Efficient and Secure public auditing scheme for cloud storage

Kairamkonda Vani¹, Mrs. Lalitha Manglaram²

¹M. Tech Student, Dept of CSE, Kshatriya College of Engineering, Armoor, Nizamabad TS, India
²Assistant Professor, Dept of CSE, Kshatriya College of Engineering, Armoor, Nizamabad TS, India

Abstract
Cloud computing poses many challenges on integrity and privacy of users’ data though it brings an easy, cost-effective and reliable way of data management. Hence, secure and efficient methods are needed to ensure integrity and privacy of data stored at the cloud. Wang et al. proposed a privacy-preserving public auditing protocol in 2010 but it is seriously insecure. Their scheme is vulnerable to attacks from malicious cloud server and outside attackers regarding to storage correctness. So they proposed a scheme in 2011 with an improved security guarantee but it is not efficient. Thus, in this paper, we proposed a scheme which is secure and with better efficiency. It is a public auditing scheme with third party auditor (TPA), who performs data auditing on behalf of user(s). With detail security analysis, our scheme is proved secure in the random oracle model and our performance analysis shows the scheme is efficient.

Keywords: Auditing, Privacy, Cloud

1. Introduction
Continued growths of the internet and advances in technology have fueled a trend towards outsourcing data and its management. One of the emerging technology due to the advancement of internet is cloud computing. Cloud computing brings an easy, cost effective and reliable way for data owners to deal with their data storage and its computation. By outsourcing data to the cloud, organizations can concentrate on their core business issues and operate other business applications through the internet rather than incurring substantial hardware, software, and personnel costs involved in deploying and maintaining applications in-house.

The pervasive nature of the internet helps cloud service providers (CSPs) to reach a global audience and market. The cloud service provider is responsible for offering not only adequate hardware, software and network resources to host owners’ personal data but also mechanisms for them to efficiently create, update and access their outsourced data. It is clear to see that the CSP hosts the data owners’ intellectual property, handle transactions with the customer, and deal with various support issues. This implies data and computation are not in the place they used to be in which is a serious security concern for cloud users. Rong et al. [1] classified cloud computing security challenges as traditional and new security challenges to emphasize that security concerns in traditional communication systems also apply to the cloud. In addition, they explain how the use of cloud computing could help attackers to perform their attacks in a more sophisticated way. It is, therefore, vital to provide sufficient security measures for protecting the stored data both from malicious outsider attacks and the service provider itself, and this lack of trust is crucial as it brings new security issues towards the cloud environment. Security for the most part, in our case, implies maintaining data integrity and guaranteeing privacy of data.

For the issue of data integrity, an obvious mechanism is checking on retrieval which is impossible to assure integrity for the whole data because huge amount of data are outsourced to the cloud and only few are frequently accessed. Since most of the data are accessed less frequently (or not at all), this implies the integrity check of such data is bypassed by this technique. Another technique is to download the whole data and perform the integrity check which is impractical due to heavy I/O overhead on the cloud server and high communication cost for the owner. Recently, a number of schemes have been proposed to check the integrity of remote data without the knowledge of the entire data with different requirements. Among these, some employ integrity checks with random sampling which consequently bounded in use of queries and some are
integrity checks with no public verifiability and no dynamic operation support.

In addition, most of the aforementioned systems are not suitable for third party auditing and some schemes with intention of providing this feature are constructed for the cloud environment. These schemes basically include privacy preserving guarantee though some are insecure and others inefficient. Wang et al.'s scheme is proposed with privacy-preserving public auditing capability for cloud storage but it is seriously insecure. The root cause for the insecurity of this scheme is inappropriate definition and use of private/public parameters during signature generation and resistant to the stated attacks but is inefficient since it involves computationally intensive pairing operation for the sake of data blinding. Therefore, users' lacks of trust to the cloud, in general, and mentioned problems of Wang et al.'s schemes, in particular, are those factors motivated us to do this work. In this article, we try to give a solution to the problem of providing secure integrity mechanism together with data privacy assurance while incurring optimal computation and communication overhead. Our protocol uses homomorphism linear authenticators which are constructed for data integrity. It also uses random masking for data hiding. It is constructed based on the security model by making use of short signatures from BLS constructions. It is in the same security level but with better efficiency.

2. Related work

In the past few years, there have been some works which deal with integrity and privacy of outsourced data. There are works particularly dealt with both possession and retrieval of remotely stored data. It is the first to construct and formally define publicly verifiable Provable Data Possession (PDP) model, which is provably secure and introducing the concept of RSA-based homo-morphed verifiability tags to acquire block less verification. However, linear combination of file blocks is needed for verification with no data privacy guarantee. Moreover, they do not consider the case of dynamic data storage and only takes care of static data possession guarantee at entrusted servers. Introduce a proof of retrieval scheme with both possession and extractability capabilities. They use spot-checking approach by embedding special check blocks, sentinels, into the data file but lead to use of bounded queries and hinder dynamic data operation support. For data dynamics support, constructed a scheme based on symmetric key cryptography with no need of any bulk encryption. But this scheme also imposes a prior bound on the number of queries and does not provide a fully dynamic environment since it lacks data block insertion operation. Provide a well defined proof of retrieval model with full security proof. They used publicly verifiable homomorphism linear authenticators which are built from BLS signatures. These short signatures help to aggregate individual signatures and provide a very small authenticated value for public verifiability. The downside of this scheme is linear combination of data blocks required for verification and consequently the verifier might have the opportunity to derive the data from collected proof responses during the auditing process. In addition, this scheme does not support dynamic data operations and gives a formal framework for dynamic provable possession of data and provide the first fully dynamic solution to support provable updates to stored data files using rank-based authenticated skip lists.

While these advantages make the cloud a perfect infrastructure for large and small applications, the cloud brings new security risks toward outsourced data. The fact that the cloud provider is a third party storage system that does not have the authority to view or edit the data is a risk by itself. Also the interests of the data owner may collide with the interests of the cloud provider that may lead to harming the data itself. On the other hand, cloud providers may also hide some incidents from the customers to preserve their reputation and prevent customers from raising law suits against them. All the previous risks are internal risks, but the cloud also introduces new external risks such as the fact that all different kinds of data are collected in one place which will be an attractive target for hackers much more than local small storage systems. Public cloud auditing and data encryption can be used to protect the data against internal and external risks.

3. Proposed Model

Our public auditing scheme comprises three different entities (parties) with well defined interactions among them, as shown in Fig. 1. Cloud server (for brevity, referred to server from here on), which is owned by CSP, has the infrastructure and expertise to host outsourced storage, and provides efficient mechanisms for its users to create, store, update and request for retrieval. User (client), who has data to be stored on the cloud, leaves information technology (IT) operations on data to professionals and concentrate on his/her core
businesses. Third party auditor, another entity who has better expertise and capabilities than the user, is trusted to measure the cloud storage reliability and validity on behalf of users when needed.

Users can put huge data on the cloud to make themselves free from the burden of storage and maintenance. As we assume that the CSP is semi-trusted, which means it follows the normal flow of the protocol in the system. However, it might not be trusted with the actual data contents and its integrity, i.e. it could behave unfaithfully toward users, regarding to the status of their outsourced data for benefits of their own.

Therefore, for users’ data to be correctly stored and maintained, some integrity checking mechanism should be in place. Users can delegate a trusted third party auditor to perform security check tasks as it is not economically feasible for them to handle it by their own.

When we start to construct our scheme, we put the following fundamental goals: Public verifiability: to allow TPA to verify the correctness of cloud data on demand without retrieving the entire data or without introducing additional online burden to the cloud users. Storage correctness: to ensure that any server can pass TPA’s verification only if it keeps user’s data intact. Privacy preserving: to assure that no data content is leaked to TPA during the auditing process. Batch auditing: to enable TPA to cope with multiple auditing delegations from possibly many users concurrently in secure and efficient manner. Block less verification: no challenged file blocks should be retrieved by TPA during the auditing process both for efficiency and security reasons. Important requirements to achieve these goals are low computation and communication complexity, and low storage cost.

4. Implementation

Encrypting data is the first and essential line of defense against attackers and unauthorized users. Data will be encrypted while being stored in the cloud to prevent cloud providers from accessing or tampering the data. Also data will be encrypted while being transferred between entities in the system or while being stored in some temporary locations such as doctors local machines. The following section describes the various encryption schemes and their usage.

Symmetrical Encryption

Symmetric-key encryption schemes are schemes in which the encryption key and the decryption key are the same. One of the schemes that follow symmetric-key encryption is Advanced Encryption Standard (AES). In those schemes the user must share the key with all the users who have the authority to view the encrypted data and if the user needs to give different access permissions to different users then there is a need to encrypt each part of the data with different keys and share each key with the authorized users only. Therefore, in a group of users each two users must have unique key shared between them for secured communication. Symmetric-key encryption systems consist of multiple rounds of substitution-permutation networks, that is why it is fast and simple, but with a large key and a good number of rounds, the system becomes secure and hard to break. However, the system becomes too complicated and hard to manage when the number of users’ increases and with the need of fine-grained access policies because of the huge number of keys and the work needed by the users and the system to manage data access [8]. Symmetric-key encryption will be used in the system to encrypt emergency report while being stored in the cloud.

Public-Private Encryption

Public-Private encryption schemes are schemes that use two different keys for data encryption and decryption. The user first generates a private key that should be kept secured, and a public key which is released to the users. Any data that was encrypted by the public key can only be decrypted by the private key and vice versa. RSA is a famous public-private encryption scheme. So if Alice wants to share data with Bob, she just needs to know the public key for Bob and use it to encrypt the message. When Bob receives the encrypted message he will use his private key to decrypt the message and read
its contents. Therefore, in a group of users, each user must have one pair of keys, a private one that is hidden from the others, and a public key that is shared with all the users. The users in those schemes should only manage their private keys, while the public keys can be kept in a shared place.

**Cipher text-Policy Attribute-Based Encryption**

Cipher text-Policy Attribute-Based Encryption (CP-ABE) [9] is a public key encryption scheme that enables the user to specify who can access the data when encrypting it by constructing a fine-grained access policy and associate the policy with the data. Each user in the system must have a private key associated with a list of attributes that describes the user; the user can decrypt the data file if and only if her key’s attributes can satisfy the access policy that was associated with the encrypted data. CP-ABE consists of a master key, a public key, a set of attributes, a set of private keys, and four fundamental algorithms (Setup, Encrypt, Key Generation, Decrypt) in addition to one optional algorithm.

**5. Real time Example - Electronic Health Records**

“The electronic health record (EHR) is an evolving concept defined as a longitudinal collection of electronic health information about individual patients and populations. Primarily, it will be a mechanism for integrating health care information currently collected in both paper and electronic medical records (EMR) for the purpose of improving quality of care”. In addition of the role that EHR systems play in health data sharing and simplifying the process of accessing and searching medical records, EHR systems also help:

- Improving quality and convenience of patient care.
- Increasing patient participation in their care.
- Improving quality and convenience of patient care.
- Improving care coordination.
- Increasing practice efficiencies and cost savings.
- Improving clinical and health services research and clinical education.
- Eliminate the physical storage requirements.
- Providing the potential to deliver a longitudinal record that can cover a long track of medications and history for patients and provide comprehensive data across populations.

The ease of accessing and sharing medical data between providers through EHR systems made it also easier for unauthorized users to access private data that was unavailable or very hard for them to access before EHR systems (internal threats). The amount of private and confidential data that is managed by EHR systems made them an active target for attackers and hackers to gain access for sharing, selling, or tampering medical data (external threats). Therefore, EHR systems in addition to traditional needs to preserve patients’ privacy, confidentiality, and security, must be able to protect the data and defend against internal and external threats.

**6. Data Auditing**

Since the cloud providers are not the real owners of the data and they are not authorized to edit, view, or delete the data, many systems were introduced to prove the ownership of the data such as, or to verify that the data was not tampered with or deleted by proving its integrity, such as. However, those systems rely on third party auditors to store testing data and keys and to do the verification for them. Using third party auditors may force the users to reveal some private data to the auditors to be able to do their job, which violates the privacy and confidentiality requirements for the system. On the other hand, systems such as were designed to preserve data privacy by making the third party auditors do their job without the need to access any confidential data. Meanwhile, the fact that this system depends on a third party auditor, which is assumed to be trusted to store the auditing data and keys and to do auditing as required from them, is a weakness in the system.

Using a single design to achieve a fully secured system that preserves privacy and confidentiality with the ability to provide fast and easy access to emergency data, is almost impossible without compromising some of the security features in EHR or without providing an easy way to override these security measures. Also even though encrypting data while being at rest or transit is essential to protect the data and preserve privacy and confidentiality, encryption is not enough by itself and extra measures should be taken to secure data [11]. The first part of our hypothesis is that EHR.

The proposed framework consists of three entities: a cloud provider (CP), a private secured webserver (PSW), and healthcare providers’ private machines (HPM). Patient data is divided into three main categories: medical data, personal data, and insurance data. Each category has many subcategories for achieving fine-grained access control. Data are stored in
a structured XML format (as shown in Figure 2). This XML format makes the proposed framework simple to integrate with current EHR systems. Any current system needs only to follow the formatted XML structure to be able to read from or write to our proposed system, which makes it platform independent. Medical records are encrypted using CP-ABE by the patients or healthcare providers before storing them in the cloud. Patients access and manage their medical data via a web application, while healthcare providers can either use the web application to view medical data and emergency reports, or use a desktop version for full capabilities over accessible medical data.

Also, a unique identification value is given to all users to be able to access the web version of the application, the identification values are also associated with the private keys as an attribute. Patients can request access to the system only through the PSW sever. Private keys and identification values will be created for patients to access the system and decrypt their private data. Patient’s control that can access their medical data by editing access policies. This can be done by requesting an update process in which PSW will re-encrypt the data with the new access policy, and then store it in the cloud.

7. Conclusion
Data outsourced to a potentially un-trusted party (CSP in our case) needs continuous checking of its security. By security we mean the integrity and privacy of data stored remotely. If no periodical data validity checking mechanism is set, the cloud service provider could do any modifications or storage arrangements over users’ data. The way we made the integrity verification also requires great attention in order not data leaked to an unauthorized party. In a public auditing scheme users are privileged to delegate third party auditor for their data validity checking. However, this delegation brings privacy concerns since the third party auditor can have the potential to derive data blocks from a number of responses he collected from server to his request made for verification. Thus our proposed scheme will alleviate the problem of data integrity by keeping its privacy using blinding technique. It is a public auditing scheme and for better efficiency, our scheme provides a privilege for TPA to handle auditing from multiple users with different messages concurrently at the same time. We improve the efficiency of our scheme by minimizing computationally intensive operations like bilinear mapping and by avoiding all in all such computations from user side. Through a detail security and performance analysis the scheme is proved to be secure and efficient. Even though our scheme is not fully dynamic, data operations such as deletion, modification and appending can be supported. Our scheme is not fully dynamic because data block insertion makes it too inefficient. Constructing both fully dynamic and secure public auditing scheme for cloud environment is an open problem, and we left it as our future research work.

References

Author Profile
Kairamkonda Vani Completed her B.Tech (IT). She is pursuing M-Tech(Software engineering), CSE, Kshatriya College of Engineering, Armoor, Nizamabad, TS, India.

Author Profile
Mrs. Lalitha Manglaram well has known author and excellent teacher. Graduated in M.TECH (C.S.E) Kshatriya College of Engineering and B. TECH (C.S.E) Vijay Rural Engineering College from JNTUH UNIVERSITY. She is working as Assistant Professor in C.S.E Dept in Kshatriya College of Engineering. She has vast teaching experience of 10 years in various colleges. Interest Areas are data mining and data warehouse, programming languages, computer organization, mobile computing and network security.