ISSN (Online): 2456-0448
International Journal Of Innovative Research In Management, Engineering And Technology
Vol. 2, Issue 7, July 2017

An Efficient Cloud Storage Auditing and data storage in Cloud Computing with Verifiable Outsourcing of Key Updates

[1] T.Kamalakannan, Mca.Mphil.Phd.Net, [2]John.Kest
[1] Professor&Head of BCA Vels University
[2] Student of BCA Vels University

Abstract: ***Key-exposure resistance has always been an important issue for in-depth cyber defence in many security applications. Recently, how to deal with the key exposure problem in the settings of cloud storage auditing has been proposed and studied. To address the challenge, existing solutions all require the client to update his secret keys in every time period, which may inevitably bring in new local burdens to the client, especially those with limited computation resources such as mobile phones. In this paper, we focus on how to make the key updates as transparent as possible for the client and propose a new paradigm called cloud storage auditing with verifiable outsourcing of key updates. In this paradigm, key updates can be safely outsourced to some authorized party, and thus the key-update burden on the client will be kept minimal. Specifically, we leverage the third-party auditor (TPA) in many existing public auditing designs, let it play the role of authorized party in our case, and make it in charge of both the storage auditing and the secure key updates for key-exposure resistance.***

Keywords- ***Cloud storage; outsourcing computing; cloud storage auditing; key update; verifiability.***

I. INTRODUCTION

Cloud computing is the delivery of computing services over the Internet[1] Cloud services allow individuals and businesses to use software and hardware that are managed by third parties at remote locations. Examples of cloud services include online file storage, social networking sites, webmail, and online business applications [2] [3]. The cloud computing model allows access to information and computer resources from anywhere that a network connection is available [5]. Cloud computing provides a shared pool of resources, including data storage space, networks, computer processing power, and specialized corporate and user applications[4].

The following definition of cloud computing has been developed by the U.S. National Institute of Standards and Technology (NIST):

Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [6]. This cloud model promotes availability and is composed of five essential characteristics, three service models, and four deployment models.

While there are benefits, there are privacy and security concerns too [7]. Data is travelling over the Internet and is stored in remote locations. In addition, cloud providers often serve multiple customers simultaneously. All of this may raise the scale of exposure to possible breaches, both accidental and deliberate [8].

Concerns have been raised by many that cloud computing may lead to “function creep” — uses of data by cloud providers that were not anticipated when the information was originally collected and for which consent has typically not been obtained [9]. Given how inexpensive it is to keep data, there is little incentive to remove the information from the cloud and more reasons to find other things to do with it [10].

Security issues, the need to segregate data when dealing with providers that serve multiple customers, potential secondary uses of the data—these are areas that organizations should keep in mind when considering a cloud provider and when negotiating contracts or reviewing terms of service with a cloud provider[11]. Given that the organization transferring this

information to the provider is ultimately accountable for its protection, it needs to ensure that the personal information is appropriate handled [12].

Inourdesign,TPAonlyneedstoholdan encrypted version of the client's secret key, while doing all these burdensome tasks on behalf of the client. The client only needs to download the encrypted secret key from the TPA when uploading new files to cloud. Besides, our design also equips the client with capability to further verify the validity of the encrypted secret keys provided by TPA. All these salient features are carefully designedtomakethewholeauditingprocedurewithkeyexposure resistance as transparent as possible for the client. We formalize the definition and the security model of this paradigm. The security proof and the performance simulation show that our detailed design instantiations are secure and efficient.

The key exposure problem, as another important problem in cloud storage auditing, has been considered [13] recently. The problem itself is non-trivial by nature. Once the client's secret key for storage auditing is exposed to cloud, the cloud is able to easily hide the data loss incidents for maintaining its reputation, even discard the client's data rarely accessed for saving the storage space. The authors in [14] constructed a cloud storage auditing protocol with key-exposure resilience by updating the user's secret keys periodically. In this way, the damage of key exposure in cloud storage auditing can be reduced. But it also brings in new local burdens for the client because the client has to execute the key update algorithm in each time period to make his secret key move forward. For some clients with limited computation resources, they might not like doing such extra computations by themselves in each time period. It would be obviously more attractive to make key updates as transparent as possible for the client, especially in frequent key update scenarios. In this paper, we consider achieving this goal by outsourcing key updates.

II IMPLEMENTATION

2. EXSISTING SYSTEM

- Yu et al. constructed a cloud storage auditing protocol with key-exposure resilience by updating the user's secret keys periodically. In this way, the damage of key exposure in cloud storage auditing can be reduced. But it also brings in new local burdens for the client because the client has to execute the key update algorithm in each time period to make his secret key move forward.

- For some clients with limited computation resources, they might not like doing such extra computations by themselves in each time period. It would be obviously more attractive to make key updates as transparent as possible for the client, especially in frequent key update scenarios.

- Wang et al. proposed a public privacy-preserving auditing protocol. They used the random masking technique to make the protocol achieve privacy preserving property.

2.1 DISADVANTAGES OF EXISTING SYSTEM:

- Existing system don't like auditing protocol with verifiable outsourcing of key updates.

- Third party has the access to see client's secret key without encryption.

- No verification system available for client's for to check validity of the encrypted secret keys when downloading them from the TPA

- All existing auditing protocols are all built on the assumption that the secret key of the client is absolutely secure and would not be exposed.

2.2 PROPOSED SYSTEM:

The main contributions are as follows:

(1) We propose a new paradigm called cloud storage auditing with verifiable outsourcing of key updates. In this new paradigm, key-update operations are not performed by the client, but by an authorized party. The authorized party holds an encrypted secret key of the client for cloud storage auditing and updates it under the encrypted state in each time period. The client downloads the encrypted secret key from the authorized party and decrypts it only when he would like to upload new files to cloud. In addition, the client can verify the validity of the encrypted secret key.

(2) We design the first cloud storage auditing protocol with verifiable outsourcing of key updates. In our design, the thirdparty auditor (TPA) plays the role of the authorized party who is in charge of key updates.

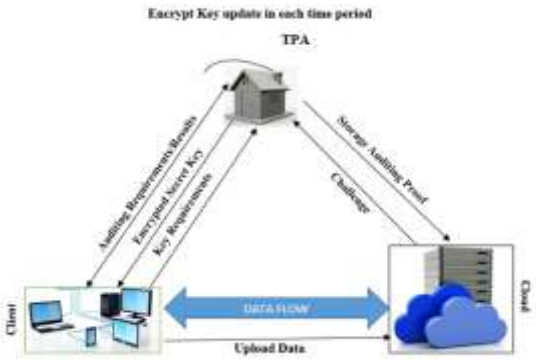
(3) We formalize the definition and the security model of the cloud storage auditing protocol with verifiable outsourcing of key updates. We also prove the security of our protocol in the formalized security model and justify its performance by concrete implementation.

2.3 ADVANTAGES OF PROPOSED SYSTEM:

- The TPA does not know the real secret key of the client for cloud storage auditing, but only holds an encrypted version. In the detailed protocol, we use the blinding technique with homomorphic property to form the encryption algorithm to encrypt the secret keys held by the TPA. It makes our protocol secure and the decryption operation efficient.
- Meanwhile, the TPA can complete key updates under the encrypted state. The client can verify the validity of the encrypted secret key when he retrieves it from the TPA.
- The client downloads the encrypted secret key from the authorized party and decrypts it only when he would like to upload new files to cloud. In addition, the client can verify the validity of the encrypted secret key.
- Cloud storage auditing protocol with verifiable outsourcing of key updates
- The client can verify the validity of the encrypted secret key when he retrieves it from the TPA
- The security model of the cloud storage auditing protocol with verifiable outsourcing of key updates.

III. SYSTEM ARCHITECTURE

3. System Architecture



3.1 MODULES

We have 3 main modules in this project;

1. Client Module
2. Cloud Module
3. Third Party Auditor (TPA)

3.1.1 Client:

The client is the owner of the files that are uploaded to cloud. The total size of these files is not fixed,that is, the client can upload the growing files to cloud in different time points.

3.1.2 Cloud:

The cloud stores the client's files and provides download service for the client.

3.1.3 TPA:

The TPA plays two important roles: the first is to audit the data files stored in cloud for the client; the second is to update the encrypted secret keys of the client in each time period.

IV. RELATED WORKS

To address the above-mentioned privacy is- sue in outsourcing functional computation over public multi- dimensional dataset, we propose an Efficient Privacy- preserving Outsourced Computation Framework over Public Data, called EPOC, which protects privacy of both the function and its output. Specifically, the main contributions of this paper are fourfold.

We consider a cloud system composed of three major entities: the cloud server, group users and the third-party auditor (TPA). The cloud server is the party that provides data storage services to group users. Group users consist of a number of general users and a master user, who is the owner of the shared data and manages the membership of other group users. All group users can access and modify data. The TPA refers to any party that checks the integrity of data being stored on the cloud. As our proposed scheme allows public integrity auditing, the TPA can actually be any cloud user as long as he/she has access to the public keys. Once the TPA detects a data corruption during the auditing process, he/she will report the error to group users. In our design, data can be uploaded/created by either the master user or other group users. We assume data are stored in form of files which are further divided into a number of blocks. For integrity auditing, each data block is attached with an authentication tag that is originally generated by the master user. When a user adds or modifies a block, he/she (the user) updates the corresponding authentication tag with his/her own secret key without contacting the master user.

To ensure data integrity can be audited publicly, users need to compute signatures on all the blocks in shared data. Different blocks are signed by different users due to data modifications performed by different users. For security reasons, once a user is revoked from the group, the blocks, which were previously signed by this revoked user must be re-signed by an existing user. The straightforward method, which allows an existing user to download the corresponding part of shared data and re-sign it during user revocation, is inefficient due to the large size of shared data in the cloud. In this paper, we propose a novel public auditing mechanism for the integrity of shared data with efficient user revocation in mind. By utilizing proxy re-signatures, we allow the cloud to re-sign blocks on behalf of existing users during user revocation, so that existing users do not need to download and re-sign blocks by themselves. In addition, a public verifier is always able to audit the integrity of shared data without retrieving the entire data from the cloud, even if some part of shared data has been re-signed by the cloud.

An efficient and secure dynamic auditing protocol, which can meet the above listed requirements. To solve the data privacy problem, our method is to generate an encrypted proof with the challenge stamp by using the Bilinearity property of the bilinear pairing, such that the auditor cannot decrypt it but can verify the correctness of the proof. Without using the mask

technique, our method does not require any trusted organizer during the batch auditing for multiple clouds. On the other hand, in our method, we let the server compute the proof as an intermediate value of the verification, such that the auditor can directly use this intermediate value to verify the correctness of the proof. Therefore, our method can greatly reduce the computing loads of the auditor by moving it to the cloud server.

In the cloud paradigm, by putting the large data files on their mote servers, the clients can be relieved of the burden of storage and computation. As clients no longer possess their data locally, it is of critical importance for the clients to ensure that their data are being correctly stored and maintained. That is, clients should be equipped with certain security means so that they can periodically verify the correctness of the remote data even without the existence of local copies. In case that clients do not necessarily have the time, feasibility or resources to monitor their data, they can delegate the monitoring task to a trusted TPA.

Cloud computing has been envisioned as the next-generation architecture of enterprise IT. In contrast to traditional enter- prise IT solutions, where the IT services are under proper physical, logical, and personnel controls, cloud computing moves the application software and databases to servers in large data centers on the Internet, where the management of the data and services are not fully trustworthy. This unique attribute raises many new security challenges in areas such as software and data security, recovery, and privacy, as well as legal issues in areas such as regulatory compliance and audit- ing, all of which have not been well understood. In this article we focus on cloud data storage security. We first present a network architecture for effectively describing, developing, and evaluating secure data storage problems. We then suggest a set of systematically and cryptographically desirable proper- ties for public auditing services of dependable cloud data storage security to become a reality. Through in-depth analysis, some existing data storage security building blocks are examined. The pros and cons of their practical implications in the context of cloud computing are summarized. Further challenging issues for public auditing services that need to be focused on are discussed too. We believe security in cloud computing, an area full of challenges and of paramount importance, is still in its infancy now but will attract enormous amounts of research effort for many years to come.

V. ALGORITHM

1. Encryption/Decryption (encrypt the secret keys held by the TPA).

2. Random values Generation algorithm.

5.1 Encryption algorithm

- ✓ Or conventional / private-key / single-key
- ✓ Sender and recipient share a common key
- ✓ All classical encryption algorithms are private-key was only type prior to invention of public-key in 1970's and by far most widely used

5.2 Decryption algorithm

- ✓ Step 1: Multiply last 5 digits of the cipher text by the Key
- ✓ Step 2: Add first 3 digits of the cipher text with the result produced in the previous step
- ✓ Step 3: If the result produced in the previous step i.e. step 2 is not an 8-bit number we need to make it an 8- bit number
- ✓ Step 4: Reverse the number to get the original text i.e. the plain text.

5.3 Random values Generation algorithm

A random-number generator (RNG) is a computational or physical device designed to generate a sequence of numbers or symbols that cannot be reasonably predicted better than by a random chance.

1.A number selected from a known set of numbers in such a way that each number in the set has the same probability of occurrence.

2. A number obtained by chance.

3. One of a sequence of numbers considered appropriate for satisfying certain statistical tests or believed to be free from conditions that might bias the result of a calculation.

Random Number Generators (RNGs) used for cryptographic applications typically produce a sequence of zero and one bits that may be combined into sub-sequences or blocks of random numbers. There are two basic classes: deterministic and nondeterministic. A deterministic RNG consists of an algorithm that produces a sequence of bits from an initial value called a seed. A nondeterministic RNG produces output that is dependent on some unpredictable physical source that is outside human control. There are no FIPS Approved nondeterministic random number generators.

VI REFERENCES

[1] M.J. Atallah, and K.B. Frikken, "Securely outsourcing linear algebra computations," Proceedings of the 5th ACM Symposium on Information, pp.48-59, 2010.

[2] X. Chen, J. Li, X. Huang, et al., "Secure Outsourced Attribute-based Signatures," IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 12, pp. 3285- 3294, 2014.

[3] F. Zhang, X. Ma, S. Liu, "Efficient computation out- sourcing for inverting a class of homomorphic functions," Information Sciences, vol. 286, pp. 19-28, 2014.

[4] B. Lynn, The pairing-based cryptographic library, online at http://crypto.Stanford.edu/pbc/, 2015.

[5] B. Chevallier-Mames, J. Coron, N. McCullagh, D. Nac- cache, and M. Scott, "Secure delegation of elliptic-curve pairing," Proc. CARDIS 2010, pp.24-35, 2010.

[6] J. Yuan and S. Yu, "Public Integrity Auditing for Dynamic Data Sharing With Multiuser Modification," IEEE Transactions on Information Forensics and Security, vol. 10, no. 8, pp. 1717-1726, Aug. 2015.

[7] J. Yu, K. Ren, C. Wang, V. Varadharajan, "Enabling Cloud Storage Auditing with Key-Exposure Resistance," IEEE Transactions on Information Forensics and Security. vol. 10, no. 6, pp. 1167-1179, Jun. 2015.

[8] B. Wang, B. Li, and H. Li, "Public auditing for shared data with efficient user revocation in the cloud," IEEE INFOCOM 2013, pp. 2904-2912, 2013.

[9] C.Wang, K. Ren, and J.Wang, "Secure and practical outsourcing of linear programming in cloud computing," IEEE INFOCOM 2011, pp. 820-828, 2011.

[10] X. Chen, J. Li, J. Ma, Q. Tang, and W. Lou, "New Algorithms for Secure Outsourcing of Modular Exponentiations," Proc. 17th European Symposium on Research in Computer Security, pp. 541-556, 2012.

[11] Y. Zhu, H. Wang, Z. Hu, G. J. Ahn, H. Hu, and S. S. Yau, "Efficient Provable Data Possession for Hybrid Clouds," Proc. 17th ACM Conference on Computer and Communications Security, pp. 756-758, 2010.

[12] C. Wang, K. Ren, W. Lou, and J. Li, "Toward Publicly Auditable Secure Cloud Data Storage Services," IEEE Network, vol. 24, no. 4, pp. 19-24, July/Aug. 2010.

[13] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing," IEEE Trans. Parallel and Distributed Systems, vol. 22, no. 5, pp. 847-859, May 2011.

[14] K. Yang and X. Jia, "Data Storage Auditing Service in Cloud Computing: Challenges, Methods and opportunities," World Wide Web, vol. 15, no. 4, pp. 409-428, 2012.

[15] Y. Zhu, H.G. Ahn, H. Hu, S.S. Yau, H.J. An, and C.J. Hu, "Dynamic Audit Services for Outsourced Storages in Clouds," IEEE Trans. on Services Computing, vol. 6, no.2, pp. 409-428, 2013.

[16] K. Yang and X. Jia, "An efficient and secure dynamic auditing protocol for data storage in cloud computing," IEEE Trans. Parallel and Distributed Systems, Vol. 24, No. 9, pp. 1717-1726, 2013.

[17] H. Wang, "Proxy Provable Data Possession in Public Clouds," IEEE Trans. Services Computing, Vol. 6, no. 4, pp. 551-559, 2013.

[18] C. Wang, S. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Secure Cloud Storage," IEEE Trans. Computers, Vol. 62, No. 2, pp. 362- 375, 2013.

[19] B. Wang, B. Li and H. Li. Oruta, "Privacy-Preserving Public Auditing for Shared Data in the Cloud," IEEE Transactions on Cloud Computing, Vol.2, pp. 43-56, 2014.

[20] C. Erway, A. Kpc, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," Proc. of the 16th ACM conference on Computer and communications security, pp. 213-222, 2009.