SECURE DATA RETRIEVAL FROM CLOUD THROUGH ASYMMETRIC GROUP KEY AGREEMENT

[1]E.Ganesh, [2]M.Prabu

[1]Research Scholar, [2]Assistant Professor,
[1]Saveetha Engineering College, Chennai, India.
[2]Department of ECE, Misrmal Navajee Munoth Jain Engineering College, Chennai -97

Abstract : Cloud storage enables users within a shared group to upload and access data in the cloud. The use of cloud storage has become increasingly prevalent due to its ability to enhance secure remote information auditing. However, recent analyses have raised concerns regarding the secure and adequate public information integrity auditing of shared dynamic information. Notably, some practical storage systems lack security measures against connivance attacks from cloud storage servers and revoked cluster users. To address these issues, we propose a project plan based on the verifier-local revocation cluster signature. Our scheme incorporates an adequate public integrity auditing mechanism with secure cluster user revocation using a signature or group key, where the group key signature provides designed re-signature capabilities. Our solution aims to overcome the challenges associated with secure and adequate public information integrity auditing schemes, particularly in the context of secure cluster user revocation with a signature**.**

Index Terms **- Cloud, Security, Key, Encryption.**

I. INTRODUCTION

The concept of cloud computing involves the delivery of computing resources, including hardware and software, as a service over a network [1,2]. Its name derives from the common use of a cloud symbol as an abstraction for the complex infrastructure contained in system diagrams. Cloud computing allows users to entrust their data, software, and computation to remote services provided by third parties over the Internet [3,4]. These services typically provide access to advanced software applications and high-end networks of server computers. The aim of cloud computing is to leverage the power of supercomputing, traditionally used by military and research facilities, to enable consumer-oriented applications such as financial portfolios, personalized information delivery, data storage, and large-scale computer games[5].

Cloud computing typically involves the use of networks comprising large groups of servers that utilize low-cost consumer PC technology with specialized connections to distribute data-processing tasks among them [6]. These shared IT infrastructures contain large pools of interconnected systems that are often virtualized to maximize computing power [7,8]. While several strategies have been implemented to reduce overhead at the data owner's end while ensuring cloud data security and privacy, they do not support dynamic data operations, such as insertion, deletion, append, and update, in single and multi-user cloud environments. These strategies are only productive when the data's integrity is verified by the public verifier [9,10]. Therefore, dynamic data operations are frequently used in cloud storage to reduce computational costs.

The use of a privacy-preserving public integrity auditing scheme with algebraic signature in conjunction with dynamic data operations, including insertion, deletion, modification, append, and update, to achieve efficient computational cost reduction in a cloud environment.

II. EXISTING SYSTEM

System analysis is a problem-solving approach that involves breaking down a system into its constituent parts to assess their individual efficacy and how well they work together to achieve their objectives. The process of system analysis entails gathering and interpreting factual data to diagnose problems and utilizing this information for system improvements.

Several solutions and their variants have been implemented to ensure the integrity and availability of remote cloud storage. Among these solutions, those that allow data modification are referred to as dynamic schemes, while those that do not are called static schemes. A publicly verifiable scheme is one that enables not only the data owners but also third-party auditors to verify data integrity. However, the dynamic schemes thus far have focused on scenarios where data owners are the only ones authorized to modify data. To support multiple users' data operations, Wang et al. introduced a data integrity scheme based on ring signatures. To further enhance this scheme and enable group user revocation, Wang et al. developed a proxy re-signature-based approach..

III. METHODOLOGY

The inadequacy of the previous schemes has prompted us to devise a new scheme that is not only efficient and dependable but also ensures secure group user revocation. In this regard, we introduce a construction that allows for group data encryption and decryption during data modification processing, while also facilitating efficient and secure user revocation. Our approach involves utilizing a vector commitment scheme for the database and implementing the Asymmetric Group Key Agreement (AGKA) and group signatures to support database updates among group users and efficient group user revocation, respectively. Specifically, the AGKA protocol is utilized by the group user to encrypt/decrypt the shared database, ensuring that any user in the group can encrypt/decrypt messages from any other group user. In addition, the group signature is used to prevent collusion between the cloud and revoked group users. The data owner is involved in the user revocation phase, and the cloud is unable to revoke data that was last modified by a revoked user.

TECHNOLOGY USED

In contrast to most programming languages, which require either compilation or interpretation to enable program execution, the Java programming language follows a unique approach where a program undergoes both compilation and interpretation. Firstly, the program is compiled and translated into an intermediate language known as Java byte codes. These codes are platform-independent and are interpreted by the interpreter on the Java platform. During execution, the interpreter parses and runs each instruction of the Java byte codes on the computer. The compilation process is executed only once, while interpretation is performed each time the program is executed as shown in figure 1.

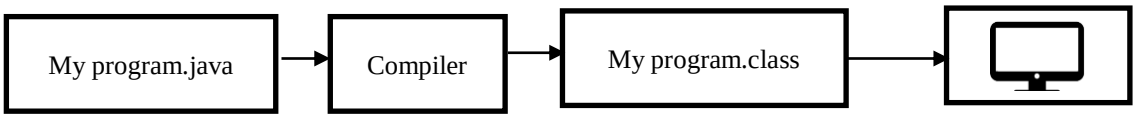

Figure 1: Overview of system.

The Java Platform

To run a program, it needs a specific hardware or software environment, which is called a platform. Widely-used platforms include Windows 2000, Linux, Solaris, and MacOS, which are usually a combination of an operating system and hardware. In contrast, the Java platform is unique because it is a software-only platform that operates on top of other hardware-based platforms.

ODBC

Microsoft Open Database Connectivity (ODBC) is a standard programming interface used by application developers and database providers. Before the introduction of ODBC, programmers had to use proprietary languages for each database they wanted to interface with. With ODBC, the choice of database system becomes almost irrelevant from a coding perspective. Through the ODBC Administrator in Control Panel, a particular database can be associated with a data source that an ODBC application program is written to use. An ODBC data source can be thought of as a door with a name on it, leading to a particular database. The ODBC system files are installed when a separate database application is installed, such as SQL Server Client or Visual Basic 4.0. ODBC allows applications to use the same set of function calls to interface with any data source, regardless of the database vendor. There are ODBC drivers available for many popular database systems, including Excel spreadsheets and plain text files.

JDBC

Sun Microsystems created Java Database Connectivity (JDBC) to establish a standardized API for databases in Java. JDBC enables a uniform interface for accessing various relational database management systems (RDBMSs) by using modular "plug-in" database connectivity drivers. To have JDBC support, a database provider must develop the driver for each platform where the database and Java will run. To expand the adoption of JDBC, Sun based its framework on ODBC, which is already widely supported on multiple platforms. This design decision allows vendors to introduce JDBC drivers more quickly than creating a new connectivity solution from scratch.

SYSTEM DESIGN

The process of defining the architecture, components, modules, interfaces, and data of a system to meet specific requirements is known as system design. It can be viewed as the application of systems theory to the development of a product. Figure 2 shows the data flow diagram of proposed approach.

SYSTEM IMPLEMENTATION

The main idea behind ring signatures is to obscure the identity of the signer on each block to maintain the privacy of sensitive information from the public verifier. However, traditional ring signatures do not support blockless verifiability, meaning that the verifier has to download all the data from the cloud to verify the accuracy of the shared data, which consumes more bandwidth and time. To address this, a new holomorphic authenticable ring signature (HARS) scheme has been developed, which extends the traditional ring signature scheme. HARS-generated ring signatures not only preserve identity privacy but also support blockless verifiability as shown in figure 3.

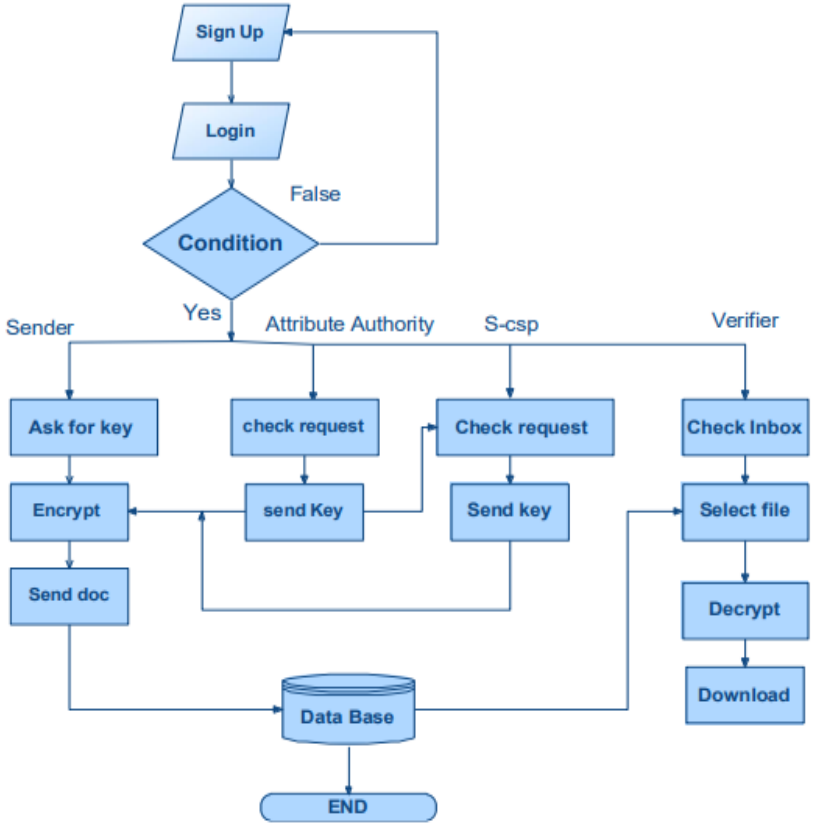

Figure 2: Data Flow Diagram

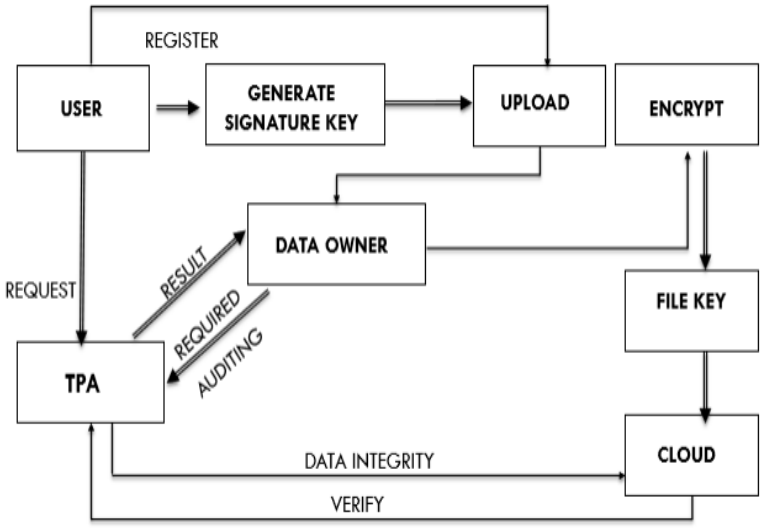

Figure 3: System Architecture

REGISTRATION & LOGIN MODULE

In KASE, the server is able to utilize an aggregate trapdoor and public information to conduct keyword searches and provide the results to Bob. This means that sharing a single aggregate key can enable the delegation of keyword search rights. It is worth noting that delegation of decryption rights can be achieved using the key-aggregate encryption approach proposed in, but the challenge of delegating keyword search rights along with decryption rights remains unresolved, and is the topic of this paper. In conclusion, the focus of this paper is to address the issue of constructing a KASE.

GROUP SIGNATURE MODULE

Chaum and Heyst introduced group signature as a method to provide anonymity to signers, wherein each member has a private key to sign messages while keeping their identity hidden. A third party is typically involved to maintain signature anonymity using a special trapdoor. Some systems support revocation of group membership without affecting unrevoked users' signing ability. Boneh and Shacham proposed an efficient group signature scheme with local revocation by verifiers, providing group signature properties such as selfless-anonymity and traceability.

REVOKED GROUP USERS

The group signature scheme is designed to prevent collusion between the cloud and revoked group users. During user revocation, the data owner is involved and the cloud is unable to revoke data last modified by a revoked user. However, an attacker outside the group, including the cloud storage server, may gain knowledge of the plaintext data. To achieve this, the attacker must at least breach the security of the adopted group data encryption scheme. In the case where the cloud storage server colludes with revoked group users, they can provide illegal data without detection. Since the cloud storage server is semi-trusted in a cloud

environment, it is reasonable to assume that a revoked user will collude with the cloud server and share its secret group key. Although server proxy group user revocation saves communication and computation costs, it also makes the scheme vulnerable to a malicious cloud storage server that can obtain the secret key of revoked users during the user revocation phase.

PUBLIC AUDITING

Our contribution in this paper is threefold: (1) We investigate the secure and efficient auditing of shared dynamic data for multi-user operations in a ciphertext database. (2) We incorporate the primitives of Victor commitment, asymmetric group key agreement, and group signature to achieve our goal. (3) We propose a novel approach for group user revocation to enhance the security of our system.

IV. CONCLUSION

To summarize, we have developed an efficient public integrity auditing scheme for cloud-based data storage that supports dynamic operations such as insertion, append, deletion, and update. The scheme is based on algebraic signatures and has a low computational cost for the data owner compared to other similar schemes in literature. Our results show that this method can be extended for an integrity auditing system for large archival files in distributed cloud storage systems as well as for a system with data traceability.

REFERENCES

[1] M. Bahrami, "Cloud Computing for Emerging Mobile Cloud Apps," 2015 3rd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering, San Francisco, CA, USA, 2015, pp. 4-5.

[2] T. Mengistu, A. Alahmadi, A. Albuali, Y. Alsenani and D. Che, "A "No Data Center" Solution to Cloud Computing," 2017 IEEE 10th International Conference on Cloud Computing (CLOUD), Honolulu, HI, USA, 2017, pp. 714-717.

[3] Abuhussein, H. Bedi and S. Shiva, "Towards a Stakeholder-Oriented Taxonomical Approach for Secure Cloud Computing," 2013 IEEE Sixth International Conference on Cloud Computing, Santa Clara, CA, USA, 2013, pp. 958-959.

[4] P. Dutta, T. Mukherjee, V. G. Hegde and S. Gujar, "C-Cloud: A Cost-Efficient Reliable Cloud of Surplus Computing Resources," 2014 IEEE 7th International Conference on Cloud Computing, Anchorage, AK, USA, 2014, pp. 986-987.

[5] M. Eldred, C. Adams and A. Good, "Trust Challenges in a High Performance Cloud Computing Project," 2014 IEEE 6th International Conference on Cloud Computing Technology and Science, Singapore, 2014, pp. 1045-1050.

[6] V. Marbukh, "Systemic Risks in the Cloud Computing Model: Complex Systems Perspective," 2016 IEEE 9th International Conference on Cloud Computing (CLOUD), San Francisco, CA, USA, 2016, pp. 863-866.

[7] M. Aazam and E. N. Huh, "Inter-cloud Media Storage and Media Cloud Architecture for Inter-cloud Communication," 2014 IEEE 7th International Conference on Cloud Computing, Anchorage, AK, USA, 2014, pp. 982-985.

[8] Zhou, A. V. Dastjerdi, R. N. Calheiros, S. N. Srirama and R. Buyya, "A Context Sensitive Offloading Scheme for Mobile Cloud Computing Service," 2015 IEEE 8th International Conference on Cloud Computing, New York, NY, USA, 2015, pp. 869-876.

[9] C. -L. Yang, B. -N. Hwang and B. J. C. Yuan, "Key consideration factors of adopting cloud computing for science," 4th IEEE International Conference on Cloud Computing Technology and Science Proceedings, Taipei, Taiwan, 2012, pp. 597-600.

[10] M. Kretzschmar, M. Golling and S. Hanigk, "Security Management Areas in the Inter-cloud," 2011 IEEE 4th International Conference on Cloud Computing, Washington, DC, USA, 2011, pp. 762-763.