IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 3, March 2017

Implementation of Dynamic Auditing Protocol for Data Storage in Cloud Computing

Patil Vaishali[1], Wagh Shrikant[2], Panchal Amol[3], Prof. G.M. Poddar[4]

Student, Information Technology Deprt., Gangamai College of Engineering, Dhule, India [1,2,3]

Professor, Information Technology Deprt., Gangamai College of Engineering, Dhule, India [4]

Abstract: Cloud computing is growing now days, all physical systems are going to be history in coming years as cloud computing provides the virtualize framework of all i.e. software, hardware etc. The one of the most efficient use of cloud is data storage on cloud server on pay as you go scheme. But as its good to hear there are some challenging aspects behind this cloud data storage as per end users perspective. How end users know their data is secure on cloud server? How they satisfied that the data is not tampered and successfully updated after performing some operation over it? Here the Trusted Third Party auditor comes in picture and using auditing framework he satisfy end users that there data is secure over server and successfully updated. Thus, an efficient and secure dynamic auditing protocol is desired to convince data owners that the data are correctly stored in the cloud. In this paper, we first design an auditing framework for cloud storage systems and propose an efficient and privacy-preserving auditing protocol. Then, we extend our auditing protocol to support the data dynamic operations, which is efficient and provably secure in the random oracle mode.

Keywords: Cloud computing, dynamic auditing, privacy-preserving auditing, batch auditing, Storage auditing.

I. INTRODUCTION

Cloud Computing has been envisioned as the next generation information technology (IT) architecture for enterprises, due to its long list of unprecedented advantages in the IT history: on-demand self-service, ubiquitous network access, location independent resource pooling, rapid resource elasticity, usage-based pricing and transference of risk. As a disruptive technology with profound implications, Cloud Computing is transforming the very nature of how businesses use information technology. One fundamental aspect of this paradigm shifting is that data is being centralized or outsourced to the Cloud. From users' perspective, including both individuals and IT enterprises, storing data remotely to the cloud in a flexible on-demand manner brings appealing benefits: relief of the burden for storage management, universal data access with location independence, and avoidance of capital expenditure on hardware, software, and personnel maintenances, etc. While Cloud Computing makes these advantages more appealing than ever, it also brings new and challenging security threats towards users' outsourced data.

Since cloud service providers (CSP) are separate administrative entities, data outsourcing is actually relinquishing user's ultimate control over the fate of their data. As a result, the correctness of the data in the cloud is being put at risk due to the following reasons. First of all, although the infrastructures under the cloud are much more powerful and reliable than personal computing devices, they are still facing the broad range of both internal and external threats for data integrity. Examples of outages and security breaches of noteworthy cloud services appear from time to time.
Secondly, there do exist various motivations for CSP to behave unfaithfully towards the cloud users regarding their outsourced data status.

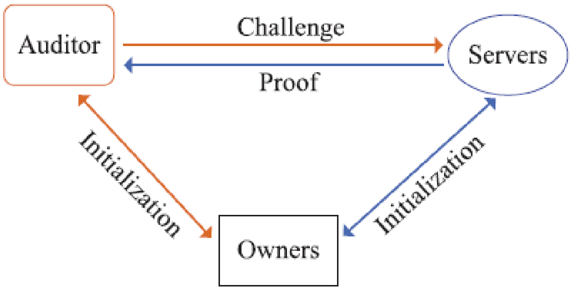

Fig. 1 System Model

For examples, CSP might reclaim storage for monetary reasons by discarding data that has not been or is rarely accessed, or even hide data loss incidents to maintain a reputation. In short, although outsourcing data to the cloud is economically attractive for long-term large-scale storage, it does not immediately offer any guarantee on data integrity and availability. This problem, if not properly addressed, may impede the success of cloud architecture. As users no longer physically possess the storage of their data, traditional cryptographic primitives for the purpose of data security protection cannot be directly adopted. In particular, simply downloading all the data for its integrity verification is not a practical solution

DOI10.17148/IJARCCE.2017.63236

due to the expensiveness in I/O and transmission cost across the network. Besides, it is often insufficient to detect the data corruption only when accessing the data, as it does not give users correctness assurance for those unaccessed data and might be too late to recover the data loss or damage. Considering the large size of the outsourced data and the user's constrained resource capability, the tasks of auditing the data correctness in a cloud environment can be formidable and expensive for the cloud users. Moreover, the overhead of using cloud storage should be minimized as much as possible, such that a user does not need to perform too many operations to use the data (in additional to retrieving the data). In particular, users may not want to go through the complexity in verifying the data integrity. Besides, there may be more than one user accesses the same cloud storage, say in an enterprise setting. For easier management, it is desirable that cloud only entertains verification request from a single designated party.

II. LITERATURE SURVEY

Different authors done the reaserach by using the different techinques and algorithms.

Q. Wang al. proposed a dynamic auditing protocol that can support the dynamic operations of the data on the cloud servers, but this method may leak the data content to the auditor because it requires the server to send the linear combinations of data blocks to the auditor.

K Ren al. extended their dynamic auditing scheme to be privacy preserving and support the batch auditing for multiple owners. However, due to the large number of data tags, their auditing protocols may incur a heavy storage overhead on the server.

Liu Yang al. proposed a secure audit scheme supporting dynamic operation and transparent verification. Utilizing BLS short signature as well as the sequence-enforced B+ Hash Tree structure, the audit scheme is more effective. The scheme introduces an organizer in the auditing process to prevent the TPA from getting any information about the data's location. Thus, the scheme is completely transparent for TPA. Meanwhile, the scheme utilizes random mask and bilinear aggregate signature technology to realize privacy protection and batch audit.

Shacham et al. provided an improved POR model with stateless verification. They also proposed a MAC-based private verification scheme and the first public verification scheme in the literature that based on BLS signature scheme.

Ateniese, et al. proposed a second scheme, the generation and verification of integrity proofs are similar to signing and verification of BLS signatures. When wielding the same security strength (say, 80-bit security), a BLS signature (160 bit) is much shorter than an RSA signature (1024 bit), which is a desired benefit for a POR scheme.

R.D. Pietro, et al. proposed the concepts of PDP and POR were in fact unified under this new compact POR model. Ateniese, et al. extended their scheme for enhanced scalability, but only partial data dynamics and a predefined number of challenges is supported.

Erway, et al. proposed the first PDP scheme based on skip list that can support full dynamic data updates. However, public auditability and variable-sized file blocks are not supported by default.

Q. Wang, et al. proposed a scheme based on BLS signature that can support public auditing (especially from a thirdparty auditor, TPA) and full data dynamics, which is one ofthe latest works on public data auditing with dynamics support. However, their scheme lacks support for finegrainedupdate and authorized auditing which are the main focuses of our work.

C. Wang et al. proposed a scheme to add a random masking technology on top of to ensure the TPA cannot infer the raw data file from a series of integrity proofs.In their scheme, they also incorporated a strategy to segment file blocks into multiple sectors. However, the use of this strategy was limited to trading-off storage cost with communication cost.

Surya Nepal et al. proposed a secure cloud storage service architecture with the focus on Data Integrity as a Service (DIaaS) based on the principles of Service- Oriented Architecture and Web services. Our approach not only releases the burdens of data integrity management from a storage service by handling it through an independent third party data Integrity Management Service (IMS), but also reduces the security risk of the data stored in the storage services by checking the data integrity with the help of IMS. We define data integrity protocols for a number of different scenarios, and demonstrate the feasibility of the proposed architecture, service and protocols by implementing them on a public cloud, Amazon S3. We also study the impact of our proposed protocols on the performance of the storage service and show that the benefits of our approach outweigh the little penalty on the storage service performance.

III.CHARACTERISTICS OF AUDITING PROTOCOLS

While designing this data integrity checking protocol, they must satisfy some requirements:

- Highly private: The TPA should not gain knowledge of the original user data during the auditing process.
- Data dynamic: The clients must be able to perform operations on data files like insert, alter and delete while maintaining
- data correctness.
- Open verifiability: Anyone, not just the clients, must be allowed to verify the integrity of data.
- Block free verification: Challenged file blocks should not be retrieved by the verifier during verification process.
- No restriction of queries: The verifier may be allowed to use unlimited number of queries in the challenge-response protocol for data verification.

DOI10.17148/IJARCCE.2017.63236

IV. SECURITY RISKS IN CLOUD COMPUTING

As the cloud services have been built over the Internet, any issue that is related to internet security will also affect cloud services. Users of online data sharing or network facilities are aware of the potential loss of privacy. According to a recent IDC survey, the top challenge for 74% of CIOs in relation to cloud computing is security. Protecting private and important information such as credit card details or patients' medical records from attackers or malicious insiders is of critical importance. Moving databases to a large data centre involves many security challenges such as virtualization vulnerability, accessibility vulnerability, privacy and control issues related to data accessed from a third party, integrity, confidentiality, and data loss or theft. Subashini and Kavitha present some fundamental security challenges, which are data storage security, application security, data transmission security, and security related to third-party resources.

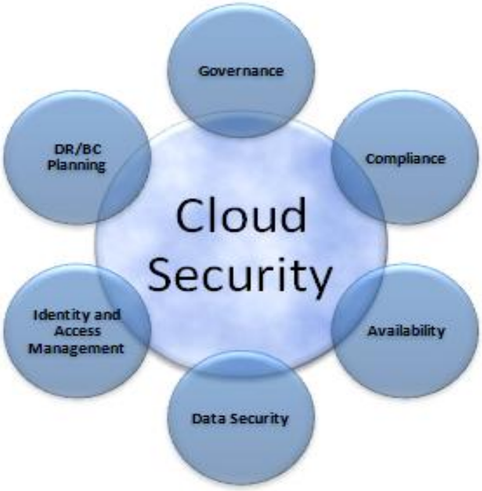

Fig. 2 Cloud Security

Each category includes several potential security problems, resulting ina classification with subdivisions that highlights the main issues identified in the base references.

V. PROPOSED DYNAMIC AUDITING PROTOCOL

In cloud data storage system, the data owners perform updating frequently. As per the definition of the auditing protocol,they should fulfil to handle the dynamic data and static data. But the dynamic operations make auditing protocol insecure, as many attacks server can make to track the data or to tamper the data as it is easier to crack update operation. Server may undergoes the following attacks which are The CSP may not update correctly the clients data on the server and may use the flesh data to pass the auditing or The client updates the data to the current version, the server may get enough information from the dynamic operations to track the data tag. If the server could track the data tag, it can use any data and its data tag to auditing and make fool to auditor easily.
To overcome this drawback in this proposed scheme the Index Table is maintained to keep the detailed information of the data stored. This table consists, the Index denotes the current FID of data block, data component.

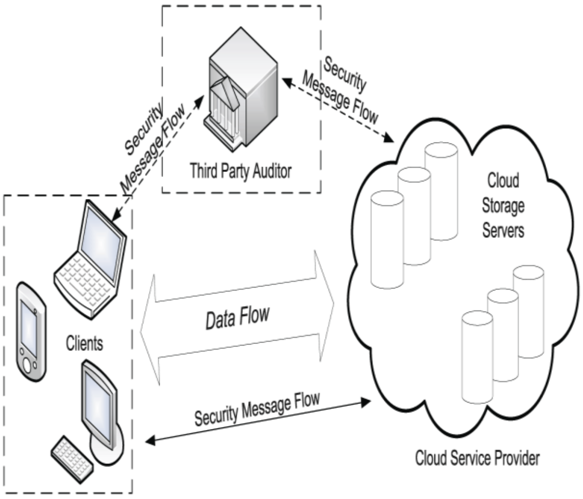

Fig. 3 The architecture of cloud data storage service

The original block number of data block and current version number of data block, the timestamp is used for generating the data tag. This Table is created by the owner during the initialization phase and the auditor manage this table afterwards. When the owner completes the dynamic data operations, it sends an update message to the auditor for updating the table which is with auditor. Once whole table is updated with auditor the auditor sends the result to the owner for the confirmation that the data on the server and the all information in Table on the auditor side are updated successfully.

VI. PRIVACY-PRESERVING AUDITING PROTOCOL

The data privacy is an important requirement in the design of auditing protocol in cloud storage systems. A storage auditing protocol consists of the following five algorithms:

1. KeyGen(sk_h, sk_t, pk_t): This key generation algorithm takes no input other than the implicit security parameter £. It outputs a secret hash key skh and a pair of secret-public tag key (sk_t, pk_t).
2. TagGen(M, sk_h, sk_t):. The tag generation algorithm takes as inputs an encrypted fileM, the secret tag key skt, and the secret hash key skh. For each data block mi, it computes a data tag ti based on skh and skt. It outputs a set of data tags T=$\{t_i\}_{i€[1,n]}$.
3. Chall(M_{info})-> C. The challenge algorithm takes as input the abstract information of the data Minfo (e.g., file identity, total number of blocks, version number, time stamp, etc.). It outputs a challenge C.

DOI10.17148/IJARCCE.2017.63236

4. Prove(M, T, C)-> P. The prove algorithm takes as inputs the file M, the tags T, and the challenge from the auditor C. It outputs a proof P.
5. Verify(C,P,sk$_h$,pk$_h$, M$_{info}$)->0/1: The verification algorithm takes as inputs P from the server, the secret hash key skh, the public tag key pkt, and the abstract information of the data Minfo. It outputs the auditing result as 0 or 1.

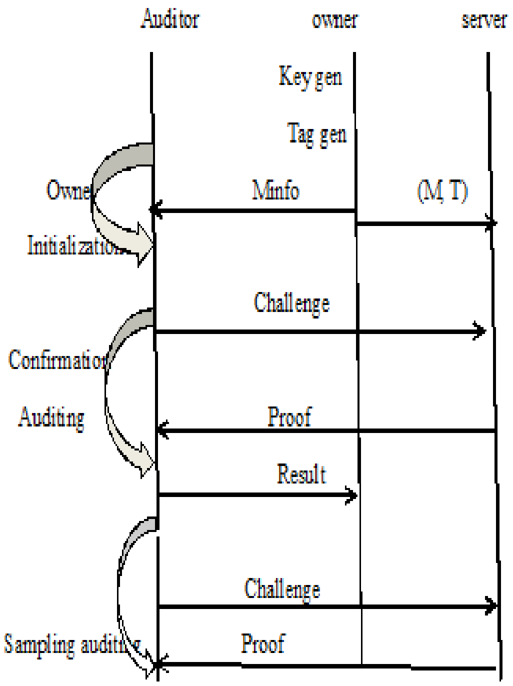

Fig. 4 Framework of our privacy-preserving auditing protocol

Advantage:
1. Auditing protocol ensures the data privacy by using cryptography method and the Bilinearity property of the bilinear pairing, instead of using the mask technique .This protocol incurs less communication cost between the auditor and the server. It also reduces the computing loads of the auditor by moving it to the server.
2. Also it supports data dynamic operations, which is efficient and provably secure in the random oracle model.
3. We further extend our auditing protocol to support batch auditing for not only multiple clouds but also multiple owners.
Multicloud batch auditing does not require any additional trusted organizer. The multiowner batch auditing can greatly improve the auditing performance, especially in large-scale.

Disadvantage:
1. This protocol is not suitable when data loss occur during auditing process. Especially when sending encrypted challenge stamp to the auditor and to the cloud server.
2. Also it can't be solving the situation when multiple owners periodically updated.

VII. BATCH AUDITING FOR MULTI CLOUD

In cloud computing auditing helps the owners to check the data integrity on the cloud servers. Due to the large number of data owners, the auditor may receive many auditing requests from multiple data owners. In this situation, it would greatly improve the system performance, if the auditor could combine these auditing requests together and only conduct the batch auditing for multiple owners simultaneously. The previous work cannot support the batch auditing for multiple owners.
The mask technique to ensure the data privacy, such that it requires an additional trusted organizer to send a commitment to the auditor during the commitment phase in multicloud batch auditing. In our method, we apply the encryption method with the bilinearity property of the bilinear pairing to ensure the data privacy, rather than the mask technique. Thus, our multicloud batch auditing protocol does not have any commitment phase, such that our method does not require any additional trusted organizer.

VIII. CONCLUSION

The proposed auditing protocol is more efficient and secure. It protects the data privacy against the auditor by combining the cryptography method with the bilinearity property of bilinear paring, rather than using the mask technique. Thus, our multicloud batch auditing protocol does not require any additional organizer.Cloud-based mechanisms are required to ensure data security and privacy, and to fulfill the regulatory and audit requirements of enterprises. Economical and inherently secure dynamic auditing protocol is proposed which protects the information privacy against the auditor and data loss by combining the cryptography method with the additive property of bilinear paring with time stamp, rather than using simple bilinear pairing without timestamp value. Thus, multicloud batch auditing protocol does not need any extra organizer. Batch auditing protocol can even support the batch auditing for multiple owners. Also, it reduces the computation time compared to the previous auditing scheme. It uses the best fragmentation technique so that the data tag generation is reduced. Thus, the storage space is preserved. In this technique, even the auditor is not aware about the actual form of data that is stored in the cloud.

REFERENCES

[1] Kan Yang, "An Efficient and Secure Dynamic Auditing Protocol for Data Storage in Cloud Computing", IEEE Transactions On Parallel And Distributed Systems, Vol. 24, No. 9, September 2013.
[2] Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), International Congress, 2017.
[3] Liu Yang, Lili Xia, "An Efficient and Secure Public Batch Auditing Protocol for Dynamic Cloud Storage Data", Computer Symposium (ICS),International 2016.
[4] Hao Jin, Hong Jiang, "Dynamic and Public Auditing with Fair Arbitration for Cloud Data", IEEE Transactions on Cloud Computing (Volume: PP, Issue: 99), 2016.

[5] Jiangtao Li, Lei Zhang, Joseph K. Liu, Haifeng Qian, Zheming Dong, "Privacy-Preserving Public Auditing Protocol for Low-Performance End Devices in Cloud", IEEE Transactions on Information Forensics and Security, Volume: 11, Issue: 11, Nov. 2016.

[6] P. Mell and T. Grance, “The NIST Definition of Cloud Computing,” technical report, Nat’l Inst. of Standards and Technology, 2009.

[7] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R.H. Katz, A. Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, “A View of Cloud Computing,” Comm. ACM, vol. 53, no. 4, pp. 50-58, 2010.

[8] T. Velte, A. Velte, and R. Elsenpeter, Cloud Computing: A Practical Approach, first ed., ch. 7. McGraw-Hill, 2010.

[9] J. Li, M.N. Krohn, D. Mazie`res, and D. Shasha, “Secure Untrusted Data Repository (SUNDR),” Proc. Sixth Conf. Symp. Operating Systems Design Implementation, pp. 121-136, 2004.

[10] G.R. Goodson, J.J. Wylie, G.R. Ganger, and M.K. Reiter, “Efficient Byzantine-Tolerant Erasure-Coded Storage,” Proc. Int’l Conf. Dependable Systems and Networks, pp. 135-144, 2004.

[11] V. Kher and Y. Kim, “Securing Distributed Storage: Challenges, Techniques, and Systems,” Proc. ACM Workshop Storage Security and Survivability (StorageSS), V. Atluri, P. Samarati, W. Yurcik, L Brumbaugh, and Y. Zhou, eds., pp. 9-25, 2005.

DOI10.17148/IJARCCE.2017.63236