International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 06 | June - 2022 Impact Factor: 7.185 ISSN: 2582-3930

AN ENHANCEMENT OF DATA SHARING IN GROUP AND STORAGE WITH HIGH SECURITY AND EFFICIENCY IN CLOUD COMPUTING

Author 1: Mrs.Shubangini Patil
E-mail :shubha.kanni@gmail.com

Author 2: Mr.Mallinath Swamy
E-mail:mallinathswamy@yahoo.in

Author 3: Mrs.Shantkumari M
E-mail:shantamachetty@gmail.com

Author 4: Mrs.Laxmi Math
E-mail:laxmi.math@gmail.com

ABSTRACT

Group data sharing in cloud environments has become a hot topic in recent decades. With the popularity of cloud computing, how to achieve secure and efficient data sharing in cloud environments is an urgent problem to be solved. In addition, how to achieve both anonymity and traceability is also a challenge in the cloud for data sharing. This paper focuses on enabling data sharing and storage for the same group in the cloud with high security and efficiency in an anonymous manner. By leveraging the key agreement and group signature, a novel traceable group data sharing scheme is proposed to support anonymous multiple users in public clouds. On the one hand, group members can communicate anonymously with respect to the group signature, and the real identities of members can be traced if necessary. On the other hand, a common conference key is derived based on the key agreement to enable group members to share and store their data securely. Note that a symmetric balanced incomplete block design is utilized for key generation, which substantially reduces the burden on members to derive a common conference key. Both theoretical and experimental analyses demonstrate that the proposed scheme is secure and efficient for group data sharing in cloud computing.

INTRODUCTION

Compared with the traditional information sharing and communication technology, cloud computing has attracted the interest of most researchers because of its low energy consumption and resource sharing characteristics. Cloud computing can not only provide users with apparently limitless computing resources but also provide users with apparently limitless storage resources .Cloud storage is one of the most important services in cloud computing,which enables the interconnection of all types of electronic products. Moreover, various forms of data information can freely flow with respect to the cloud storage service, for instance, social networks, video editing, and home networks. However, little attention has been given to group data sharing in the cloud, which refers to the situation in which multiple users want to achieve information sharing in a group manner for cooperative purposes .Group data sharing has many practical applications, such as electronic health networks, wireless body area networks, and electronic literature in libraries. There are two ways to share data in cloud storage.The first is a one-to-many pattern, which refers to the scenario where one client authorizes access to his/her data for many clients. The second is a many-to-many pattern, which refers to a situation in which many clients in the same group authorize access to their data for many clients at the same time.Consider the following real-life scenario: in a research group at a scientific research institution, each member wants to share their results and discoveries with their team members. In this case, members on the same team are able to access all team's results (e.g., innovative ideas, research results, and experimental data). However, the maintenance and challenges caused by the local storage increase the difficulty and workload of information sharing in the group. Outsourcing data or time-consuming computational workload to the cloud solves the problems of maintenance and challenges caused by local storage and reduces the redundancy of data information, which reduces the burden on enterprises, academic institutions or even individuals. However, due to the unreliability of the cloud, the outsourced data are prone to be leaked and tampered with. In many cases, users have only relatively low control in the cloud service and cannot guarantee the security of the stored data. In addition, in some cases, the user would prefer to anonymously achieve data sharing in the cloud .Our goal is to achieve anonymous data sharing under a cloud computing environment in a group manner with high security and efficiency. To achieve this goal, the following challenging problems should be taken into consideration.

LITERATURE SURVEY

[1] J. Yu, K. Ren, C. Wang, and V. Varadharajan Cloud storage auditing is viewing important service.

Cloud storage auditing is viewed as an important service to verify the integrity of the data in the public cloud. Current auditing protocols are all based on the assumption that the client's secret key for auditing is absolutely secure. However, such an assumption may not always be held, due to the possibly weak sense of security and/or low security settings at the client. If such a secret key for auditing is exposed, most of the current auditing protocols would inevitably become unable to work. In this paper, we focus on this new aspect of cloud storage auditing. We investigate how to reduce the damage of the client's key exposure in cloud storage auditing and give the first practical solution for this new problem setting. The security proof and performance analysis show that our proposed protocol is secure and efficient.

[2] X. Chen, J. Li, X. Huang, J. Ma, and W. Lou proposed the notion of verifiable database

The notion of verifiable database (VDB) enables a resource-constrained client to securely outsource a very large database to an trusted server so that it could later retrieve a database record and update it by assigning a new value. Moreover, any attempt by the server to tamper with the data will be detected by the client. Very recently, Catalano and Fiore proposed an elegant framework to build an efficientVDB that supports public verifiability using a new primitive named vector commitment. In this paper, we point out Catalano-Fiore's VDB framework with vector commitment is vulnerable to the so-called forward automatic update (FAU) attack. We prove that our construction can achieve the desired security properties.

[3] X. Chen, J. Li, J. Ma, Q. Tang, and W. Lou, "New algorithms for secured outsourcing of modular exponentiations.

Cryptography-based privacy-preserving data mining has been proposed to protect the privacy of participating parties' data in this process. In this paper, we address the challenge of outsourcing ID3 decision tree algorithm in the malicious model. Particularly, to securely store and compute private data, the two-participant symmetric homomorphic encryption supporting addition and multiplication is proposed. To keep from malicious behavior of the cloud computing server, the secure garbled circuits are adopted to propose the privacy-preserving weight average protocol. Security and performance are analyzed.

[4] J. Li, Y. Zhang, X. Chen, and Y. Xiang, "Secure attribute-based data sharing for resource-limited users in cloud computing," Compute. Secur., vol. 72, pp. 1–12, Jan. 2018, data sharing becomes an exceptionally attractive

service supplied by cloud computing platforms because of its convenience and economy. As a potential technique for realizing fine-grained data sharing, attribute-based encryption (ABE) has drawn wide attentions. However, most of the existing ABE solutions suffer from the disadvantages of high computation overhead and weak data security, which has severely impeded resource-constrained mobile devices to customize the service. The problem of simultaneously achieving fine- grained ness, high efficiency on the data owner's side, and standard data confidentiality of cloud data sharing actually still remains unresolved. This paper addresses this challenging issue by proposing a new attribute-based data sharing scheme suitable for resource-limited mobile users in cloud computing.

SYSTEM ARCHITECTURE

Below architecture diagram represents mainly the flow of request from the users to the database through servers. In this scenario, the overall system is designed in three tiers separately with three layers called presentation layer, business layer, and data link layer. This project was developed using 3-tier architecture.

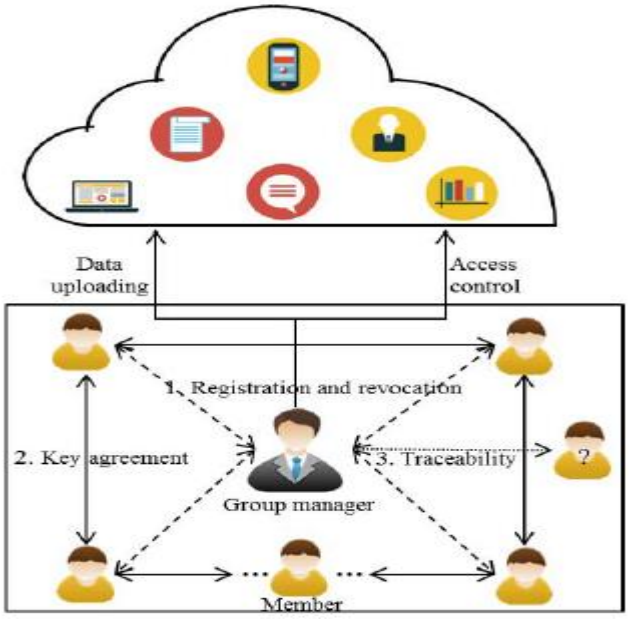

Figure -Tier Architecture diagram

The three-tier software architecture (three- layer architecture) emerged in the 1990s to overcome the limitations of the two-tier architecture. The third tier (middle tier server) is between the user interface (client) and the data management (server) components. This middle tier provides process management where business logic and rules are

executed and can accommodate hundreds of users (compared to only 100 users with the two-tier architecture) by providing functions such as queuing, application execution, and database staging.

The three-tier architecture is used when an effective distributed client/server design is needed that provides (when compared to the two-tier) increased performance, flexibility, maintainability, reusability, and scalability, while hiding the complexity of distributed processing from the user. These characteristics have made three-layer architectures a popular choice for Internet applications and net-centric information systems

IMPLEMENTATION

MODULES:

Members: they are composed of a series of users based on the SBIBD communication model. In our scheme, members are people with the same interests (e.g., bidder, doctors, and businessmen) and they want to share data in the cloud. The most worrying problem when users store data in the
cloud server is the confidentiality of the outsourced data. In our system, users of the same group conduct a key agreement

Cloud:
provides users with seemingly unlimited storageservices. In addition to providing efficient and convenientstorage services for users, the cloud can also provide datasharing services. However, the cloud has the characteristic of honest but curious. In other words, the cloud will not deliberately delete or modify the uploaded data of users,but it will be curious to understand the contents of the storeddata and the user's identity. The cloud is a semitrusted party in our scheme.

Group Manager:
Group manager is responsible for generating system parameters, managing group members (i.e., uploading members 'encrypted data, authorizing group members, revealing the real identity of a member) and for the fault tolerance detection. The group manager in our scheme is a fully trusted third party to both the cloud and group members.
Firstly, users with the same interest register at the group manager to share data in the cloud. In addition, user revocation is also performed by the group manager. Secondly, all members of the group based on the SBIBD structure jointly negotiate a common session key, which can be used to encrypt or decrypt the outsourced data. Finally, when a dispute occurs,the group manager is able to reveal the real identity of the group member. Note that in our system model, data uploading and access control are performed by the group manager.

THE PROPOSED SCHEME

In this section, we present our scheme in detail. Benefiting from the SBIBD structure, the presented scheme can be applied for group data sharing with low communication and computation complexity. The detailed analysis and comparison are introduced in Section 6. Our scheme can be divided into five parts: initialization, key generation, fault detection, file generation and key update, file access, and traceability. A. Initialization, initialization is performed by the group manager, and this part includes parameter initialization, user registration, and SBIBD construction.

1) Parameter Initialization: - A security parameter l is selected as the input of the BDH parameter generator. Then, a bilinear map group system q, G1, G2, eˆ is returned. - Two elements H, H0 ∈ G1 and two integers ξ1, ξ2 ∈ Z∗ q are randomly chosen. Then, the group manager computes H1 = ξ1 ·H0, H2 = ξ2·H0 and U = ξ−1 1 ·H, V = ξ−1 2 ·H. - A generator G ∈ G1 and an integer γ ∈ Z∗ q are randomly selected, and P = γ · G, W = γ · P is computed. In addition, the group manager selects two hash functions h1 and h2, which map its arbitrary length to a nonzero integer and a nonzero point of G1, respectively (i.e., h1 : {0, 1}∗ → Z∗ q , h2 : {0, 1}∗ → G∗ 1). - An additional integer f ∈ Z∗ , q is selected, and the system parameters (G, P, H, H0, H1, H2, U, V, W, h1, h2, f, EncK()) are published; however, the parameters (γ , ξ1, ξ2) is kept private as the master key. Note that EncK() is a secure symmetric encryption algorithm with secret key K. 2) User Registration: Each group member registers with the group manager with his/her identity I Di . After receiving the identity information I, Di of the user, the group manager randomly selects a number xi ∈ Z∗ q and computes Ai = 1 γ +xi · P. In addition, the group manager maps the identity information of member i to a nonzero point Qi = h2(Di) in G1 and computes Si = γ · Qi , which can be used as the secret key for member i. Finally, the secret key for member i is (xi, Ai, Si), and (xi, Ai, I Di) is added to the group user list of the group manager. 3) SBIBD Construction: The communication model for the key generation is based on the SBIBD structure, which reduces the communication complexity and computational complexity for generating a common conference key. Meanwhile, the key agreement based on this structure also supports efficient key updating. Therefore, the structure of the SBIBD should first be constructed. After the registration, the group manager is responsible for building the structure of SBIBD according to the number of group members. SHEN et al.: ANONYMOUS AND TRACEABLE GROUP DATA SHARING IN CLOUD COMPUTING 917

2) **Algorithm 1**

Generation of a (v, k + 1, 1)-
Design Input:
A prime number k.
Output: An SBIBD structure B.
for i = 0; i ≤ k; i + + do
for j = 0; j ≤ k; j + + do
if j == 0 then Bi,j = 0;
else Bi,j = i × k + j;

end if
end for
end for
for i = k + 1; i ≤ k2 + k; i + + do
for j = 0; j ≤ k; j + + do if j == 0
then Bi,j = (i − k − 1) /k + 1;
else Bi,j = jk + 1 + MODk ((i − k − 1)+ (j − 1) (i − k − 1)/k);
end if
end for
end for

Algorithm 2

The Reconstruction of B Input:
An SBIBD structure B.
Output: An SBIBD structure E.
E0 = B0;
(step 1) for t = 1; t ≤ k; t + + do
Et = Btk+1;
(step 1) Btk+1[f lag] = 1;
EEt,t = B(Et,t−1)/k ;
(step 2) Btk+1[f lag] = 1;
end for for i = k + 1;
i ≤ k2 + k;
i + + do
if = 1 then
EBi,(i−1)/k = Bi ;
(step 3)
end if
end for √

CONCLUSION:

In this paper, we present a secure and fault-tolerant key agreement for group data sharing in a cloud storage scheme. Based on the SBIBD and group signature technique, the proposed approach can generate a common conference key efficiently, which can be used to protect the security of the outsourced data and support secure group data sharing in the cloud at the same time. Note that algorithms to construct the SBIBD and mathematical descriptions of the SBIBD are

presented in this paper. Moreover, authentication services and efficient access control are achieved with respect to the group signature technique. In addition, our scheme can support the

Traceability of user identity in an anonymous environment. In terms of dynamic changes of the group member, taking advantage of the key agreement and efficient access control, the computational complexity and communication complexity for updating the common conference key and the encrypted data are relatively low.

REFERENCES

[1] J. Yu, K. Ren, C. Wang, and V. Varadharajan, "Enabling cloud storageauditing with key-exposure resistance," *IEEE Trans. Inf. ForensicsSecurity*, vol. 10, no. 6, pp. 1167–1179, Jun. 2015.

[2] X. Chen, J. Li, X. Huang, J. Ma, and W. Lou, "New publicly verifiabledatabases with efficient updates," *IEEE Trans. Depend. Sec. Comput.*,vol. 12, no. 5, pp. 546–556, Sep. 2015.

[3] X. Chen, J. Li, J. Ma, Q. Tang, and W. Lou, "New algorithms for secureoutsourcing of modular exponentiations," *IEEE Trans. Parallel Distrib.Syst.*, vol. 25, no. 9, pp. 2386–2396, Sep. 2014.

[4] J. Li, Y. Zhang, X. Chen, and Y. Xiang, "Secure attribute-based datasharing for resource-limited users in cloud computing," *Comput. Secur.*,vol. 72, pp. 1–12, Jan. 2018, doi: 10.1016/j.cose.2017.08.007.