Dynamic Audit Services for Outsourced Storages in Clouds

Devi Parvathy Mohan, K.J.Jagdish

PG Student
Affiliated Anna University Chennai,Dept of computer science and enginnering
Gnanamani college of engineering
Namakkal .india
shalumohan4890@gmail.com

Asst Professor Dept of computer science and engineering
Gnanamani college of engineering
Namakkal .india
info@gce.org.in

Abstract: In cloud, the security issue in outsourced storage data is the difficult challenge. To overcome the problem, traditional approach developed a dynamic audit service for verifying the integrity of an untrusted and outsourced storage. An audit service is constructed based on the techniques, fragment structure, random sampling, and index-hash table, supporting provable updates to outsourced data and timely anomaly detection. The method based on probabilistic query and periodic verification for improving the performance of audit services. The audit service is performed by TPA monitoring. Sometimes the TPA may have chances to hide anomaly details to cloud users. To overcome the drawback, propose a dynamic audit service in the cloud. By the method it can dynamically audit the anomaly and send intimation to cloud user. So that it can secure the cloud storage data.

Keywords: about four key words separated by commas**.**

1. Introduction

The cloud storage service (CSS) relieves the burden for storage management and maintenance. However, if such an important service is vulnerable to attacks or failures, it would bring irretrievable losses to the clients because their data or archives are stored in an uncertain storage pool outside the enterprises. These security risks come from the following reasons: First, the cloud infrastructures are much more powerful and reliable than personal computing devices, but they are still susceptible to internal threats (e.g., via virtual machine) and external threats (e.g., via system holes) that can damage data integrity; second, for the benefits of possession, there exist various motivations for cloud service providers (CSP) to behave unfaithfully toward the cloud users; furthermore, disputes occasionally suffer from the lack of trust on CSP because the data change may not be timely known by the cloud users, even if these disputes may result from the users' own improper operations. Therefore, it is necessary for CSP to offer an efficient audit service to check the integrity and availability of stored data.

Existing System

. Existing work introduced a dynamic audit service for integrity verification of untrusted and outsourced storages. Constructed on interactive

proof system (IPS) with the zero knowledge property, our audit service can provide public auditability without downloading raw data and protect privacy of the data. Also, the audit system can support dynamic data operations and timely anomaly detection with the help of several effective techniques, such as fragment structure, random sampling, and index-hash table (IHT). It also developed an efficient approach based on probabilistic query and periodic verification for improving the performance of audit services. A proof-of-concept prototype is also implemented to evaluate the feasibility and viability of our approaches. The experimental results not only validate the effectiveness of our approaches, but also show that our system does not create any significant computation cost and require less extra storage for integrity verification. The method also has one drawback that is named as TPA monitoring.

Disadvantages

- It must requires external TPA monitoring
- No Secure

Proposed System

In existing work, the audit service is performed by TPA monitoring. Sometimes the TPA may have chances to hide anomaly details to cloud users. To overcome this drawback, we propose dynamic audit service in the cloud. In this method user sent query request to server and that server matches the user query and keyword if it is match, user can proceed the process otherwise, the user is automatically/dynamically marked as untrusted and sends intimation about anomaly detection to cloud user. So that it can secure the cloud storage data.

Advantages

- No need external TPA
- Secure & Effective

.**Modules:**

a. Authentication
b. Cloud Storage
c. Auditing
d. Secure Notification
e. Performance & Evaluation

Modules Description

Authentication:

Authorization is the process of giving user permission to do or have something. In multi-user computer systems, a system administrator defines for the system which users are allowed access to the system and what privileges of use (such as access to which file directories, time access, maintain history, and so forth). Assuming that someone has logged in to a computer operating system or application, the system or application may want to identify what resources the user can be given during this session. Thus, authorization is sometimes seen as both the preliminary setting up of permissions by a system administrator and the actual checking of the permission values that have been set up when a user is getting access. Authentication is the process of determining whether someone or something is, in fact, who or what it is declared to be. In private and public computer networks (including the Internet), authentication is commonly done through the use of logon passwords. Knowledge of the password is assumed to guarantee that the user is authentic. Each user registers initially (or is registered by someone else), using an assigned or self-declared password. On each subsequent use, the user must know and use the previously declared password. The weakness in this system for transactions that are significant (such

as the exchange of money) is that passwords can often be stolen, accidentally revealed, or forgotten.

Cloud Storage:

To utilize the public key based homomorphic authenticator and uniquely integrate it with random mask technique; we can upload and download and view our data into and from the cloud for privacy, to achieve a public auditing system for cloud data storage security while keeping all above requirements in mind.

Auditing:

With the establishment of auditing in Cloud Computing, TPA may concurrently handle multiple auditing delegations upon different users' requests. The individual auditing of these tasks for TPA can be tedious and very inefficient. Auditing not only allows TPA to perform the multiple auditing tasks simultaneously, but also greatly reduces the computation cost on the TPA side.

Secure Notification

Detection and notification refers to automatic detection of changes made to User pages and notification to interested users by Cloud Server or other means. Whereas search engines are designed to find User pages, detection and notification systems are designed to monitor changes to User pages. Efficient and effective change detection and notification is hampered by the fact that most servers do accurately track content changes through Modified.

Performance and Evaluation

To detect anomalies in a low-overhead and timely manner, we attempt to optimize the audit performance from two aspects: Performance evaluation of probabilistic queries and scheduling of periodic verification. Our basic idea is to maintain a tradeoff between overhead and accuracy, which helps us improve the performance of audit systems.

Literature Survey

1. **Privacy-Preserving Public Auditing for Secure Cloud Storage**

Using Cloud Storage, users can remotely store their data and enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources, without the burden of local data storage and maintenance. However, the fact that users no longer have physical possession of the outsourced data makes the data integrity protection in Cloud Computing a formidable task, especially for users with constrained computing resources. Moreover, users should be able to just use the cloud storage as if it is local, without worrying about the need to verify its integrity. Thus, enabling public auditability for cloud storage is of critical importance so that users can resort to a third party auditor (TPA) to check the integrity of outsourced data and be worry-free. To securely introduce an effective TPA, the auditing process should bring in no new vulnerabilities towards user data privacy, and introduce no additional online burden to user. It propose a secure cloud storage system supporting privacy-preserving public auditing. We further extend result to enable the TPA to perform audits for multiple users simultaneously and efficiently. Extensive security and performance analysis show the proposed schemes are provably secure and highly efficient. To address these problems, work utilizes the technique of public key based homomorphic linear authenticator (or HLA for short), which enables TPA to perform the auditing without demanding the

local copy of data and thus drastically reduces the communication and computation overhead as compared to the straightforward data auditing approaches. By integrating the HLA with random masking, our protocol guarantees that the TPA could not learn any knowledge about the data content stored in the cloud server during the efficient auditing process. The aggregation and algebraic properties of the authenticator further benefit our design for the batch auditing. Specifically, our contribution can be summarized as the following three aspects:

1) It motivate the public auditing system of data storage security in Cloud Computing and provide a privacy-preserving auditing protocol, i.e., our scheme enables an external auditor to audit user's outsourced data in the cloud without learning the data content.

2) To the best of our knowledge, the scheme is the first to support scalable and efficient public auditing in the Cloud Computing. Specifically, our scheme achieves batch auditing where multiple delegated auditing tasks from different users can be performed simultaneously by the TPA.

3) It prove the security and justify the performance of our proposed schemes through concrete experiments and comparisons with the state-of-the-art.

2. BAF: An Efficient Publicly Verifiable Secure Audit Logging Scheme for Distributed Systems

Audit logs, providing information about the current and past states of systems, are one of the most important parts of modern computer systems. Providing security for audit logs on an untrusted machine in a large distributed system is a challenging task, especially in the presence of active adversaries. In such a system, it is critical to have forward security such that when an adversary compromises a machine, she cannot modify or forge the log entries accumulated before the compromise. Unfortunately, existing secure audit logging schemes have significant limitations that make them impractical for real-life applications: Existing Public Key Cryptography (PKC) based schemes are computationally expensive for logging in task intensive or resource-constrained systems, while existing symmetric schemes are not publicly verifiable and incur significant storage and communication overheads. In this paper, we propose a novel forward secure and aggregate logging scheme called Blind-Aggregate-Forward (BAF) logging scheme, which is suitable for large distributed systems. BAF can produce publicly verifiable forward secure and aggregate signatures with near-zero computational, storage, and communication costs for the loggers, without requiring any online Trusted Third Party (TTP) support. We prove that BAF is secure under appropriate computational assumptions, and demonstrate that BAF is significantly more efficient and scalable than the previous schemes. Therefore, BAF is an ideal solution for secure logging in both task intensive and resource-constrained systems.

To address the above problems, a set of cryptographic countermeasures have been proposed to enable secure logging on untrusted machines, without assuming a tamper-resistant hardware or continuous real-time log verifiers In order to fulfill this requirement, we propose a novel forward secure and aggregate logging scheme for secure audit logging in distributed systems, which we call BlindAggregate-Forward (BAF) logging scheme.

BAF can address all the aforementioned limitations of the existing approaches simultaneously.

3. Dynamic Provable Data Possession

. As storage-outsourcing services and resource-sharing networks have become popular, the problem of efficiently proving the integrity of data stored at untrusted servers has received increased attention. In the provable data possession (PDP) model, the client preprocesses the data and then sends it to an untrusted server for storage, while keeping a small amount of meta-data. The client later asks the server to prove that the stored data has not been tampered with or deleted (without downloading the actual data). However, the original PDP scheme applies only to static (or append-only) files. We present a definitional framework and efficient constructions for dynamic provable data possession (DPDP), which extends the PDP model to support provable updates to stored data. We use a new version of authenticated dictionaries based on rank information. The price of dynamic updates is a performance change from O(1) to O(logn) (or $O(n^{\epsilon}logn)$), for a file consisting of n blocks, while maintaining the same (or better, respectively) probability of misbehavior detection. Our experiments show that this slowdown is very low in practice (e.g., 415KB proof size and 30ms computational overhead for a 1GB file). We also show how to apply our DPDP scheme to outsourced file systems and version control systems (e.g., CVS). In this paper, we provide a definitional framework and efficient constructions for dynamic provable data possession (DPDP), which extends the PDP model to support provable updates on the stored data. Given a file F consisting of n blocks, we define an update as either insertion of a new block (anywhere in the file, not only append), or modification of an existing block, or deletion of any block. Therefore our update operation describes the most general form of modifications a client may wish to perform on a file. Our DPDP solution is based on a new variant of authenticated dictionaries, where we use rank information to organize dictionary entries. Thus we are able to support efficient authenticated operations on files at the block level, such as authenticated insert and delete. It prove the security of our constructions using standard assumptions.

4. Scalable and Efficient Provable Data Possession

Storage outsourcing is a rising trend which prompts a number of interesting security issues, many of which have been extensively investigated in the past. However, Provable Data Possession (PDP) is a topic that has only recently appeared in the research literature. The main issue is how to frequently, efficiently and securely verify that a storage server is faithfully storing its client's (potentially very large) outsourced data. The storage server is assumed to be untrusted in terms of both security and reliability. (In other words, it might maliciously or accidentally erase hosted data; it might also relegate it to slow or off-line storage.) The problem is exacerbated by the client being a small computing device with limited resources. Prior work has addressed this problem using either public key cryptography or requiring the client to outsource its data in encrypted form. In this paper, we construct a highly efficient and provably secure PDP technique based entirely on symmetric key cryptography, while not requiring any bulk encryption. Also, in contrast with its predecessors, our PDP technique allows outsourcing of dynamic

data, i.e, it efficiently supports operations, such as block modification, deletion and append.

we construct a highly efficient and provably secure PDP technique based entirely on symmetric key cryptography, while not requiring any bulk encryption. Also, in contrast with its predecessors, our PDP technique allows outsourcing of dynamic data, i.e, it efficiently supports operations, such as block modification, deletion and append

5. Cooperative Provable Data Possession for Integrity Verification in Multi-Cloud Storage

Provable data possession (PDP) is a technique for ensuring the integrity of data in storage outsourcing. In this paper, we address the construction of an efficient PDP scheme for distributed cloud storage to support the scalability of service and data migration, in which we consider the existence of multiple cloud service providers to cooperatively store and maintain the clients' data. We present a *cooperative* PDP (CPDP) scheme based on homomorphic verifiable response and hash index hierarchy. We prove the security of our scheme based on multi-prover zero-knowledge proof system, which can satisfy completeness, knowledge soundness, and zero-knowledge properties. In addition, we articulate performance optimization mechanisms for our scheme, and in particular present an efficient method for selecting optimal parameter values to minimize the computation costs of clients and storage service providers. Our experiments show that our solution introduces lower computation and communication overheads in comparison with non-cooperative approaches To check the availability and integrity of outsourced data in cloud storages, researchers have proposed two basic approaches called Provable Data Possession and Proofs of Retrievability .Ateniese et al. first proposed the PDP model for ensuring possession of files on untrusted storages and provided an RSA-based scheme for a static case that achieves the communication cost. They also proposed a publicly verifiable version, which allows anyone, not just the owner, to challenge the server for data possession..They proposed a lightweight PDP scheme based on cryptographic hash function and symmetric key encryption, but the servers can deceive the owners by using previous metadata or responses due to the lack of randomness in the challenges. The numbers of updates and challenges are limited and fixed in advance and users cannot perform block insertions anywhere.

6. Efficient Audit Service Outsourcing For Data Integrity In Clouds

Cloud-based outsourced storage relieves the client's burden for storage management and maintenance by providing a comparably low-cost, scalable, location-independent platform. However, the fact that clients no longer have physical possession of data indicates that they are facing a potentially formidable risk for missing or corrupted data. To avoid the security risks, audit services are critical to ensure the integrity and availability of outsourced data and to achieve digital forensics and credibility on cloud computing.

Provable data possession (PDP), which is a cryptographic technique for verifying the integrity of data without retrieving it at an untrusted server, can be used to realize audit services. In this paper, profiting from the interactive zero-knowledge proof system, we address the construction of an interactive PDP protocol to prevent the fraudulence of prover

(soundness property) and the leakage of verified data (zero-knowledge property). We prove that our construction holds these properties based on the computation Diffie–Hellman assumption and the rewind able black-box knowledge extractor. We also propose an efficient mechanism with respect to probabilistic queries and periodic verification to reduce the audit costs per verification and implement abnormal detection timely. In addition, we present an efficient method for selecting an optimal parameter value to minimize computational overheads of cloud audit services. Our experimental results demonstrate the effectiveness of our approach.

7. Identity-Based Encryption from the Weil Pairing

It propose a fully functional identity-based encryption scheme (IBE). The scheme has chosen cipher text security in the random oracle model assuming an elliptic curve variant of the computational Difi Hellman problem. Our system is based on the Weil pairing. We give precise definitions for secure identity based encryption schemes and give several applications for such systemsShamir's original motivation for identity-based encryption was to simplify certificate management in e-mail systems. When Alice sends mail to Bob at bob@hotmail.com she simply encrypts her message using the public key string \bob@hotmail.com". There is no need for Alice to obtain Bob's public key certificate. When Bob receives the encrypted mail he contacts a third party, which we call the Private Key Generator (PKG). Bob authenticates himself to the PKG in the same way he would authenticate himself to a CA and obtains his private key from the PKG. Bob can then read his e-mailNote that unlike the existing secure e-mail infrastructure, Alice can send encrypted mail to Bob even if Bob has not yet setup his public key certificateIn this paper we propose a fully functional identity-based encryption scheme. The performance of our system is comparable to the performance of ElGamal encryption in F . The security of our system is based on a natural analogue of the computational Difie-Hellman assumption on elliptic curves. Based on this assumption we show that the new system has chosen ciphertext security in the random oracle model. Using standard techniques from threshold cryptography the PKG in our scheme can be distributed so that the master-key is never available in a single location. Unlike common threshold systems, we show that robustness for our distributed PKG is free.

8. Tight Proofs for Signature Schemes without Random Oracles

It present the first tight security proofs for two general classes of Strong RSA based signature schemes. Among the affected signature schemes are the Cramer-Shoup, Camenisch-Lysyanskaya, Zhu, and Fischlin signature scheme. As the representation of elements in prime order bilinear groups is much smaller than in RSA groups, we also present two bilinear variants of our signature classes that output short signatures. Similar to before, we are able to show that the sevariants have tight security proof sunder the StrongDi Hellman (SDH) assumption. We so obtain very efficient SDH based variants of the Cramer-Shoup, Fischlin, and Zhu signature scheme and the first tight security proof for the recent Camenisch-Lysyanskaya scheme that was proposed and proven secure under the SDH assumption. Central to our results is a new proof

technique that allows the simulator to avoid guessing which of the attacker's signature queries will be re-used in the forgery. In contrast to previous proofs, our security reduction does not lose a factor of q here. In this work, we ask the following question: are there tight security proofs for the existing practical signature schemes However, our result is not limited to the original schemes. In our analysis, we generalize the schemes by Camenisch-Lysyanskaya, Fischlin and Zhu by introducing a new family of randomization functions, called combining functions. The result of this generalization is an abstract signature scheme termed 'combining scheme'. In a similar way, we introduce a second general class of signature schemes called 'chameleon hash scheme' that can be regarded as a generalization of the Cramer-Shoup signature scheme. Then, we prove the combining signature scheme and the chameleon hash scheme to be tightly secure under the SRSA assumption when instantiated with any secure combining function, respectively chameleon hash function.

9. Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing

. Cloud Computing has been envisioned as the next-generation architecture of IT Enterprise. It moves the application software and databases to the centralized large data centers, where the management of the data and services may not be fully trustworthy. This unique paradigm brings about many new security challenges, which have not been well understood. This work studies the problem of ensuring the integrity of data storage in Cloud Computing. 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. The introduction of TPA eliminates the involvement of client through the auditing of whether his data stored in the cloud is indeed intact, which can be important in achieving economies of scale for Cloud Computing. The support for data dynamics via the most general forms of data operation, such as block modification, insertion and deletion, is also a significant step toward practicality, since services in Cloud Computing are not limited to archive or backup data only. While prior works on ensuring remote data integrity often lacks the support of either public verifiability or dynamic data operations, this paper achieves both. We first identify the difficulties and potential security problems of direct extensions with fully dynamic data updates from prior works and then show how to construct an elegant verification scheme for seamless integration of these two salient features in our protocol design. In particular, to achieve efficient data dynamics, we improve the Proof of Retrievability model by manipulating the classic Merkle Hash Tree (MHT) construction for block tag authentication. Extensive security and performance analysis show that the proposed scheme is highly efficient and provably secure.

10. Ensuring Data Storage Security in Cloud Computing

Cloud computing has been envisioned as the next-generation architecture of IT enterprise. In contrast to traditional solutions, where the IT services are under proper physical, logical and personnel controls, cloud computing moves the application software and databases to the large data centers,

where the management of the data and services may not be fully trustworthy. This unique attribute, however, poses many new security challenges which have not been well understood. In this article, we focus on cloud data storage security, which has always been an important aspect of quality of service. To ensure the correctness of users' data in the cloud, we propose an effective and flexible distributed scheme with two salient features, opposing to its predecessors. By utilizing the homomorphic token with distributed verification of erasure-coded data, our scheme achieves the integration of storage correctness insurance and data error localization, i.e., the identification of misbehaving server (s). Unlike most prior works, the new scheme further supports secure and efficient dynamic operations on data blocks, including: data update, delete and append. Extensive security and performance analysis shows that the proposed scheme is highly efficient and resilient against Byzantine failure, malicious data modification attack, and even server colluding attacks.

References

[1] Amazon Web Services, "Amazon S3 Availability Event: July 20, 2008," http://status.aws.amazon.com/s3-20080720.html, July 2008.

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

[3] M. Mowbray, "The Fog over the Grimpen Mire: Cloud Computing and the Law," Technical Report HPL-2009-99, HP Lab., 2009.

[4] A.A. Yavuz and P. Ning, "BAF: An Efficient Publicly Verifiable Secure Audit Logging Scheme for Distributed Systems," Proc. Ann. Computer Security Applications Conf. (ACSAC), pp. 219-228, 2009.

[5] G. Ateniese, R.C. Burns, R. Curtmola, J. Herring, L. Kissner, Z.N.J. Peterson, and D.X. Song, "Provable Data Possession at Untrusted Stores," Proc. 14th ACM Conf. Computer and Comm. Security, pp. 598-609, 2007.

[6] 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. Netowrks (SecureComm), pp. 1-10, 2008.

[7] C.C. Erway, A. Ku¨ pc¸u¨ , C. Papamanthou, and R. Tamassia, "Dynamic Provable Data Possession," Proc. 16th ACM Conf. Computer and Comm. Security, pp. 213-222, 2009.

[8] 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 Advances in Cryptology (ASIACRYPT '08), J. Pieprzyk, ed., pp. 90-107, 2008.

[9] H.-C. Hsiao, Y.-H. Lin, A. Studer, C. Studer, K.-H. Wang, H. Kikuchi, A. Perrig, H.-M. Sun, and B.-Y. Yang, "A Study of User- Friendly Hash Comparison Schemes," Proc. Ann. Computer Security Applications Conf. (ACSAC), pp. 105-114, 2009.

[10] A.R. Yumerefendi and J.S. Chase, "Strong Accountability for Network Storage," Proc. Sixth USENIX Conf. File and Storage Technologies (FAST), pp. 77-92, 2007.