Improving Data Integrity Auditing with Group user using Shared Dynamic Scheme in Cloud

Summiya Farheen
M. Tech Scholar
Department of Computer Science and Engineering
Don Bosco Institute of Technology
Bangalore, India

Mrs. Shruthi G
Assistant Professor
Department of Computer Science and Engineering
Don Bosco Institute of Technology
Bangalore, India

***Abstract* —The storage in cloud has become a rising trend these days that boosts the secure remote information auditing. Latterly, some analysis thought-about in shared dynamic information that there's a haul of secure and adequate public information integrity auditing. However there are schemes in sensible storage systems that don't seem to be secure against the connivance of cloud storage server and annul cluster user throughout user revocation. In this paper, based on the verifier-local revocation cluster signature, the connivance attack in existing scheme is work out associated equipped an adequate public integrity auditing scheme with secure cluster user revocation. The general public checking and adequate and secure cluster user revocation with the properties like traceability, with confidence, countability and adequacy are supported by this scheme.**

Keywords* — *Asymmetric Group Key Agreement, Cloud computing, Group signature, Dynamic data, Vector commitment

I. INTRODUCTION

The evolution of cloud computing motivates enterprises and organizations to source their information to third-party cloud service providers, which can improve the storage limitation of resource constrain native devices. Recently, some industrial cloud storage services, like the simple storage service on-line information backup services of Amazon and a few sensible cloud based software system such as Google Drive, Drop box, etc. are engineered for cloud application. Since the cloud servers could come back Associate in Nursing invalid end in some cases, like server hardware/software failure, human maintenance and malicious attack [6], new kinds of assurance of information integrity and accessibility square measure enquired to shield the safety and privacy of cloud user's data. to beat the on top of essential security challenge of today's cloud storage services, easy replication and protocols like Rabin's information dispersion scheme [1] square measure off from application.

For providing the integrity and handiness of remote cloud store, some solutions are projected. In these solutions, once a scheme supports information modification, we have a tendency to decision it dynamic scheme, otherwise static one A theme is publically verifiable implies that the information integrity check is performed not solely by data owners, however additionally by any third-party auditor. However, the dynamic schemes higher than target the cases wherever there's an information owner and solely the info owner might modify the info. In these software package development environments, multiple users in a very cluster got to share the source code, and that they got to access, modify, compile and run the shared source code at any time and place. The new cooperation network model in cloud makes the remote information auditing schemes become impracticable, wherever solely the information owner will update its data.

To support multiple user information operation, Wang et al. [10] projected an information integrity supported ring signature. Within the scheme, the user revocation drawback isn't thought of and therefore the auditing price is linear to the cluster size and information size. To any enhance the previous theme and support cluster user revocation, Wang et al. [11] designed a scheme supported proxy re-signatures. The authors designed polynomial authentication tags and adopt proxy tag update techniques in their scheme that build their scheme support public checking and economical user revocation. However, in their scheme, the authors don't take into account the info secrecy of cluster users. It implies that, their scheme might expeditiously support plaintext information update and integrity auditing, whereas not cipher text information. In their scheme, if the info owner trivially shares a cluster key among the group users, the defection or revocation any cluster user can force the cluster users to update their shared key. Also, the info owner doesn't participate within the user revocation part, wherever the cloud itself might conduct the user revocation part. During this case, the collusion of revoked user and therefore the cloud server can provide probability to malicious cloud server wherever the cloud server might update the information as several time as designed and supply a legal data finally. To the most effective of our data, there's still no answer for the higher than drawback publicly integrity auditing with cluster user modification.

II. RELATED WORK

Many researchers concentrated on the problems on how to securely outsource local store to remote cloud server. Among which the problem of remote information integrity and accessibility auditing attacks the attestation of many researchers. The concepts and solution provable information possession and proofs of retrievability were first proposed by Ateniese and Juels [10] to improve the efficiency and enhance the function of basic schemes, such

as allowing public auditing and supporting data update. Yuan and Yu designed a dynamic public integrity auditing scheme with secure group user revocation. The scheme is based on polynomial authentication tags and adopts proxy tag update techniques, which makes their scheme support public checking and efficient user revocation.

Group signature is introduced by Chaum and Heyst[12] which provides anonymity for signers, wherever every cluster member includes a personal key that enables the user to sign messages. However, the resulting signature keeps the identity of the signer secret. Usually, there is a third party that can conduct the signature anonymity using a special trapdoor. Wang designed a scheme to support share knowledge integrity auditing, that shield the privacy of users exploitation ring signature. However it will no support dynamic cluster and additionally suffers from a process overhead linear to the cluster size and therefore the range of information auditing.

To support user revocation he designed another scheme supported the belief that no collusion happens between cloud servers and revoked user. Gennaro[11] formalized the notion of verifiable computation that permits a consumer to source the computation of associate degree arbitrary perform. The disadvantage here is predicated on cryptography, the consumer should repeat the expensive pre-processing stage if the malicious server tries to cheat and learn a little information. Catalano and Fiore[2] planned a sensible answer to create verifiable database from vector commitment that supports the general public verifiability. It assumes that the scale of the outsourced information ought to be fastened and therefore the consumer will apprehend the outsourcing perform ahead.

III. PROBLEM FORMULATION

A. Cloud Storage Model

There are three entities within the cloud storage model as shown in Figure 1, specifically the cloud storage server, cluster users and a third party Auditor. The cloud storage server is semi-trusted, who provides information storage services for the cluster users. Group users include a data owner and variety of users who are licensed to access and modify the information by the data owner. The third party Auditor will ready to conduct the data integrity of the shared data hold on within the cloud server. In this system, the data owner will do the subsequent:

1) he/she could encrypt and transfer its information to the remote cloud storage server.

2) he/she can shares the privilege like access and modify and may compile and execute if necessary to variety of cluster users.

Even the information is usually updated by the cluster users; the integrity of the information hold on within the cloud storage server is verified by the third party Auditor. If the cluster user is found malicious or the contract of the user is terminated, the cluster users are often revoked by the data owner.

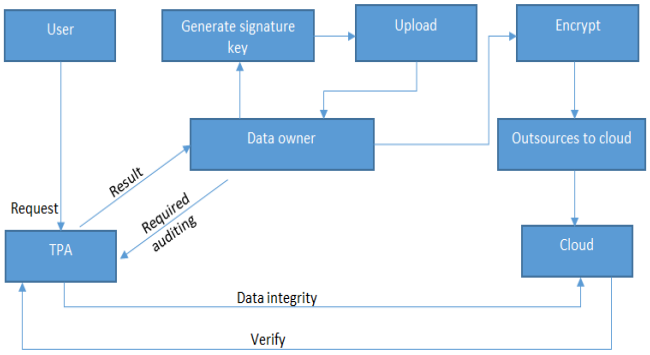

Fig 1. Cloud Storage Model

B. Threat Model and Security Goals

Two forms of attacks are considered within the threat model:

1) Outside the cluster an assaulter as well as the revoked cluster user might get some information in a very plaintext. This kind of assaulter must a minimum of break the protection of the adopted cluster data encryption scheme.

2) The cloud storage server colludes with the revoked cluster users, and that they need to supply illegal information while not being detected.

To overcome the issues above, the subsequent security goals are achieved:

1) Security

A scheme is secure if for any information and any probabilistic polynomial time adversary, the adversary cannot convince a verifier to just accept an invalid output.

2) Correctness

A scheme is correct if for any information and for any updated information m by a legitimate cluster user, the output of the verification by an honest cloud storage server is usually the value m. Here, m may be a ciphertext if the scheme may efficiently support encrypted information.

3) Efficiency

A scheme is efficient if for any information, the computation and storage overhead invested by any client user should be freelance of the size of the shared information.

4) Countability

A scheme is enumerable, if for any information the TPA will give a proof for this wrongful conduct, once the dishonest cloud storage server has tampered with the information.

IV. PROPOSED SCHEMES

In propose technique a construction which not solely supports cluster encoding and coding during the data modification processing, but also realizes efficient and secure user revocation. Our plan is to use vector commitment scheme over the database. Then we leverage the uneven cluster Key Agreement and cluster signatures to support ciphertext data base update among group users and economical cluster user revocation severally.

Specifically, the group users use the uneven cluster Key Agreement protocol to encrypt/decrypt the share info, which is able to guarantee that a user in the group will be able to encrypt/decrypt a message from any other group users. The group signature will prevent the collusion of cloud and revoked group users, where the data owner can participate within the user revocation part and the cloud could not revoke the data that last modified by the revoked user. The security of the scheme depends on the Strong Diffie- Hellman assumption and the Decision Linear assumption.

A. Vector Commitment

It is a basic primitive in cryptography and it plays a vital role in security protocols. The formal definition of Vector Commitment [2] as follows:
A vector commitment scheme may be assortment of six polynomial-time algorithms: VT.KeyGenn, VT.Comm, VT.Openn, VT.Verf, VT.Upd, VT.ProofUpd such that:

1) VT.KeyGenn(1^c, x)
Given the protection parameter c and therefore the size x of the committed vector (with x =poly(c)), the key generation outputs some public parameters p.

2) VT.$Comm_p(y_1, ...,y_x)$
On input a sequence of x messages $y_1,...,y_x \in N$ and therefore the public parameters p, the committing algorithmic program outputs a commitment string S and an auxiliary data auxi.

3) VT.$Openn_p$(y, a, auxi)
This algorithmic program is pass by the committer to provide a proof a that y is the a-th committed message. Within the case once some updates have occurred the auxiliary data auxi will embody the update data created by these updates.

4) VT.$Verf_p(S, y, a, n_a)$
The verification algorithmic program accepts its output as one only if n_a is a valid proof that S was created to a sequence $y_1, ...,y_x$ such that $y = y_a$.

5) VT.$Upd_p(S, y, y', a)$
This algorithmic program is pass by the committer who produces S and needs to update it by ever-changing the a-th message to y′. The algorithmic program takes as input the previous message y, the new message y′ and therefore the position a. It outputs a new commitment S' along with an update data I.

6) VT.$ProofUpd_p(S, n_b, y', a, I)$
This algorithmic program are often pass by any user who holds a proof n_b for few message at position b w.r.t. S, and it permits the user to compute an updated proof n'_b such that n'_b is valid with relevance S′ that contains y′ because the new message at position a.

B. Group Signature with User Revocation

A verifier-local group signature scheme is a collection of three polynomial-time algorithms: VR.KeyGenn, VR.Sgn, VR.Vrfy, which behaves as follows:

1) VR.KeyGenn(z*)*
This takes as input a parameter z, the number of members of the group. It outputs a group public key gk, an z-element vector of user keys gk =(gk[1], gk[2], ..., gk[z]), and an z-element vector of user revocation tokens rt.

2) VR.Sgn(gk, gnk[i],N)
This takes as input the group public key gk, a private key gnk[a], and a message $N \in \{0, 1\}*$, and returns a signature σ.

3) VR.Vrfy(gk,RL, σ,M).
The verification algorithm takes as input the group public key gk, a set of revocation tokens rvt and a purported signature σ on a message N. It returns either valid or invalid. The latter response can mean either that σ is not a valid signature, or that the user who generated it has been revoked.

C. New Framework

The proposed framework of our public integrity auditing for shared dynamic cloud data with secure group user revocation is given as follows:
1) Setup(1^c,DB):
Data owner shares the database with a group of z users.

a) To obtain the public parameters the data owner runs the key generation algorithm of vector commitment.

b) To obtain the user keys and revocations run the key generation of verifier-local revocation.

c) To compute commitment and auxiliary information run the computing algorithm.

d. Run the signing algorithm over the commitment S.

e) Compute and output a signature σt ← VR.Sgn(gk, gnk[s], {S(t − 1),St, t}) for the t-th time the group user whose secret key is gnk[s].

2) Query(PK, P, auxi, DB, a):
A group user run the opening algorithm to compute a proof na ← Vt.Opennp(Sa, a, auxi).

3) Verify(PK,RL, i, τ):
Parse τ = (sa, na,n(t)). If the signature is valid after running the algorithm VR.Vrfy(gk,RL,_(T)). Then, run the verification algorithm of vector commitment {0, 1} ← VT.Verrp(S(t), σt, sa, a,na). The algorithm accepts when it output 1, return an error ⊥ otherwise.

4) Update(a, τ):
a) A group user first queries and verifies the database to make certain this information is valid. Additional exactly, the group user obtain τ ←Query(PK, PP, auxi,DB, a) and check that Verify(PK, a, τ) = ya.

b) Run the update algorithm over the new data and output the updated commitment and the update information (S′,I) ← Vt.Update(S,y,y′, a).

5) ProofUpdate(S,n_b,c′a,a,I):

a) A third part auditor can first verify that, compared with the stored counter t, the latest counter equals t + 1. Then, run the proof of update algorithm of vector commitment to compute an update proof nb ←VT.ProofUpdp(S,nb ,y′a, a,I)

b) Verify the commitment S′, and its corresponding proof n_a is also valid over message y′a.

6) UserRevocation(PK, a, τ):

The third part auditor can run the verification algorithm of verifier-local revocation and return either valid or invalid {0, 1} ←VLR.Verify(gpk,RL, σ,N).

V. PERFORMANCE EVALUATION

This provides both the numerical and also the experimental analysis, and conducts the computation time value comparison.

A. Numerical Analysis

In the Setup part the 3 schemes need one-time costly procedure effort. The scheme is secure against the collusion attack of the cloud storage server and also the revoked users [10]; the costly computation overhead is outsourced to the cloud storage server. The cloud storage server stores all the information and its relevant materials. The cluster users don't need storing any information locally except some non-public key materials.

In this scheme five algorithms are used they are Query algorithm, Verify algorithm, Update algorithm, ProofUpdate algorithm and UserRevocation algorithm.

1) Query Algorithm:

The information will increase with the computation overhead and also the server doesn't have to be compelled to figure the proof anytime since it's identical for identical data item. The server solely has to figure once for the primary query on every index. Therefore the procedure cost gets reduced once the server adopts some storage overhead.

2) Verify algorithm:

This brings rather more computation overhead since the scheme [10] adopt the delegation technology for information change. The group signature scheme is adopted to forestall the attack against the collusion of the malicious and revoked cluster users. This algorithm brings rather more process overhead than scheme [10].

3) Update

With the rise of data parts the computation value grows.

4) ProofUpdate algorithms:

The computation value grows with the rise of information parts.

5) UserRevocation algorithm:

All the computation time value grows with the rise of the challenge blocks (items) range. It is efficient as a result of the computation time value won't grow with the rise of the cluster users. Thus, it provides constant computation overhead with totally different cluster size.

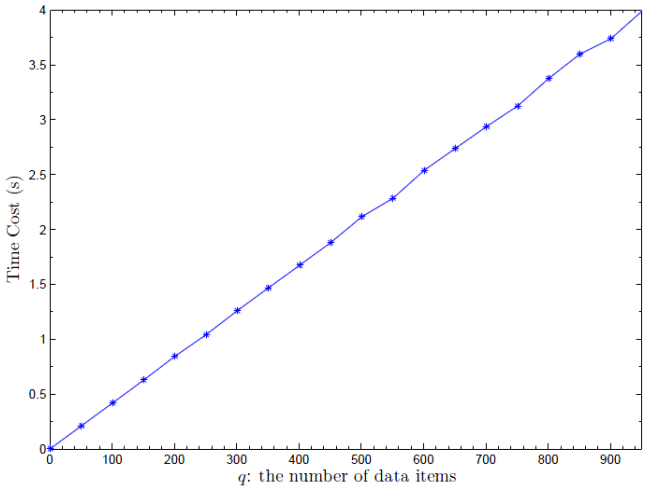

Figure 2. Query Time Cost

B. Experimental evaluation

To evaluate exactly the computation complexness at completely different entities, all the entities are simulated on this machine. The query time value is linear with the information things number q, which is able to take close to four seconds to query about a thousand information things as shown within the figure 2. The server doesn't have to run the complete query algorithm each time. First the integrity of the signature is to be verified, which implies that it has to generate the time value parameters.

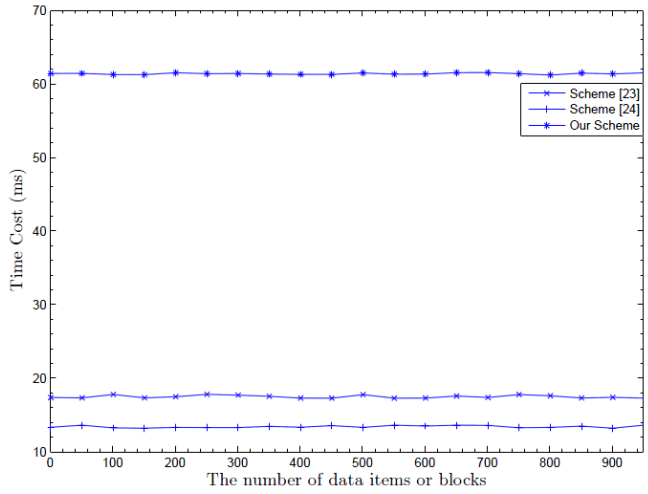

Figure 3. Time Cost Verification

The data update computation comparison is shown within the figure 4 and also the computation time value grows with the rise of the elements number of every block and additionally the number; and therefore the simulation of UserRevocation algorithm. Computation overhead grows quickly with the rise of cluster users number and also the elite challenging blocks number. They achieve this by permitting the cloud storage server to recompute the authentication tag of blocks last changed by a revoked cluster user. It tends to analyze this tag update delegation means within the previous section and suggests that it's not

secure against the cloud storage server and revoked cluster users collusion attack.

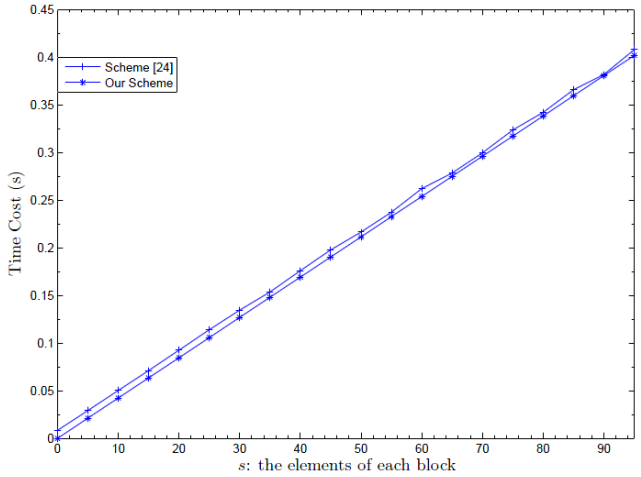

Figure 4. Time Cost Updation

VI. CONCLUSION

A scheme is proposed to understand efficient and secure data integrity auditing for share dynamic data with multi-user modification. To solve the matter of verifiable outsourcing of storage verifiable information with efficient updates is a vital method. Few schemes are adopted like vector commitment, asymmetric group Key Agreement and group signatures with user revocation to attain the information integrity auditing of remote data. The combining of the three primitive enable outsource ciphertext information to remote cloud and support secure group users revocation to shared dynamic information. Security is provided against the collusion attack from the cloud storage server and revoked cluster users at the side of the information confidentiality for group users.

REFERENCES

[1] M. Rabin,"Efficient dispersal of information for security," Journal of the ACM (JACM), vol. 36(2), pp. 335–348, Apr. 1989.

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

[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] C. Gentry, "Fully homomorphic encryption using ideal lattices," in Proc. of ACM STOC 2009, Washington DC, USA, May 2009, pp. 169–178.

[5] 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.

[6] M. A., "Above the clouds: A berkeley view of cloud computing," Tech. Rep. UCBEECS, vol. 28, pp. 1–23, Feb. 2009.

[7] J. G. (2006) The expanding digital universe: A forecast of worldwide information growth through 2010. IDC. [Online].Available: Whitepaper.

[8] K. D. Bowers, A. Juels, and A. Oprea, "Proofs of retrievability: theory and implementation," in *Proc. of CCSW 2009*, llinois, USA, Nov. 2009, pp. 43–54.

[9] R. Gennaro, C. Gentry, and B. Parno, "Non-interactive verifiable computing: Outsourcing computation to untrusted workers," in *Proc. of CRYPTO 2010*, CA, USA, Sep. 2010, pp. 465–482.

[10] B. Wang, B. Li, and H. Li, "Oruta: Privacy-preserving public auditing for shared data in the cloud," in Proc. of IEEE CLOUD 2012, Hawaii, USA, Jun. 2012, pp. 295–302.

[11] A. Juels and B. S. Kaliski, "Pors: Proofs of retrievability for large files," in *Proc. of ACM CCS*, Virginia, USA, Oct. 2007, pp. 584–597.

[12] 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.