International Journal of Advance Research in Science and Engineering
Volume No.07, Special Issue No.01, March 2018
www.ijarse.com
IJARSE
ISSN: 2319-8354

A SECURE ANTI-COLLUSION DATA SHARING SCHEME FOR DYNAMIC GROUPS IN THE CLOUD

G.M.Viji[1], Abarna.P[2], Malini.M[3] , Shobana.J[4]

AP/CSE , Final year IT Students

Indira Gandhi College Of Engineering & Technology For Women, Chengalpattu

ABSTRACT

A secure way for key distribution without any secure communication channels, and the users can securely obtain their private keys from group manager. It can achieve fine-grained access control, any user in the group can use the source in the cloud and revoked users cannot access the cloud again after they are revoked. It can protect the scheme from collusion attack, which means that revoked users cannot get the original data file even if they conspire with the untrusted cloud. In our approach, by leveraging polynomial function, we can achieve a secure user revocation scheme. It can achieve fine efficiency, which means previous users need not to update their private keys for the situation either a new user joins in the group or a user is revoked from the group.

keywords— Public integrity auditing, dynamic data, Victor commitment.

I.INTRODUCTION

The development of cloud computing motivates enterprises and organizations to outsource their data to third-party cloud service providers , which will improve the storage limitation of resource constrain local devices. Recently, some commercial cloud storage services, such as the simple storage service on-line data backup services of Amazon and some practical cloud based software Google Drive , Dropbox , Mozy, Bitcasa, and Memopal , have been built for cloud application.

For providing the integrity and availability of remote cloud store, some solutions and their variants have been proposed. In these solutions, when a scheme supports data modification, we call it dynamic scheme, otherwise static one. A scheme is publicly verifiable means that the data integrity check can be performed not only by data owners, but also by any third-party auditor. However, the dynamic schemes above focus on the cases where there is a data owner and only the data owner could modify the data. The deficiency of above schemes motivates us to explore how to design an efficient and reliable scheme, while achieving secure group user revocation.To the end, we propose a construction which not only supports group data encryption and decryption during the data modification processing, but also realizes efficient and secure user revocation.

Our idea is to apply vector commitment scheme over the database. Then we leverage the Asymmetric Group Key Agreement (AGKA) and group signatures to support ciphertext data base update among group users and efficient group user revocation respectively.Specifically, the group user use the AGKA protocol to

encrypt/decrypt the share database, which will 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 will take part in the user revocation phase and the cloud could not revoke the data that last modified by the revoked user.

i) OUR CONTRIBUTION

In this paper, we further study the problem of construing public integrity auditing for shared dynamic

data with group user revocation. Our contributions are three folds:

1) We explore on the secure and efficient shared data integrate auditing for multi-user operation for ciphertext database.

2) By incorporating the primitives of victor commitment, asymmetric group key agreement and group signature, we propose an efficient data auditing scheme while at the same time providing some new features, such as traceability and countability.

3) It provides the security and efficiency analysis of our scheme, and the analysis results show that our scheme is secure and efficient.

ii) CLOUD STORAGE MODEL

Group users consist of a data owner and a number of users who are authorized to access and modify the data by the data owner. The cloud storage server is semi-trusted, who provides data storage services for the group users. TPA could be any entity in the cloud, which will be able to conduct the data integrity of the shared data stored in the cloud server. In our system, the data owner could encrypt and upload its data to the remote cloud storage server. Also, he/she shares the privilege such as access and modify (compile and execute if necessary) to a number of group users.

iii) THREAT MODEL AND SECURITY GOALS

Our threat model considers two types of attack:

1) An attacker outside the group may obtain some knowledge of the plaintext of the data. Actually, this kind of attacker has to at lease break the security of the adopted group data encryption scheme.

2) The cloud storage server colludes with the revoked group users, and they want to provide a illegal data without being detected.

The main contributions of this work can be summarized as follows.

1) A scheme is secure if for any database and any probabilistic polynomial time adversary, the adversary cannot convince a verifier to accept an invalid output. A scheme is correct if for any database and for any updated data m by a valid group user, the output of the verification by an honest cloud storage server is always the value m. Here, m is a cipher text if the scheme could efficiently support encrypted database.

2) A scheme is efficient if for any data, the computation and storage overhead invested by any client user must be independent of the size of the shared data. A scheme is countable, if for any data the TPA can provide a proof for this misbehavior, when the dishonest cloud storage server has tampered with the database.

3) We require that the data owner is able to trace the last user who update the data ,when the data is generated by the generation algorithm and every signature generated by the user is valid.

II.PRELIMINARIES AND DEFINITIONS

Our scheme makes use of bilinear groups. The security of the scheme depends on the Strong Diffie-Hellman assumption and the Decision Linear assumption. In this section, we review the definitions of bilinear groups and the complexity assumption.

A. Bilinear Groups:

Let G1 and G2 be two multiplicative cyclic groups of prime order p, g1 is a generator of G1 and g2 is a generator of G2. ψ is an efficiently computable isomorphism from G2 to G1 with $\psi(g2) = g1$, 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 map e;

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:** $e(g1, g2) 6= 1$.

B. Complexity Assumption:

The security of our scheme relies on the difficulty of some problems: the Strong Diffie-Hellman problem, the Decision Linear problem, and the Computational Diffie-Hellman problem. We describe these problems as follows.

Definition 1. Q-Strong Diffie-Hellman problem. Let G1, G2 be cyclic group of prime order p, where possibly G1 = G2. Let g1 be a generator of G1 and g2 be a generator of G2. Given a $(q + 2)$ −tuple$(g1, g2, g 2 , g(2)2 , ..., g(q)$ as input, output a pair $(g1/(+x) 1 , x)$ where $x \in Z* p$.The assumption could be used to construct short signature scheme without random oracles . The assumption has properties similar to the Strong-RSA assumption and the properties are adopted for building short group signature in our scheme.

C.Vector Commitment:

A vector commitment scheme is a collection of six polynomial-time algorithms (VC.KeyGen, VC.Com, VC.Open, VC.Ver,VC.Update, VC.ProofUpdate) such that:

VC.KeyGen(1k, 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.Compp(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 committer 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.Verpp(C,m, i,_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.Updatepp(C,m,m′, i). This algorithm is run by the committer 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.ProofUpdatepp(C,_j ,m′, i,U). This algorithm can be run by any user who holds a proof _j for some message at position j w.r.t. C, and it allows the user to compute an updated proof _′ j (and the updated commitment C′) such that _′ j will be valid with regard to C′ which contains m′ as the new message at position i.

D.Group Signature with User Revocation:

Definition 2. A verifier-local group signature scheme is a collection of three polynomial-time algorithms (VLR.KeyGen, VLR.Sign, VLR.Verify), which behaves as follows:VLR.KeyGen(n). This randomized algorithm takes as input a parameter n, the number of members of the group. It outputs a group public key gpk, an n-element vector of user keys gsk = (gsk[1], gsk[2], ..., gsk[n]), and an n-element vector of user revocation tokens grt, similarly indexed. VLR.Sign(gpk, gsk[i],M). This randomized algorithm takes as input the group public key gpk, a private key gsk[i], and a message $M \in \{0, 1\}*$, and returns a signature σ. VLR.Verify(gpk,RL, σ,M). The verification algorithm takes as input the group public key gpk, a set of revocation tokens RL (whose elements form a subset of the elements of grt), and a purported signature σ on a message M.

III. SYSTEM MODEL

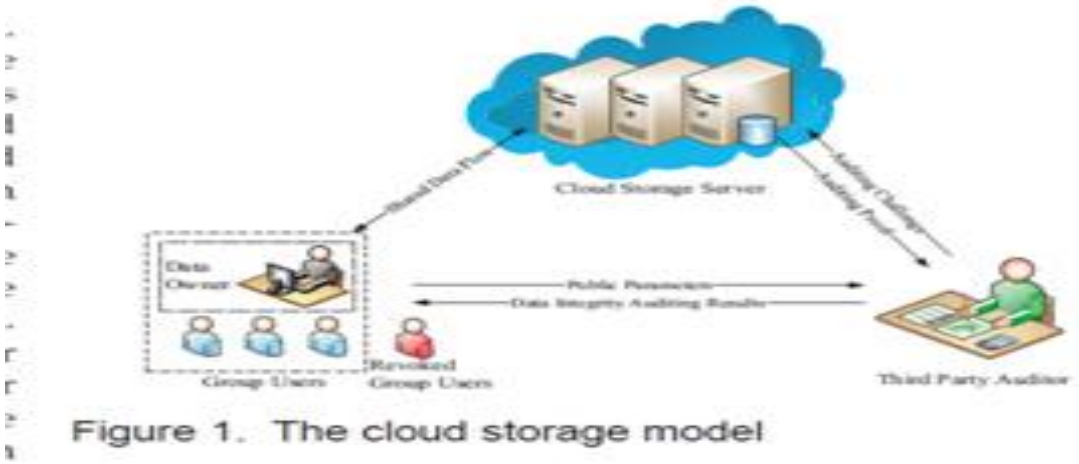

Figure 1. The cloud storage model

In the cloud storage model as shown in Figure 1, there are three entities, namely the cloud storage server,group users and a Third Part Auditor (TPA).Group users consist of a data owner and a number of users who are authorized to access and modify the data by the data owner. The cloud storage server is semi-trusted, who provides data storage services for the group users. TPA could be any entity in the cloud,which will be able to conduct the data integrity of the shared data stored in the cloud server. In our system,the data owner could encrypt and upload its data to the remote cloud storage server. Also, he/she shares the privilege such as access and modif to a number of group users.The TPA could efficiently verify the integrity of the data stored in the cloud storage server, even the data is frequently updated by the group users. The data owner is different from the other group users, he/she could securely revoke a group user when a group
user is found malicious or the contract of the user is expired.

It is reasonable that a revoked user will collude with the cloud server and share its secret group key to the cloud storage server. In this case, although the server proxy group user revocation way brings much communication and computation cost saving, it will make the scheme insecure against a malicious cloud
storage server who can get the secret key of revoked users during the user revocation phase. Thus, a malicious cloud server will be able to make data m, last modified by a user that needed to be revoked, into a malicious data m′. In the user revocation process, the cloud could make the malicious data m′ become valid.

- **Threat Model and Security Goals** is to achieve the following security goals in our paper:

1) ***Security.*** A scheme is secure if for any database and any probabilistic polynomial time adversary, the adversary cannot convince a verifier to accept an invalid output.

2) ***Correctness***. A scheme is correct if for any database and for any updated data m by a valid group user, the output of the verification by an honest cloud storage server is always the value m. Here, m is a ciphertext if the scheme could efficiently support encrypted database.

3) ***Efficiency***. A scheme is efficient if for any data, the computation and storage overhead invested by any client user must be independent of the size of the shared data.

4)Countability. A scheme is countable, if for any data the TPA can provide a proof for this misbehavior,when the dishonest cloud storage server has tampered with the database.

5) ***Traceability***. We require that the data owner is able to trace the last user who update the data (data item), when the data is generated by the generation algorithm and every signature generated by the user is valid.

IV. OUR PROPOSED EPOC SCHEME

Our scheme consists of five phases, namely **New Framework, A Concrete Scheme, Supporting Cipher text Database, and Probabilistic Detection.**

1) New Framework: A public integrity auditing scheme with updates allows a resource-constrained client to

outsource the storage of a very large database to a remote server. Later, the client can retrieve and update the database records stored in the server and publicly audit the integrity of the updated data.

2) A Concrete Scheme: Let k be a security parameter and DB = (i,mi) for $1 \leq i \leq q$ be the database. The database DB = (i,mi) is shared by a group of n users with only one data owner. The message space is M= Zp.Let G,GT be two bilinear groups of prime order p equipped with a bilinear map $e : G \times G \rightarrow GT$, and g be a random generator of G. Randomly choose z1, ..., zq $\leftarrow_R$ Zp. For all i = 1, ..., q, set hi = gzi . For all i, j = 1, ..., q, i 6= j, set hi,j = gzizj . The data owner runs the key generation algorithm of vector commitment VC.KeyGen(1k, q) to obtain the public parameters PP = (p, q,G,GT ,H, g, ({hi})i∈[q], {hi,j}i,j∈[q],i6=j) and the message space M= Zp. By using a collision resistant hash function $H : \{0, 1\}* \rightarrow Zp$, our scheme can be easily extended to support arbitrary messages in $\{0, 1\}*$.

3) Supporting 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 ciphertext database. However, it is not traightforward to extend a scheme to support encrypted database.

4) Probabilistic Detection*:* The position binding property of vector commitment of the scheme allows the cloud storage server to prove the data item correctness of certain position. The result is interesting that when y is a fraction of the total item number q, the detection probability of server misbehavior is a constant amount of item. For example, if y = 1% of q, then the third part auditor asks for 460 blocks and 300 blocks in order to achieve

the detection probability of at least 99% and 95%,respectively.

V.CONCLUSION

The scheme vector commitment, Asymmetric Group Key Agreement (AGKA) and group signatures with user revocation are adopt to achieve the data integrity auditing of remote data. 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. We provide security analysis of our scheme, and it shows that our scheme provide data confidentiality for group users, and it is also secure against the collusion attack from the cloud storage server and revoked group users. Also, the performance analysis shows that, compared with its relevant schemes, our scheme is also efficient in different phases.

REFERENCES

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

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

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

[4] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner,Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in *Proc. of ACM CCS*, Virginia, USA, Oct. 2007,pp. 598–609.

[5] 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.

[6] 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.

[7] Y. Dodis, S. Vadhan, and D. Wichs, "Proofs of retrievability via hardness amplification," in *Proc. of TCC 2009*, CA, USA,Mar. 2009, pp. 109–127.

[8] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Proofs of retrievability via hardness amplification," in *Proc. of ESORICS 2009*, Saint-Malo, France, Sep. 2009, pp. 355–370.