A Survey on Automatic Protocol Blocker for Privacy -Preserving Public Auditing in Cloud Computing

Rohini Kadlag, Rohit Devikar

Abstract—Cloud Computing is nothing but specific style of computing where everything from computing power to infrastructure, business apps are provided "as a service". In cloud, shared resources, softwares and information is provided as a metered service over the network. By data outsourcing, users can be relieved from the burden of local data storage and maintenance. However, the fact that users no longer have physical possession of the possibly large size of outsourced data makes the data integrity protection in Cloud Computing a very challenging and potentially formidable task, especially for users with constrained computing resources and capabilities. Thus, enabling public auditability for cloud data storage security is of critical importance so that users can resort to an external audit party to check the integrity of outsourced data when needed. In particular, we consider the task of allowing a third party auditor (TPA), on behalf of the cloud client, to verify the integrity of the dynamic data stored in the cloud. In this paper we are extending the previous system by using automatic blocker for privacy preserving public auditing for data storage security in cloud computing. We utilize the public key based homomorphic authenticator and uniquely integrate it with random mask technique and automatic blocker. In particular, to achieve efficient data dynamics, we improve the existing proof of storage models by manipulating the block tag authentication. Thus, TPA eliminates the involvement of the client through the auditing of whether his Data stored in the Cloud are indeed intact, which can be important in achieving economies of scale For Cloud Computing.

Index Terms—Cloud computing, CSP, Data storage, Data dynamics, Protocol Blocker, Public auditability, TPA.

I. INTRODUCTION

Cloud Computing is the dreamed vision of computing as a public utility. It is a model for enabling convenient, on-demand network access to shared pool of configurable computing resources (e.g. networks, servers, storage, application and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. A cloud provider is a company which hosts the servers on its premises and makes the services available on-demand. 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.

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 [1].

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 into the Cloud. From users' perspective, including both individuals and IT enterprises, storing data remotely into the cloud in a flexible on-demand manner brings appealing benefits: relief of the burden for storage management, universal data access with independent geographical locations, and avoidance of capital expenditure on hardware, software, and personnel maintenances, etc[2].But even with this new data storage paradigm "Cloud" in the cloud computing bring challenging design issues, that influence on the performance and security of overall system. One of the biggest concerns with cloud data storage is that of data integrity verification at untrusted servers. Examples of outages and security breaches of noteworthy cloud services appear from time to time. In short, although outsourcing data to the cloud is economically attractive for long-term large-scale data storage, it does not immediately offer any guarantee on data integrity and availability [3]. This problem, if not properly attended, may obstruct the successful deployment of the cloud architecture. Simply downloading all the data for its integrity verification is not a practical solution due to the expensiveness in I/O and transmission cost across the network. Also, it is generally not possible 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. Besides, there may be more than one user accessing the same cloud storage. 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]. In all these works, great efforts are made to design solutions that meet various requirements: high scheme efficiency, stateless verification, unbounded use of queries and retrievability of data, etc. For this it is desirable that the cloud server only entertains verification request from a single designated party, thus enabling easy management of cloud server. Considering the role of the verifier in the model, all the schemes presented before fall into two categories: private auditability and public auditability.

Where private auditability can achieve higher pattern efficiency, public auditability allows anyone, not just the client (data owner), to challenge the cloud server for correctness of data storage while keeping no private information. Then, clients are able to delegate the evaluation of the service performance to an independent third party auditor (TPA), without devotion of their computation resources [1]. There may be clients in cloud that are capricious or may not afford the continuous integrity checks overhead. Practically, it seems more rational to equip the verification protocol with public auditability, which is expected to play a more important role in achieving economies of scale for Cloud Computing. Moreover,

Manuscript Received on May, 2014.

Rohini Kadlag, Department of Information Technology, Amrutvahini College of Engineering, Sangamner, India.

Rohit Devikar, Department of Information Technology, Amrutvahini College of Engineering, Sangamner, India.

for efficiency consideration, the outsourced data themselves should not be required by the verifier for the verification purpose. Other vital concern is pervious design is dynamic data operation for application of data storage in cloud, In cloud computing, the remotely stored electronic data might not only be accessed but also updated by the clients, e.g., through block modification, deletion, insertion, etc. .Considering public auditability and data dynamics for cloud data storage, we propose an efficient construction for the seamless integration of these two components in the protocol design. How to enable a privacy-preserving third-party auditing protocol, independent to data encryption, is the problem we are going to tackle in this paper. Our work utilizes the technique of public key based homomorphic authenticator [7],[9],[11], which enables TPA to perform the auditing without demanding the local copy of data reducing the computational overhead and interaction as in straightforward data auditing schemes .By integrating the homomorphic authenticator with random mask technique, our protocol guarantees that TPA could not learn any knowledge about the data content stored in the cloud server during the efficient auditing process. Our contribution can be summarized as follows:

1. We motivate the public auditing system of data storage security in Cloud Computing, and propose a protocol supporting for fully dynamic data operations, especially to support block insertion, which is missing in most existing schemes.
2. We extend our scheme to support scalable and efficient public auditing in Cloud Computing. In particular, our scheme achieves batch auditing where multiple delegated auditing tasks from different users can be performed simultaneously by the TPA.
3. We prove the security of our proposed construction and justify the performance of our scheme through concrete implementation and comparisons with the state of the art

Thus, following these concepts we are going to attend the automatic protocol blocker secures cloud environment from unauthorized access by the external user for privacy preserving in cloud computing [2].

II. RELATED WORK

Recently, much of growing interest has been pursued in the context of remotely stored data verification [2], [3], [4], [5],[6], [7], [8], [9], [10], [12], [13], [14], [15]. Ateniese et al. [7] are the first to consider public auditability in their defined "Provable Data Possession" (PDP) model for ensuring possession of data files on untrusted storages. Their scheme utilizes the RSA-based homomorphic authenticators for auditing outsourced data and suggests randomly sampling a few blocks of the file. However, the public auditability in their scheme demands the linear combination of sampled blocks exposed to external auditor. Ateniese et al.propose a dynamic version of the prior PDP scheme. However, the system imposes a priori bound on the number of queries and does not support fully dynamic data operations, i.e., it only allows very basic block operations with limited functionality, and block insertions cannot be supported. A straightforward Merkle-tree construction for public PoRs, this approach only works with encrypted data. Shacham et al. [11], design an improved PoR scheme built from BLS signatures with full proofs of security in the security model defined in [12]. Similar to the construction in [7], they use publicly verifiable homomorphic authenticators that are built from provably secure BLS signatures. Based on the elegant BLS construction, public retrievability is achieved. Again, their approach does not support privacy-preserving auditing for the same reason as [7]. Although the existing schemes aim at providing integrity verification for different data storage systems, the problem of supporting both public auditability and data dynamics has not been fully addressed. How to achieve a secure and efficient design to seamlessly integrate these two important components for data storage service remains an open challenging task in Cloud Computing. Portions of the work presented in this paper have previously appeared as an extended abstract [1]. We revise the paper a lot and add more technical details as compared to[1]. First, before the introduction of our proposed construction, we present two basic solutions (i.e., the MAC-based and signature-based schemes) for realizing data auditability and discuss their demerits in supporting public auditability and data dynamics. Second, we generalize the support of data dynamics to both PoR and PDP models and discuss the impact of dynamic data operations on the overall system efficiency both. In particular, we emphasize that while dynamic data updates can be performed efficiently in PDP models more efficient protocols need to be designed for the update of the encoded files in PoR models. For completeness, the designs for distributed data storage security are also discussed. Third, we extend our data auditing scheme for the single client and explicitly include a concrete description of the multiclient. We believe the analysis of these basic schemes will lead us to our main result, which overcomes all these drawbacks. Basic Scheme I and Basic Scheme II [a].The Privacy-Preserving Public Auditing Scheme[a], Batch Auditing[a], Data Dynamics [a]. The Privacy-Preserving Public Auditing Scheme[a], Batch Auditing[a], Data Dynamics [a]. Our proposed system enable privacy-preserving public auditing for cloud data storage under the aforementioned model, our protocol design should achieve the following security and performance guarantee:

1. **Public Auditability** to allow TPA to verify the correctness of the cloud data on demand without retrieving a copy of the. Whole data or introducing additional on-line burden to the cloud users.
2. **Storage Correctness** to ensure that there exists no cheating cloud server that can pass the audit from TPA without indeed storing users' data intact.
3. **Privacy-Preserving** to ensure that there exists no way for TPA to derive users' data content from the information collected during the auditing process;
4. **Batch Auditing** to enable TPA with secure and efficient auditing capability to cope with multiple auditing delegations from possibly large number of different users simultaneously.
5. **Lightweight** to allow TPA to perform auditing with minimum communication and computation overhead figures and tables.

III. PROBLEM STATEMENT

A. The System and Threat Model

Representative network architecture for cloud data storage is illustrated in Fig. 1. Three different network entities can be identified as follows:

1. **Client:** An entity, which has large data files to be stored in the cloud and relies on the cloud for data maintenance and computation, can be either individual consumers or organizations.
2. **Cloud Storage Server (CSS):** An entity, which is managed by Cloud Service Provider (CSP), has significant storage space and computation resource to maintain the clients' data.
3. **Third Party Auditor:** An entity, which has expertise and capabilities that clients do not have, is trusted to assess and expose risk of cloud storage services on behalf of the clients upon request.

In the cloud paradigm, by putting the large data files on the remote 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 [3], [5].

To authorize the CS to respond to the audit delegated to TPA's, the user can sign a certificate granting audit rights to the TPA's public key, and all audits from the TPA are authenticated against such a certificate. These authentication handshakes are omitted in the following presentation.

B. Design Goal

For papers accepted for publication, it is essential that the electronic version of the manuscript and artwork match the hardcopy exactly! The quality and accuracy of the content of the electronic material submitted is crucial since the content is not recreated, but rather converted into the final published version. Besides the deployment goal discuss above the system also summarize to attain the following goal:

1. **Dynamic data operation support:** To allow the clients to perform block-level operations on the data files while maintaining the same level of data correctness assurance. The design should be as efficient as possible so as to ensure the seamless integration of public auditability and dynamic data operation support [3].
2. **Blockless verification:** No challenged file blocks should be retrieved by the verifier (e.g., TPA) during verification process for efficiency concern [2].

IV. PROPOSED SCHEMES

This section states our public auditing scheme which provides a complete outsourcing solution of only the data itself, but also its integrity checking. We show how to extent our main scheme to support batch auditing for the TPA upon delegations from multiple users. Finally, we discuss how to generalize our privacy-preserving public auditing scheme and its support of data dynamics.

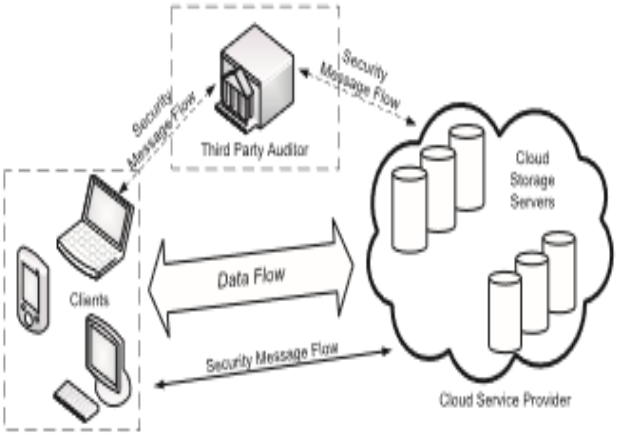

Fig 1: Architecture of cloud storage structure.

A. Definitions and Framework

We follow a similar definition of previously proposed schemes in the context of remote data integrity checking [8], [11], [13] and adapt the framework for our privacy-preserving public auditing system. A public auditing scheme consists of four algorithms (KeyGen, SigGen, GenProof, VerifyProof). KeyGen is a key generation algorithm that is run by the user to setup the scheme. SigGen is used by the user to generate verification metadata, which may consist of MAC, signatures, or other related information that will be used for auditing. GenProof is run by the cloud server to generate a proof of data storage correctness, while VerifyProof is run by the TPA to audit the proof from the cloud server. There are two steps involved in the system evaluation of batch auditing, Setup, Audit and Pblocker. Fig 2 shows the block diagram for proposed system.

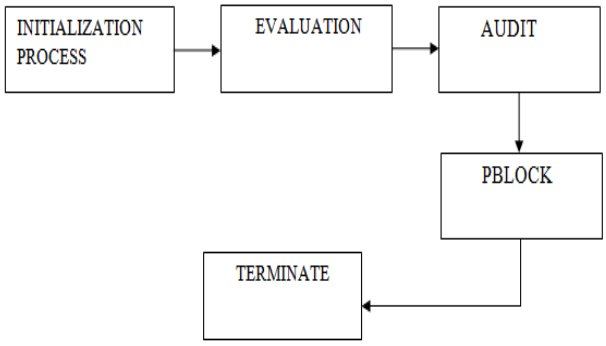

Fig 2: Block diagram for the proposed system.

Table I: Protocols for Default Integrity Verification

TPA		Cloud Server
1. Retrieve file tag t, verify its signature, and quit if fail;		
2. Generate a random challenge $chal = \{(i, v_i)\}_{i\in I}$;	$\xrightarrow[\text{challenge request } chal]{\{(i,v_i)\}_{i\in I}}$	3. Compute $\mu' = \sum_{i\in I} v_i m_{i,t}$ and also $\sigma = \prod_{i\in I} \sigma_i^{v_i}$; 4. Randomly pick $r \leftarrow \mathbb{Z}_p$, and compute $R = e(u,v)^r$ and $\gamma = h(R)$;
6. Compute $\gamma = h(R)$, and then verify $\{\mu, \sigma, R\}$ via Equation 1.	$\xleftarrow[\text{storage correctness proof}]{\{\mu,\sigma,R\}}$	5. Compute $\mu = r + \gamma\mu' \mod p$;

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

B. Basic Schemes

- **MAC-based Solution:**

There are two possible ways to make use of MAC to authenticate the data. A trivial way is just uploading the data blocks with their MACs to the server, and sends the corresponding secret key sk to the TPA. Later, the TPA can randomly retrieve blocks with their MACs and check the correctness via sk. Apart from the high (linear in the sampled data size) communication and computation complexities, the TPA requires the knowledge of the data blocks for verification. However, it suffers from the following severe drawbacks: 1) the number of times a particular data file can be audited is limited by the number of secret keys that must be fixed a priori. Once all possible secret keys are exhausted, the user then has to retrieve data in full to re-compute and re-publish new MACs to TPA; 2) The TPA also has to maintain and update state between audits, i.e., keep track on the revealed MAC keys. Considering the potentially large number of audit delegations from multiple users, maintaining such states for TPA can be difficult and error prone; 3) it can only support static data, and cannot efficiently deal with dynamic data at all. However, supporting data dynamics is also of critical importance for cloud storage systems [8].

- **HLA-based Solution:**

To effectively support public auditability without having to retrieve the data blocks themselves, the HLA technique [8], [10], [13] can be used. The difference is that HLAs can be aggregated. It is possible to compute an aggregated. HLA which authenticates a linear combination of the individual data blocks. Though allowing efficient data auditing and consuming only constant bandwidth, the direct adoption of these HLA-based techniques is still not suitable for our purposes. This is because the linear combination of blocks.

C. Audit Protocol Blocker

The proposed system include the existing system advantages as well as extends to find the unauthorized user ,to prevent the unauthorized data access for preserving data integrity. The proposed system keep a check on the user requests according the user specified parameters and also the parameters for the new and existing users .The system accepts request of only the existing validated user, and prompts for the new users for the parameter to match.

requirement specified during user creation for new users. If the new user prompt parameter matches with cloud server, it gives privileges to access the Audit protocol authorize the system automatically blocks the Audit protocol for specific user.

D. The Privacy-Preserving Public Auditing Scheme

In our proposed system we introduce to uniquely integrate the homomorphic authenticator with random mask technique. In our protocol, the linear combination of sampled blocks in the server's response is masked with randomness generated by a Pseudo Random Function (PRF). With random mask, the TPA no longer has all the necessary information to build up a correct group of linear equations and therefore cannot derive the user's data content, no matter how many linear combinations of the same set of file blocks can be collected. Due to the algebraic property of the homomorphic authenticator, the correctness validation of the block-authenticator pairs will not be affected by the randomness generated from a PRF. The system is based on BLS signature [15] that is used with public auditability which will benefit us for multi-task auditing. Scheme Details Let G1, G2 and GT be multiplicative cyclic groups of prime order p, and e: G1 × G2 → GT be a bilinear map as introduced in preliminaries. Let g be the generator of G2. H(·) is a secure map-to-point hash function: {0, 1} · → G1, which maps strings uniformly to G1.Another hash function h(·) : GT → Zp maps group element of GT uniformly to Zp. The proposed scheme is as follows: Setup Phase, Audit Phase[a].

Table II: Comparison between different remote data integrity checking schemes.

Scheme / Metric	[2]	[4]	[12]*	[14]	Our Scheme
Data dynamics	**No**		**Yes**		
Public auditability	**Yes**	**Yes**	**No**	**No**	**Yes**
Sever comp. complexity	**O(1)**	**O(1)**	**O(1)**	**O(log n)**	**O(log n)**
Verify comp. complexity	**O(1)**	**O(1)**	**O(1)**	**O(log n)**	**O(log n)**
Comm. complexity	**O(1)**	**O(1)**	**O(1)**	**O(log n)**	**O(log n)**
Verifier storage complexity	**O(1)**	**O(1)**	**O(1)**	**O(1)**	**O(1)**

V. CONCLUSION

In this paper, we propose a privacy-preserving public auditing system for data storage security. We designed the simulation by considering the single user. In Cloud computing, where TPA can perform the storage auditing without demanding the local copy of data. We use the homomorphic authenticator and random mask technique to guarantee that TPA would not learn any knowledge about the data content stored on the cloud server during the efficient auditing process, which not only eliminates the burden of cloud user from the tedious and possibly expensive auditing task, but also alleviates the users' fear of their outsourced data leakage.

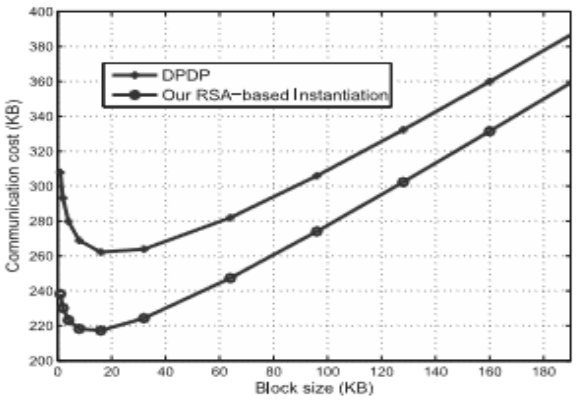

Fig 3: Block diagram for proposed system Comparison of communication complexity between our RSA- based instantiation and DPDP [14], for 1 GB file with variable block sizes. The detection probability is maintained to be 99 percent.

We further extend our privacy-preserving public auditing protocol into a multi-user setting, where TPA can perform the multiple auditing tasks in a batch manner, i.e., simultaneously. Many research issues have been highlighted and direction for future work have been suggested. Many open issues have been highlighted by the researchers in the protocol based privacy auditing in cloud environment. Security and performance of system states proposed schemes are provably secure and highly efficient. We believe all these advantages of the proposed schemes will shed light on economies of scale for Cloud Computing.

REFERENCES

1. Qian Wang, Student Member, IEEE, Cong Wang, Student Member, IEEE, Kui Ren, Member, IEEE Wenjing Lou, Senior Member, IEEE, and Jin Li "Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing" IEEE transactions on parallel and distributed systems, vol. 22, no.5, May 2011.
2. P. Mell, T. Grance (2009),"Draft NIST working definition of cloud computing", [Online]Available: http://www.csrc.nist. gov/groups/SNS/cloud-computing/index.html.
3. A. Juels and B.S. Kaliski Jr., "Pors: Proofs of Retrievability forLarge Files,"Proc. 14th ACM Conf. Computer and Comm. Security(CCS '07),pp. 584-597, 2007.
4. 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.
5. K.D. Bowers, A. Juels, and A. Oprea, "Proofs of Retrievability: Theory and Implementation," Report 2008/175, Cryptology ePrint Archive, 2008.
6. 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.
7. 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.
8. M.A. Shah, R. Swaminathan, and M. Baker, "Privacy-Preserving Audit and Extraction of Digital Contents," Report 2008/186,Cryptology ePrint Archive, 2008.
9. Oprea, M.K. Reiter, and K. Yang, "Space-Efficient Block Storage Integrity,"Proc. 12th Ann. Network and Distributed System Security Symp. (NDSS '05), 2005.
10. 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.
11. 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.
12. 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),2009.
13. 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), 2009.
14. G. Ateniese, R.D. Pietro, L.V. Mancini, and G. Tsudik, "Scalable and Efficient Provable Data Possession,"Proc. Fourth Int'l Conf. Security and Privacy in Comm. Networks (SecureComm '08),pp. 1-10,20080.
15. D. Boneh, C. Gentry, B. Lynn, H. Shacham,"Aggregate and verifiably encrypted signatures from bilinear maps", in Proc. of Eurocrypt 2003, Vol. 2656 of LNCS. Springer-Verlag, 2003, pp. 416–432