is how to audit the integrity of shared data in the cloud with **dynamic** groups a new user can be added into the group and an existing group member can be revoked during data sharing while still preserving identity privacy. This will leave to future work.

3. PROPOSED SYSTEM

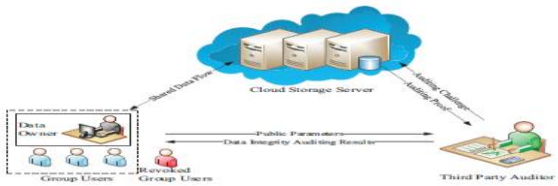
In this paper, we study the problem of constructing public authentication inspection for shared dynamic data with group user revocation. Our contributions are:

1) For cipher text database, we explore on the secure and efficient shared data integrate auditing for multi-user operation.

2) We intend an efficient data auditing scheme along with new features such as traceability and countability by incorporating the vector commitment primitives, asymmetric group key agreement and group signature.

3) The analysis results show that our scheme is secure and efficient as we provide the security and efficiency analysis of our scheme which will result in back-up and data storage in cloud.

4) The authorized duplicate check in the hybrid cloud architecture is supported by several deduplication constructions and this authorized duplicate check scheme comparatively incurs minimum overhead than normal operations.



3.1 Vector Commitment

Commitment is a fundamental primitive in cryptography and it plays an important role in security protocols such as voting, identification, zero-knowledge proof, etc. The hiding property of commitment requires that it should not reveal information of the committed message, and the binding property requires that the committing mechanism should not allow a sender to change his/her mind about the committed message.

Vector Commitment satisfies position binding that an adversary should not be able to open a commitment to two different values at the same position, and the Vector Commitment is concise, which means that the size of the commitment string and its openings have to be independent of the vector length.

Definition 1. A vector commitment is a collection of six polynomial-time algorithms *(VC.KeyGen, VC.Com, VC.Open, VC.Ver, VC.Update, VC.ProofUpdate)* such that: *VC.KeyGen*(1^k, *q).* Given the security parameter k and the size q of the committed vector (with q =poly(k)), the key generation outputs some public parameters pp.

$VC.Com_{pp}(m1, ...,mq)$. On input a sequence of q messages *m1, ...,mq* $\in M$ (M is the message space) and the public parameters pp, the committing algorithm outputs a commitment string C and an auxiliary information aux.

VC.Openpp(m, i, aux). This algorithm is run by the committee to produce a proof i that m is the i^{th} committed message. In particular, notice that in the case when some updates have occurred the auxiliary information aux can include the update information produced by these updates.

VC.Ver_{pp}*(C,m, i,*Λ_I*).*The verification algorithm accepts (i.e., it outputs 1) only if _i is a valid proof that C was created to a sequence m1, ... , mq such that m = mi.

VC. $Update_{pp}$ *(C,m,m′, i).* This algorithm is run by the committee who produces C and wants to update it by changing the i-th message to m′. The algorithm takes as input the old message m, the new message m′ and the position i. It outputs a new commitment C' together with an update information U.

VC. $ProofUpdate_{pp}$ *(C,_j ,m′, i,U).* This algorithm can be run by any user who holds a proof Λ_j for some message at position j w.r.t. C, and it allows the user to compute an updated proof Λ_j (and the updated commitment C′) such that Λ_j will be valid with regard to C′ which contains m′ as the new message at position i. Basically, the value U contains the update information which is needed to compute such values.The problem of verifiable data outsourcing is solved by the primitive of verifiable database with efficient update based on vector commitment. Recently, it was figured that the basic vector commitment scheme includes issues like forward automatic update attack and backward substitution update attack. To avoid these attacks and also to provide public verifiable for dynamic outsourced data, a new framework for verifiable database with efficient update from vector commitment was proposed.

3.2 Group Signature with User Revocation

We represent the formal definition of group signatures with valid user revocation as follows.

Definition 2. The signature scheme of authorized group user is a collection of three polynomial-time algorithms, which are VLR.KeyGen, VLR.Sign, and VLR.Verify behave as follows:

VLR.KeyGen(n). This randomized algorithm takes n parameter as input where n represent number of group user. Its output result in group public key(gpk), an n-element vector of user keys *gsk = (gsk(1),gsk(2),..,gsk(n)),* n-element vector of group user revocation tokens *grt=(grt(1),grt(2),…,grt(n)).*

VLR.Sign(gpk, gsk[i], M). This randomized algorithm takes group public key(gpk), a private key(gsk[i]) and a message M ϵ {0,1}* , and return user signature σ.

VLR.Verify(*gpk, RL, σ ,M*). The randomized verification algorithm takes group public key gpk, a set of revocation tokens RL, and a purported signature σ on a message M input as a parameter. It returns either valid or invalid result. It represent that σ is not a valid signature, or the user has been revoked who has generated it.

3.3 Supporting Ciphertext Database

In cloud storage environment, the outsourced data is usually stored in encrypted database, which was assumed in the previous research. Basically this scheme is designed to support auditing of both plaintext database and ciphertext database. However, this is not a simple approach to extend a scheme to support encrypted database.In order to achieve data confidentiality of record r_x , the user can use his/her secrete key to encrypt each record r_x using encryption scheme. When the group consist of only one user that is data owner, he/she need to choose a random secrete key and encrypt the data using secure encryption scheme. However, when the scheme needs to support multiuser data modification, it is difficult to keep the shared data encrypted, a shared secrete key among the number of group user will result in single point failure. It means there is chance of leakage of shared secrete key which break confidentiality of the shared data. To overcome this problem, we need to use scheme, which support multi-user group modification. Luckily, Wu et al. Designed an Asymmetric Group Key Agreement scheme (ASGKA). The scheme says, instead of a common secret key, only a shared encryption key is maintained 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 ciphertext under this public key.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.

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 Ciphertext database DB = (i, ci).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).

4. PROBLEM FORMULATION

In this section, we first describe the cloud storage model of our system and then we provide the threat model considered and also security goals we want to achieve.

4.1 Cloud Storage Model

As shown in the figure of cloud storage model there are three entities, namely the cloud storage server, a Third Party Auditor (TPA) and group users. Group user consists of a data owner and a number of users who are authorized to access and modify the data by the data owner. The data storage services for the group a user is provided by the cloud storage server which is semi trusted. The data integrity of the shared data store in the cloud server is conducted by a TPA. In our system the data owner could encrypt and upload its data to the remote cloud storage server. Also, the access and modify privileges is shared to a number of group users. Even if the data is frequently updated by the group users, the TPA efficiently verifies the integrity of the data stored in the cloud storage server. The owner of data is different from other group users. When a group user is found malicious or the contract of the user is expired, he/she could securely revoke a group user.

4.2 Threat Model and Security Goals

Our threat model considers two type of attack:

1. The knowledge of the pain text of the data may be obtained by an attacker outside the group (include the revoked group user cloud storage server). This kind of attacker has to at least break security of the adopted group data encryption scheme.
2. The cloud storage server colludes with the revoked group users and they want to provide an illegal data without being detected.

In the cloud environment we assume the cloud storage server is semi –trusted. Thus, here a malicious cloud server will be able to make data m, class modified by a user that needed to be a revoked in to a malicious data m'. The cloud could

make the malicious data m' and becomes valid in the user revocation process we aim to achieve the following security goals in our paper to overcome the problems mentioned above:

a) Security: The scheme should check the user authenticity by provided password to verify user identity and authentication. The scheme should satisfy privacy certifications by satisfying digital signature.

b) Efficiency: This scheme is efficient if for any data the computation as well as storage issues facilitated by any group user should not be dependent on the size of the shared data.

c) Countability: When the improper storage server of the cloud has tampered with the database, the TPA provides proof for this misbehavior to achieve countability.

d) Traceability: When the generation algorithm generates the data and when the group signature is valid, the data owner traces the last user who updates the shared data item.

e) Correctness: When any data updated by valid group user, this scheme efficiently supports encrypted database by providing the correct result.

4.3 MATHEMATICAL MODEL

$\psi(g2) = g1$, and $e : G1 \times G2 \rightarrow GT$ is a bilinear map with the following properties:

1. Computability: there exits an efficiently computable algorithm for computing mape;
2. Bilinearity: for all $u \in G1$, $v \in G2$ and $a, b \in Zp$, $e(ua, vb) = e(u, v)ab$;
3. Non-degeneracy: $e(g1, g2) \neq 1$.

Framework:
DB = (i,mi) for $1 \leq i \leq q$
key generation of verifier-local revocation to obtain the user keys and revocations (*gsk, grt)* ← *VLR.KeyGen(1k, n), where gsk* =(gsk[1], gsk[2]...gsk[n]) and an n element vector of user revocation tokens *grt.*
User Revocation (PK, i, τ):
VLR.Verify(gpk, RL, σ,M), M = C(t − 1),Ct, t.

5. CONCLUSION

The preserving of verifiable database with efficient and secure updates is an important way to solve the problem of verifiable data storage. We propose a scheme to realize secure and efficient auditing of data integrity for share dynamic data with multi-user modification. In this paper, the concept of authorized data deduplication was proposed to achieve the data security by including differential privileges of users in the duplicate check. The scheme vector commitment, Asymmetric Group Key Agreement (AGKA) and group signatures with user revocation are used to achieve the auditing remote data integrity. Beside the public data auditing, the combining of the three primitive enable our scheme to outsource ciphertext database to remote cloud and support secure group users revocation to shared dynamic data.

This paper involve the successful implementation of data backup and efficient storage for maintain the confidentiality of shared data. We provide Security analysis of our scheme, and it prove that our Scheme provide confidentiality of shared data for group users, Also, the performance analysis shows that our scheme also efficient in different phases as compare to its relevant schemes.

6. ACKNOWLEDGMENT

This work is supported by JSPM's Jayawantrao Sawant College of Engineering, Pune Maharashtra. We would like to express our gratitude towards **Prof. A. S. Devare** whose support and consideration has been a valuable asset during course of this paper. First and foremost, we would like to thank our guide **Prof. R. P. Bachate** providing us with their invaluable support, motivation, suggestion and guidance throughout the course of the paper. We convey our gratitude to our respected **HEAD OF DEPARTMENT, Prof. H. A. Hingoliwala** for his motivations and guidance throughout the work. And, last but not least we would like to thank **Principal Dr. M. G. Jadhav** for directly and indirectly help us for this work.

7. REFERENCES

[1] T. Jiang, X. Chen, and J. Ma, "Public integrity auditing for shared dynamic cloud data with group user revocation," in Proc. Of IEEE TRANSACTIONS ON COMPUTERS VOL: PP NO: 99 YEAR 2015

[2] Hugo E. Camacho, J. Alfredo Brambila, Alfredo Peña, José M. Vargas, "A cloud environment for backup and data storage," in Engineering Information Technology, Polytechnic University of Altamira, Altamira Tamaulipas, México

[3] Jin Li, Yan Kit Li, X. Chen, Patrick P. C. Lee, Wenjing Lou, "A Hybrid Cloud Approach for Secure Authorize Deduplication," in Proc. of IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEM VOL:PP NO:99 YEAR 2014

[4] Boyang Wang, Baochun Li, Member, IEEE, and Hui Li, Member, IEEE, " Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud" in Proc. Of IEEE TRANSACTIONS ON XXXXXX, VOL. X, NO. X, XXXX 201X

[5] C. Wang Student Member, IEEE, Sherman S.-M. Chow, Q. Wang, Student Member, IEEE, K. Ren, Member, IEEE, and W. Lou, Member, IEEE , "Privacy-Preserving Public Auditing for Secure Cloud Storage".

[6] B. Wang, B. Li, Member, IEEE, and H. Li, "Panda: Public Auditing for Shared Data with Efficient User Revocation in the Cloud" in Proc. Of IEEE TRANSACTIONS ON XXXXXX, VOL. X, NO. X, XXXX 201X

[7] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in Proc. of IEEE INFOCOM 2010, CA, USA, Mar. 2010, pp. 525– 533.

[8] D. Catalano and D. Fiore, "Vector commitments and their applications," in Public-Key Cryptography - PKC 2013, Nara, Japan, Mar. 2013, pp. 55–72.

[9] 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