Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 9, September 2014, pg.684 – 690

RESEARCH ARTICLE

Cloud Data Service for Issues in Scalable Data Integration Using Multi Authority Attribute Based Encryption

Praveena.A[1], Sasikala.C[2], Keerthana.P[3], Chithrakumar.T[4]

[1]A.Praveena, Student, Department of Computer Science, VSB College of Engineering Technical Campus, Coimbatore, TamilNadu (India)
praveenaayyasamy@gmail.com

[2]C.Sasikala, Assistant Professor, Department of Computer Science, VSB College of Engineering Technical Campus, Coimbatore, TamilNadu (India)
cmsasi.87@gmail.com

[3]P.Keerthana, Student, Department of Computer Science VSB College of Engineering Technical Campus, Coimbatore, TamilNadu (India)
Keerthana1492@gmail.com

[4]Chithrakumar.T, Student, Department of Computer Science VSB College of Engineering Technical Campus, Coimbatore, TamilNadu (India)
Chithrakumar.t@gmail.com

Abstract:

Identity Privacy of the outsourced data as of public auditing is modelled as privacy concern in the cloud data service through the public auditing. With cloud data services, it is common place for data to be not only stored in the cloud, but also shared across frequent users. Regrettably, the integrity of cloud data is focus to cynicism due to the prolongation of hardware/software failures and human errors. We propose a novel privacy-preserving mechanism that supports public auditing on shared data stored in the cloud. Yet, issues such as risks of privacy exposure, scalability in key management, supple access and efficient user revocation, have remained the foremost challenges and achieving fine -grained ,cryptographically enforced data access control. In particular, we exploit multi authority attribute based encryption to compute verification of the data stored in the cloud to audit the correctness of shared data. Through imposing the multi authority-ABE technique our mechanism, the identity of the attribute on each block in shared data is kept private from public verifiers so, that they can efficiently verify the data integrity without retrieving the entire file. It can also perform multiple auditing tasks simultaneously.

Keywords: Cloud Security, Privacy Preserving, MA-ABE, Cloud Auditing, Anonymization

I. Introduction

Several trends are opening up the era of Cloud Computing, which is an Internet-based development and use of computer technology. The ever cheaper and more powerful processors, together with the "software as a service" (SaaS) computing architecture, are transforming data centers into pools of computing service on a huge scale. Although envisioned as a promising service platform for the Internet, this new data storage paradigm in "Cloud" brings about many challenging design issues which have profound influence on the security and performance of the overall system. Meanwhile, the increasing network bandwidth and reliable yet flexible network connections make it even possible that clients can now subscribe high-quality services from data and software that reside solely on remote data centers.

One of the biggest concerns with cloud data storage is that of data integrity verification at untrusted servers. For example, the storage service provider, which experiences Byzantine failures occasionally, may decide to hide the data errors from the clients for the benefit of their own. Consider the large size of the outsourced electronic data and the client's constrained resource capability, the core of the problem can be generalized as how can the client find an efficient way to perform periodical integrity verifications without the local copy of data files. In order to solve the problem of data integrity checking, many schemes are proposed under different systems and security models [3], [4], [5], [6], [7], [8], [9], [10], [11], [12].

In these mechanisms, data is divided into many small blocks, where each block is independently signed by the owner; and a random combination of all the blocks instead of the whole data is retrieved during integrity checking [10]. A public verifier could be a data user (e.g., researcher) who would like to utilize the owner's data via the cloud or a third-party auditor (TPA) who can provide expert integrity checking services. Moving a step forward, Wang et al. designed an advanced auditing mechanism [6] (named as WWRL in this paper), so that during public auditing on cloud data, the content of private data belonging to a personal user is not disclosed to any public verifiers.

Unfortunately, current public auditing solutions mentioned above only focus on personal data in the cloud [2]. We believe that sharing data among multiple users is perhaps one of the most attractive features that motivates cloud storage. Therefore, it is also necessary to ensure the integrity of shared data in the cloud is correct. Existing public auditing mechanisms can actually be extended to verify shared data integrity [2], [6], [17], [18]. However, a new significant privacy issue introduced in the case of shared data with the use of existing mechanisms is the leakage of identity privacy to public verifiers [1].

For instance, Alice and Bob work together as a group and share a file in the cloud (as presented in Fig. 1). The shared file is divided into a number of small blocks, where each block is independently signed by one of the two users with existing public auditing solutions (e.g., [6]). Once a block in this shared file is modified by a user, this user needs to sign the new block using his/her private key. Eventually, different blocks are signed by different users due to the modification introduced by these two different users. Then, in order to correctly audit the integrity of the entire data, a public verifier needs to choose the appropriate public key for each block (e.g., a block signed by Alice can only be correctly verified by Alice's public key).

As a result, this public verifier will inevitably learn the identity of the signer on each block due to the unique binding between an identity and a public key via digital certificates under public key infrastructure (PKI). Specifically, as shown in Fig. 1, after performing several auditing tasks, this public verifier can first learn that Alice may be a more important role in the group because most of the blocks in the shared file are always signed by Alice; on the other hand, this public verifier can also easily deduce that the eighth block may contain data of a higher value (e.g., a final bid in an auction), because this block is frequently modified by the two different users. Failing to preserve identity privacy on shared data during public auditing will reveal significant confidential information (e.g., which particular user in the group or special block in shared data is a more valuable target) to public verifiers.

In order to protect these confidential information, it is essential and critical to preserve identity privacy from public verifiers during public auditing. In this paper, to solve the above privacy issue on shared data, we propose Oruta,1 a novel privacy-preserving public auditing mechanism. More specifically, we utilize ring signatures to construct homomorphic authenticators [11] in Oruta, so that a public verifier is able to verify the integrity of shared data without retrieving the entire data—while the identity of the signer on each block in shared data is kept private from the public verifier.

More specifically, we utilize ring signatures to construct homomorphic authenticators [11] in Oruta, so that a public verifier is able to verify the integrity of shared data without retrieving the entire data—while the identity of the signer on each block in shared data is kept private from the public verifier. In addition, we further extend our mechanism to support batch auditing, which can perform multiple auditing tasks simultaneously and improve the efficiency of verification for multiple auditing tasks.

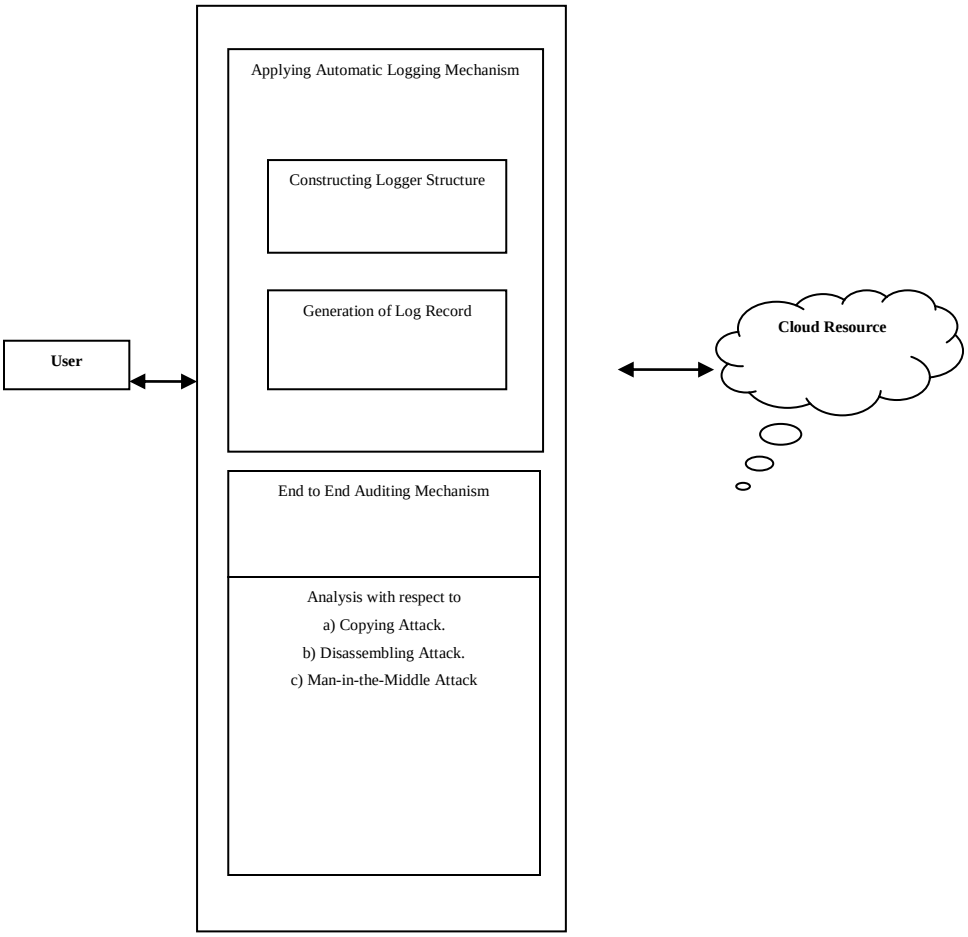

Fig 1: MA-ABE scheme in Cloud

Meanwhile, Oruta is compatible with random masking [6], which has been utilized in WWRL and can preserve data privacy from public verifiers. Moreover, we also leverage index hash tables from a previous public auditing solution to support dynamic data. The remainder of this paper is organized as follows. In Section 2, we present the related work. In Section 3, we introduce MA-ABE Scheme as proposed architecture, The detailed design and security analysis .In Section 4, we evaluate the performance of MA-ABE. Finally, we conclude this paper in Section 6.

II. Related Works

Privacy-preserving mechanism that supports public auditing on shared data stored in the cloud. In particular, through exploitation of ring signatures to compute verification metadata needed to audit the correctness of shared data. In addition, our mechanism is able to perform multiple auditing tasks simultaneously instead of

verifying them one by one With mechanism, the identity of the signer on each block in shared data is kept private from public verifiers, who are able to efficiently verify shared data integrity without retrieving the entire file.

III. System Analysis and Design

A.1. Secure Architecture for Information sharing and Management of data stored in the cloud

The user centric, secure sharing of data's stored on semi-trusted servers, and focus on addressing the complicated and challenging key management issues. In order to protect the personal health data stored on a semi-trusted server, we adopt attribute-based encryption as the main encryption primitive. Using ABE, access policies are expressed based on the attributes of users or data, which enables a user to selectively share data among a set of users by encrypting the file under a set of attributes, without the need to know a complete list of users. We provide a thorough analysis of the complexity and scalability of our proposed secure data sharing solution, in terms of multiple metrics in computation, communication, storage and key management.

We also compare our scheme to several previous ones in complexity, scalability and security. Furthermore, we demonstrate the efficiency of our scheme by implementing it on a modern workstation. Finally, most of the existing works do not differentiate between the personal and public domains, which have different attribute definitions, key management requirements and scalability issues. Our idea of conceptually dividing the system into two types of domains is similar with that in, however a key difference is in a single TA is still assumed to govern the whole professional domain. A single data owner can encrypt her data and share with multiple authorized users, by distributing keys to them that contain attribute-based access privileges. They also propose a method for the data owner to revoke a user efficiently by delegating the updates of affected cipher texts and user secret keys to the cloud server. Since the key update operations can be aggregated over time, their scheme achieves low amortized overhead.

A.2. Multiuser Attribute based Encryption scheme for Public domains:

The Users obtain their attribute-based secret keys from the AAs, without directly interacting with the owners. To control access from users, owners are free to specify role-based fine-grained access policies for data files, while do not need to know the list of authorized users when doing encryption. For the purpose of PSD access, each PHR file is labelled with its data attributes, while the key size is only linear with the number of file categories a user can access. Since the PUDs contain the majority of users, it greatly reduces the key management overhead for both the owners and users.

Since the number of users in a PSD is often small, it reduces the burden for the owner. When encrypting the data for PSD, all that the owner needs to know is the intrinsic data properties. The multi-domain approach best models different user types and access requirements in a system. The use of ABE makes the encrypted self-protective, they can be accessed by only authorized users even when storing on a semi-trusted server, and when the owner is not online. In addition, efficient and on-demand user revocation is made possible via our ABE enhancements.

A.3. Applying the AES Algorithm based on user

The AES encryption algorithm theory, this paper proposes the AES encryption methods and steps which are appropriate for the identity authentication of engine anti-theft system. It also designs the experimental program and obtains good results. The encryption algorithms like AES, DES, RC4 and others are available for the same. We have used the Advanced Encryption Standards algorithm for encryption and decryption of the data. The most widely accepted algorithm is AES algorithm. We have developed an application on Android platform which allows the user to encrypt the messages before it is transmitted over the network. This application can run on any device which works on Android platform. This application provides a secure, fast, and strong encryption of the data. There is a huge amount of confusion and diffusion of the data during encryption which makes it very difficult for an attacker to interpret the encryption pattern and the plain text form of the encrypted data. The messages encrypted by the developed application are also resistant to Brute-Force and pattern attacks.

A.4. Key Management with policy Strategies and revocation Process

Compared with existing revocable ABE schemes, the main advantage of our solution is small re-keying message sizes. To further show the storage and communication costs, we provide a numerical analysis using typical parameter settings in the supplementary material. In evaluate the computational cost of our scheme through combined implementation and simulation. We provide the first implementation of the MA-ABE scheme, and also

integrated the ABE algorithms into a prototype system To revoke a user, the maximum re-keying message size is linear with the number of attributes in that user's secret key. These indicate our scheme is more scalable than existing works.

The MA-ABE scheme is tested on a PC with 3.4 GHz processor, using the pairing based cryptography library. The public parameters are chosen to provide 80 bits security level, and we use a pairing-friendly type-A 160-bit elliptic curve group. This parameter setting has also been adopted in other related works in ABE. We then use the ABE algorithms to encrypt randomly generated XML-formatted files, and implement the user-interfaces for data input and output. Due to space limitations, the details of prototype implementation are reported in. In the supplementary material we present benchmarks of cryptographic operations and detailed timing results for the two ABE algorithms used by our framework. It is shown that, the decryption operation in our enhanced MA-ABE scheme is quite fast.

IV. Performance Analysis

The server's computation cost spent in user revocation to evaluate the system performance of user revocation. Especially, the lazy-revocation method greatly reduces the cost of revocation, because it aggregates multiple cipher text/key update operations, which amortizes the computations over time. The details of the experimental/simulation evaluation results are presented in the supplementary material. A novel framework of secure sharing of personal health records in cloud computing.

Considering partially trustworthy cloud servers, we argue that to fully realize the User-centric concept, user shall have complete control of their own privacy through encrypting their data files to allow fine-grained access.

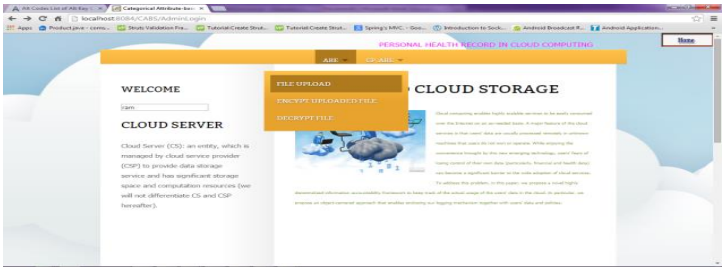

Fig 2 : Performance of the Privacy Preserving

We utilize ABE to encrypt the data, so that user can allow access not only by personal users, but also various users from public domains with different professional roles, qualifications and affiliations. The framework addresses the unique challenges brought by multiple data owners and users, in that we greatly reduce the complexity of key management while enhance the privacy guarantees compared with previous works. Furthermore, we enhance an existing MA-ABE scheme to handle efficient and on-demand user revocation.

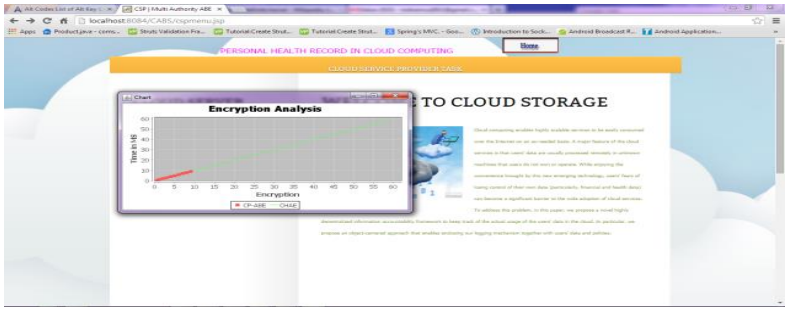

Figure 3: Performance analysis of the encryption logics

Through implementation and simulation, we show that our solution is both scalable and efficient.

V. Conclusion

We modelled a privacy-preserving mechanism that supports public auditing on shared data stored in the cloud against risks of privacy exposure. scalability in key management, flexible access and efficient user revocation, have remained the most important constraints and it has been modelled toward achieving fine-grained, cryptographically enforced data access control In particular, we exploit and explore Multi Authority Attribute based Encryption to compute verification of the data stored in the cloud to audit the correctness of shared data. Through imposing the Multi Authority - ABE technique our mechanism, the identity of the attribute on each block in shared data is kept private from public verifiers, who are able to efficiently verify shared data integrity without retrieving the entire file. In addition, our mechanism is able to perform multiple auditing tasks simultaneously as a substitute of verifying them one by one.

References

[1] B.Wang, B. Li, and H. Li, "Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud," Proc. IEEE Fifth Int'l Conf. Cloud Computing, pp.295-302, 2012.

[2] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing," Proc. 14th European Symp. Research in Computer Security (ESORICS '09), pp. 355-370, 2009.

[3] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable Data Possession at Untrusted Stores," Proc. 14th ACM Conf. Computer and Comm. Security (CCS '07), pp. 598-609, 2007.

[4] A. Juels and B.S. Kaliski Jr., "Pors: Proofs of Retrievability for Large Files," Proc. 14th ACM Conf. Computer and Comm. Security (CCS '07), pp. 584-597, 2007.

[5] H. Shacham and B. Waters, "Compact Proofs of Retrievability,"Proc. 14th Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT '08), pp. 90-107, 2008.

[6] K.D. Bowers, A. Juels, and A. Oprea, "Proofs of Retrievability: Theory and Implementation," Report 2008/175, Cryptology ePrint Archive, 2008.

[7] M. Naor and G.N. Rothblum, "The Complexity of Online Memory Checking," Proc. 46th Ann. IEEE Symp. Foundations of Computer Science (FOCS '05), pp. 573-584, 2005.

[8] E.-C. Chang and J. Xu, "Remote Integrity Check with Dishonest Storage Server," Proc. 13th European Symp. Research in Computer Security (ESORICS '08), pp. 223-237, 2008.

[9] M.A. Shah, R. Swaminathan, and M. Baker, "Privacy-Preserving Audit and Extraction of Digital Contents," Report 2008/186, Cryptology ePrint Archive, 2008.

[10] A. Oprea, M.K. Reiter, and K. Yang, "Space-Efficient Block Storage Integrity," Proc. 12th Ann. Network and Distributed System Security Symp. (NDSS '05), 2005.

[11] T. Schwarz and E.L. Miller, "Store, Forget, and Check: Using Algebraic Signatures to Check Remotely Administered Storage," Proc. 26th IEEE Int'l Conf. Distributed Computing Systems (ICDCS '06), p. 12, 2006.

[12] Q. Wang, K. Ren, W. Lou, and Y. Zhang, "Dependable and Secure Sensor Data Storage with Dynamic Integrity Assurance," Proc. IEEE INFOCOM, pp. 954-962, Apr. 2009.

[13] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling Public Verifiability and Data Dynamic for Storage Security in Cloud Computing," Proc. 14th European Conf. Research in Computer Security (ESORICS'09), pp. 355-370, 2009.

[14] C. Wang, Q. Wang, K. Ren, and W. Lou, "Ensuring Data Storage Security in Cloud Computing," Proc. 17th Int'l Workshop Quality of Service (IWQoS'09), pp. 1-9, 2009.

[15] B. Chen, R. Curtmola, G. Ateniese, and R. Burns, "Remote Data Checking for Network Coding-Based Distributed Storage Systems," Proc. ACM Workshop Cloud Computing Security Workshop (CCSW'10), pp. 31-42, 2010.

[16] Y. Zhu, H. Wang, Z. Hu, G.-J. Ahn, H. Hu, and S.S Yau, "Dynamic Audit Services for Integrity Verification of Outsourced Storages in Clouds," Proc. ACM Symp. Applied Computing (SAC'11), pp. 1550-1557, 2011.

[17] N. Cao, S. Yu, Z. Yang, W. Lou, and Y.T. Hou, "LT Codes-Based Secure and Reliable Cloud Storage Service," Proc. IEEE INFOCOM, 2012.

[18] B. Wang, B. Li, and H. Li, "Certificateless Public Auditing for Data Integrity in the Cloud," Proc. IEEE Conf. Comm. and Network Security (CNS'13), pp. 276-284, 2013.

