International Journal of Scientific Research in Computer Science, Engineering and Information Technology

© 2017 IJSRCSEIT | Volume 2 | Issue 3 | ISSN : 2456-3307

Provable Multicopy Dynamic Data Possession in Cloud Computing

Manasa.B. K, Bhavya B M
MCA Department, P.E.S. College of Engineering, Mandya, Karnataka, India

ABSTRACT

Increangly more and more organizations are opting for outsourcing data to remote cloud service providers (CSPs). Customers can rent the CSPs storage infrastructure to store and retrieve almost unlimited amount of data by paying fees metered in gigabyte/month. For an increased level of scalability, availability, and durability, some customers may want their data to be replicated on multiple servers across multiple data centers. The more copies the CSP is asked to store, the more fees the customers are charged. Therefore, customers need to have a strong guarantee that the CSP is storing all data copies that are agreed upon in the service contract, and all these copies are consistent with the most recent modifications issued by the customers. In this paper, we propose a map-based Provable multicopy dynamic data possession (MB-PMDDP) scheme that has the following features: 1) it provides an evidence to the customers that the CSP is not cheating by storing fewer copies; 2) it supports outsourcing of dynamic data, i.e., it supports block-level operations, such as block modification, insertion, deletion, and append; and 3) it allows authorized users to seamlessly access the file copies stored by the CSP. We give a comparative analysis of the proposed MB-MDDP scheme with a reference model obtained by extending existing possession of dynamic single-copy schemes. The theoretical analysis is validated through experimental results on a commercial cloud platform. In addition, we show the security against colluding servers, and discuss how to identify corrupted copies by slightly modifying the proposed scheme. a wide range of users, including healthcare providers, family members or friends. Due to the high cost of building and maintaining specialized data centers, many PHR services are outsourced to or provided by third-party service providers, for example, Microsoft HealthVault1. Recently, architectures of storing PHRs in cloud computing have been proposed in. While it is exciting to have convenient PHR services for everyone, there are many security and privacy risk which could impede its wide adoption. The main concern is about whether the patients could actually control the sharing of their sensitive personal health information (PHI), especially when they are stored on a third-party server which people may not fully trust. On the one hand, although there exist healthcare regulations such as HIPAA which is recently amended to incorporate business associates, cloud providers are usually not covered entities. On the other hand, due to the high value of the sensitive personal health information (PHI), the third-party storage servers are often the targets of various malicious behaviors which may lead to exposure of the PHI. As a famous incident, a Department of Veterans Affairs database containing sensitive PHI of 26.5 million military veterans, including their social security numbers and health problems was stolen by an employee who took the data home without authorization. To ensure patient-centric privacy control over their own PHRs, it is essential to have fine-grained data access control mechanisms that work with semi-trusted servers.(1) We propose a novel ABE-based framework for patient-centric secure sharing of PHRs in cloud computing environments, under the multi-owner settings. To address the key management challenges, we conceptually divide the users in the system into two types of domains, namely public and personal domains. In particular, the majority professional users are managed distributively by attribute authorities in the former, while each owner only needs to manage the keys of a small number of users in her personal domain. In this way, our framework can simultaneously handle different types of PHR sharing applications' requirements, while incurring minimal key management overhead for both owners and users in the system. In addition, the framework enforces write access control, handles dynamic policy updates, and provides break-glass access to PHRs under emergence scenarios. (2) In the public domain, we

CSEIT1723262 | Received : 05 June 2017 | Accepted : 13 June 2017 | May-June-2017 [(2)3: 772-776]

use multi-authority ABE (MA-ABE) to improve the security and avoid key escrow problem. Each attribute authority (AA) in it governs a disjoint subset of user role attributes, while none of them alone is able to control the security of the whole system. We propose mechanisms for key distribution and encryption so that PHR owners can specify personalized fine-grained role-based access policies during file encryption. In the personal domain, owners directly assign access privileges for personal users and encrypt a PHR file under its data attributes. Furthermore, we enhance MA-ABE by putting forward an efficient and on-demand user/attribute revocation scheme, and prove its security under standard security assumptions. In this way, patients have full privacy control over their PHRs. (3) We provide a thorough analysis of the complexity and scalability of our proposed secure PHR 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 and performing experiments/simulations.

Keywords : PDP, CSP, Dynamic Data, MB-MDDP, MDDP, PHR, MA-ABE

I. INTRODUCTION

Outsourcing data to a remote cloud service provider (CSP) allows organizations to store more data on the CSP than on private computer systems. Such outsourcing of data storage enables organizations to concentrate on innovations and relieves the burden of constant server updates and other computing issues. Moreover, many authorized users can access the remotely stored data from different geographic locations making it more convenient for them. Once the data has been outsourced to a remote CSP which may not be trustworthy, the data owners lose the direct control over their sensitive data. This lack of control raises new formidable and challenging tasks related to data confidentiality and integrity protection in cloud computing. The confidentiality issue can be handled by encrypting sensitive data before outsourcing to remote servers. As such, it is a crucial demand of customers to have a strong evidence that the cloud servers still possess their data and it is not being tampered with or partially deleted over time. Consequently, many researchers have focused on the problem of provable data possession (PDP) and proposed different schemes to audit the data stored on remote servers. PDP is a technique for validating data integrity over remote servers. In a typical PDP model, the data owner generates some metadata/information for a data file to be used later for verification purposes through a *challenge-response* protocol with the remote/cloud server. The owner sends the file to be stored on a remote server which may be un trusted, and deletes the local copy of the file. As a proof that the server is still possessing the data file in its original form, it needs to correctly compute a response to a challenge vector sent from a verifier who can be the original data owner or a trusted entity that shares some information with the owner. Researchers have proposed different variations of PDP schemes under different cryptographic assumptions.

One of the core design principles of outsourcing data is to provide dynamic behavior of data for various applications. This means that the remotely stored data can be not only accessed by the authorized users, but also updated and scaled (through block level operations) by the data owner.

PDP schemes presented in focus on only *static* or warehoused data, where the outsourced data is kept unchanged over remote servers. Examples of PDP constructions that deal with d*ynamic* data. The latter are however for a *single* copy of the data file. Although PDP schemes have been presented for *multiple* copies of *static* data to the best of our knowledge, this work is the first PDP scheme directly dealing with *multiple* copies of *dynamic* data. In Appendix A, we provide a summary of related work. When verifying multiple data copies, the overall system integrity check fails if there is one or more corrupted copies. To address this issue and recognize which copies have been corrupted, we discuss a slight modification to be applied to the proposed scheme.

II. METHODS AND MATERIAL

A. Main Contributions

Our contributions can be summarized as follows: We propose a map-based multi-copy dynamic data possession (MB-MDDP) scheme. This scheme provides an adequate guarantee that the CSP stores all copies that are agreed upon in the service contract. Moreover, the scheme supports outsourcing of dynamic data, i.e., it supports block-level operations such as block modification, insertion, deletion, and append. The authorized users, who have the right to access the owner's file, can seamlessly access the copies received from the CSP. • We give a thorough comparison of MB-MDDP with a reference scheme, which one can obtain by extending existing PDP models for dynamic single-copy data. We also report our implementation and experiments using Amazon cloud platform. We show the security of our scheme against colluding servers, and discuss a slight modification of the proposed scheme to identify corrupted copies. Remark 1: Proof of retrievability (POR) is a complementary approach to PDP, and is stronger than PDP in the sense that the verifier can reconstruct the entire file from responses that are reliably transmitted from the server. This is due to encoding of the data file, for example using erasure codes, before outsourcing to remote servers. Various POR schemes can be found in the literature, which focus on static data. In this work, we do not encode the data to be outsourced for the following reasons. First, we are dealing with dynamic data, and hence if the data file is encoded before outsourcing, modifying a portion of the file requires re-encoding the data file which may not be acceptable in practical applications due to high computation overhead. Second, we are considering economically-motivated CSPs that may attempt to use less storage than required by the service contract through deletion of a few copies of the file. The CSPs have almost no financial benefit by deleting only a small portion of a copy of the file. Third, and more importantly, unlike erasure codes, duplicating data files across multiple servers achieves scalability which is a fundamental customer requirement in CC systems. A file that is duplicated and stored strategically on multiple servers – located at various geographic locations – can help reduce access time and communication cost for users. Besides, a server's copy can be reconstructed even from a complete damage using duplicated copies on other servers.

B. Paper Organization

The remainder of the paper is organized as follow. Our system and assumptions are presented in Section II. The proposed scheme is elaborated in Section III. The performance analysis is shown in Section IV. Section V presents the implementation and experimental results using Amazon cloud platform. How to identify the corrupted copies is discussed in Section VI. Concluding remarks are given in Section VII.

System Design:

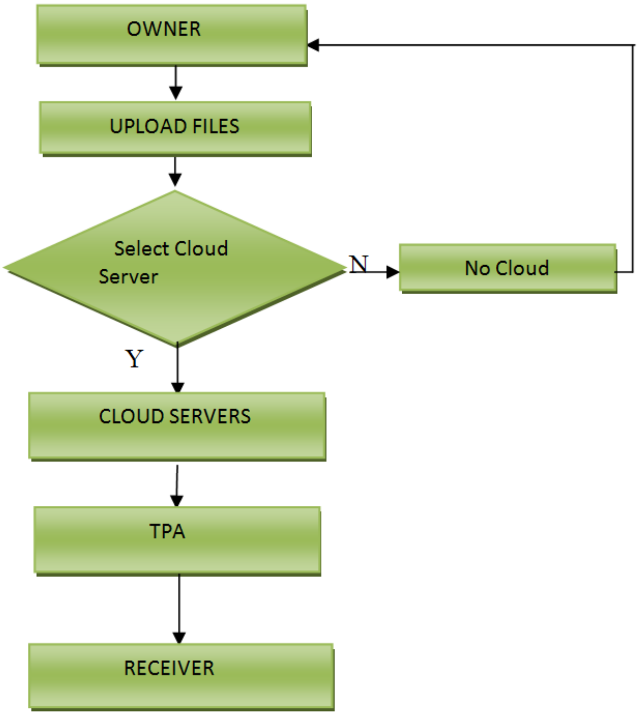

Figure 1: Flowchart

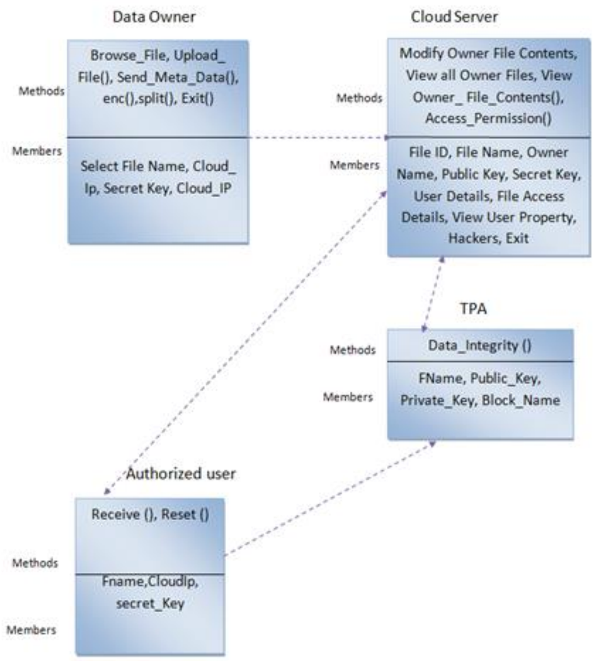
Data Owner
Methods
Browse_File, Upload_File(), Send_Meta_Data(), enc(),split(), Exit()
Members
Select File Name, Cloud_Ip, Secret Key, Cloud_IP
Cloud Server
Methods
Modify Owner File Contents, View all Owner Files, View Owner_ File_Contents(), Access_Permission()
Members
File ID, File Name, Owner Name, Public Key, Secret Key, User Details, File Access Details, View User Property, Hackers, Exit
TPA
Methods
Data_Integrity ()
Members
FName, Public_Key, Private_Key, Block_Name
Authorized user
Methods
Receive (), Reset ()
Members
Fname,CloudIp, secret_Key

Figure 2: Class Diagram

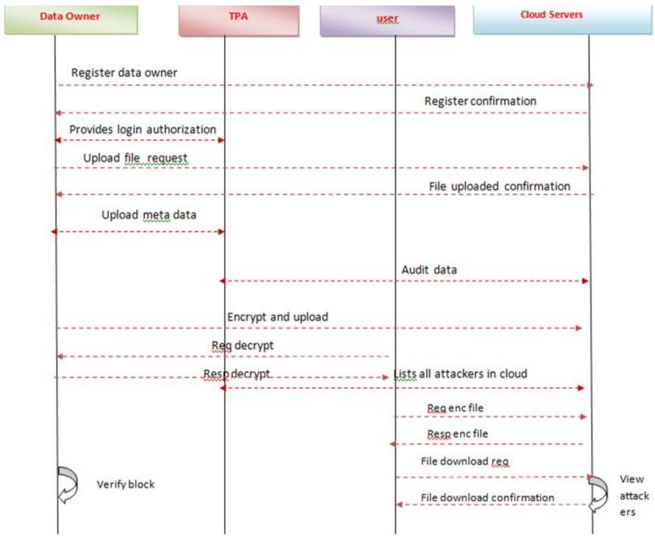

Figure 3: Sequence Diagram

III. RESULTS AND DISCUSSION

System Analysis

Existing System

Once the data has been outsourced to a remote CSP which may not be trustworthy, the data owners lose the direct control over their sensitive data. This lack of control raises new formidable and challenging tasks related to data confidentiality and integrity protection in cloud computing. The confidentiality issue can be handled by encrypting sensitive data before outsourcing to remote servers. As such, it is a crucial demand of customers to have a strong evidence that the cloud servers still possess their data and it is not being tampered with or partially deleted over time. Consequently, many researchers have focused on the problem of provable data possession (PDP) and proposed different schemes to audit the data stored on remote servers.

One of the core design principles of outsourcing data is to provide dynamic behavior of data for various applications. This means that the remotely stored data can be not only accessed by the authorized users, but also updated and scaled (through block level operations) by the data owner.

PDP schemes presented focus on only static or warehoused data, where the outsourced data is kept unchanged over remote servers. Examples of PDP constructions that deal with dynamic data. The latter are however for a single copy of the data file.

Proposed System

When verifying multiple data copies, the overall system integrity check fails if there is one or more corrupted copies. To address this issue and recognize which copies have been corrupted, we discuss a slight modification to be applied to the proposed scheme.

We propose a map-based provable multi-copy dynamic data possession (MB-PMDDP) scheme. This scheme provides an adequate guarantee that the CSP stores all copies that are agreed upon in the service contract. Moreover, the scheme supports outsourcing of dynamic data, i.e., it supports block-level operations such as block modification, insertion, deletion, and append. The authorized users, who have the right to access the owner's file, can seamlessly access the copies received from the CSP.

We give a thorough comparison of MB-PMDDP with a reference scheme, which one can obtain by extending existing PDP models for dynamic single-copy data.

We show the security of our scheme against colluding servers, and discuss a slight modification of the proposed scheme to identify corrupted copies.

In this work, we do not encode the data to be outsourced for the following reasons. First, we are dealing with dynamic data, and hence if the data file is encoded before outsourcing, modifying a portion of the file requires re-encoding the data file which may not be acceptable in practical applications due to high computation overhead.

Second, we are considering economically-motivated CSPs that may attempt to use less storage than required by the service contract through deletion of a few copies of the file. The CSPs have almost no financial benefit by deleting only a small portion of a copy of the file.

Third, and more importantly, unlike erasure codes, duplicating data files across multiple servers achieves scalability, which is a fundamental customer requirement in CC systems. A file that is duplicated and stored strategically on multiple servers – located at various geographic locations

IV.CONCLUSION

Outsourcing data to remote servers has become a growing trend for many organizations to Alleviate the burden of local data storage and maintenance. In this work we have studied the problem of creating multiple copies of dynamic data file and verifying those copies stored on untrusted cloud servers. We have proposed a new PDP scheme (referred to as MB-MDDP), which supports outsourcing of multi-copy dynamic data, where the data owner is capable of not only archiving and accessing the data copies stored by the CSP, but also updating and scaling these copies on the remote servers. To the best of our knowledge, the proposed scheme is the first to address *multiple* copies of *dynamic* data. The interaction between the authorized users and the CSP is considered in our scheme, where the authorized users can seamlessly access a data copy received from the CSP using a single secret key shared with the data owner. Moreover, the proposed scheme supports public verifiability, enables arbitrary number of auditing, and allows *possession-free* verification where the verifier has the ability to verify the data integrity even though he neither possesses nor retrieves the file blocks from the server. Through performance analysis and experimental results, we have demonstrated that the proposed MB-MDDP scheme outperforms the TB-MDDP approach derived from a class of dynamic single-copy PDP models. The TB-MDDP leads to high storage overhead on the remote servers and high computations on both the CSP and the verifier sides. The MB-MDDP scheme significantly reduces the computation time during the challenge-response phase, which makes it more practical for applications where a large number of verifiers are connected to the CSP causing a huge computation overhead on the servers. Besides, it has lower storage overhead on the CSP, and thus reduces the fees paid by the cloud customers. The dynamic block operations of the map-based approach are done with less communication cost than that of the tree-based approach. A slight modification can be done on the proposed scheme to support the feature of identifying the indices of corrupted copies. The corrupted data copy can be reconstructed even from a complete damage using duplicated copies on other servers. Through security analysis, we have shown that the proposed scheme is provably secure.

V. REFERENCES

[1]. G. Ateniese et al., "Provable data possession at untrusted stores," in Proc. 14th ACM Conf. Comput. Commun. Secur. (CCS), New York, NY,USA, 2007, pp. 598–609.

[2]. K. Zeng, "Publicly verifiable remote data integrity," in Proc. 10th Int. Conf. Inf. Commun. Secur. (ICICS), 2008, pp. 419–434.

[3]. Y. Deswarte, J.-J. Quisquater, and A. Saïdane, "Remote integrity checking," in Proc. 6th Working Conf. Integr. Internal Control Inf. Syst. (IICIS), 2003, pp. 1–11.

[4]. D. L. G. Filho and P. S. L. M. Barreto, "Demonstrating data possession and uncheatable data transfer," IACR (International Association for Cryptologic Research) ePrint Archive, Tech. Rep. 2006/150, 2006.

[5]. F. Sebé, J. Domingo-Ferrer, A. Martinez-Balleste, Y. Deswarte, and J.-J. Quisquater, "Efficient remote data possession checking in critical information infrastructures," IEEE Trans. Knowl. Data Eng., vol. 20, no. 8, pp. 1034–1038, Aug. 2008.

[6]. P. Golle, S. Jarecki, and I. Mironov, "Cryptographic primitives enforcing communication and storage complexity," in Proc. 6th Int. Conf. Financial Cryptograph. (FC), Berlin, Germany, 2003, pp. 120–135.

[7]. M. A. Shah, M. Baker, J. C. Mogul, and R. Swaminathan, "Auditing to keep online storage services honest," in Proc. 11th USENIX Workshop Hot Topics Oper. Syst. (HOTOS), Berkeley, CA, USA, 2007, pp. 1–6.

[8]. M. A. Shah, R. Swaminathan, and M. Baker, "Privacy-preserving audit and extraction of digital contents," IACR Cryptology ePrint Archive, Tech. Rep. 2008/186, 2008.

[9]. E. Mykletun, M. Narasimha, and G. Tsudik, "Authentication and integrity in outsourced databases," ACM Trans. Storage, vol. 2, no. 2, pp. 107–138, 2006.

[10]. G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in Proc. 4th Int. Conf. Secur. Privacy Commun. Netw. (SecureComm), New York, NY, USA, 2008, Art. ID 9.