Improving Cloud Storage Security using Public-Key Based Homomorphic linear authenticator (HLA) and Threshold Cryptography

Reshma Suryawanshi¹, S.N.Shelke²
¹M.E Student, ²Assistant Professor, Computer Engineering, Sinhgad Academy of Engineering, Pune, India

Abstract: The use of cloud computing in many organizations is increasing rapidly. Cloud computing has gained its popularity due to its advantages like cost reduction, greater flexibility, elasticity, and optimal resource utilization. Besides all the benefits of the cloud computing security of the stored data needs to be considered while storing sensitive data on cloud. Cloud users cannot rely only on cloud service provider in concern of security of their sensitive data. For that purpose in this paper we are proposing two schemes in first scheme we are proposing a public auditor which will check security of sensitive data stored on cloud servers. That is we are providing a technique of Third Party Auditor (TPA) to check integrity of data stored on cloud for cloud owner. Using TPA, the public auditing process does not bring new vulnerabilities towards user data privacy, and does not impose an additional online burden to user. In second scheme we are proposing a threshold cryptography technique in which data owner divides users in groups and gives single key to each user group for decryption of data, and each user in the group shares parts of the key so as to achieve high data confidentiality.

Keywords: Cloud Computing, Cloud Storage, Data Integrity, Third party auditor, threshold cryptography, key

I. INTRODUCTION

Over the last few years clouds have become the buzzword in computing. Cloud services are commercial and offered as pay per use model. Cloud-based outsourced storage relieves the client’s burden for storage management and maintenance by providing a comparably low-cost, scalable, location-independent platform. If such an important service is vulnerable to attacks or failures, it would bring irretrievable losses to users since their data or archives are stored into an uncertain storage pool outside the enterprises [1]. The sensitive data on cloud can be compromised by various means such as cloud service provider (CSP), user’s own improper operations, unauthorized user etc. Examples of outages and security reaches of noteworthy cloud services appear from time to time. Cloud data is susceptible to security threats both from outside and inside the cloud [5]. For their benefits, the cloud service providers (CSP) may behave unfaithfully toward the cloud users. Cloud computing security risk incident has happened when Google a major cloud computing and Software as a Service (SaaS) provider had its systems attacked and hacked [3]. One of the top security concerns of enterprises are the physical location of the data that are being stored in the cloud especially if they are located in another country because the laws of the host country of the equipment apply to the data on the machines and that could be a big issue if the host country does not have adequate laws to protect sensitive data or if the host nation becomes hostile or when the government of the hosting nation changes and become unfriendly [3]. Even Amazon’s S3, the best-known storage service, has experienced significant downtime.

Data owner even can't trust on users as they may be malicious. Data confidentiality may violet through collusion attack of malicious users and service providers [16]. Studies of deployed large-scale storage systems show that no storage service can be completely reliable; all have the potential to lose or corrupt customer data [1, 3, 4]. To know if data owner’s data is safe, he must either blindly trust the service provider or laboriously retrieve the hosted data every time he wants to verify its integrity, neither of which is satisfactory. The outsourced data can be large in size and the user may have limited resource capability so that the tasks of auditing the data integrity 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) [5]. Unfortunately, to date, there are no fair and explicit mechanisms for making these services accountable for data loss [4]. Therefore, it is crucial to realize public auditability for cloud storage service along with threshold cryptography, so that data owners may resort to a third party auditor (TPA), who has expertise and capabilities that a common user does not have, for periodically auditing the outsourced data [1] and can simultaneously secure his data from malicious data users. Most importantly, our technique is privacy-preserving, which will never reveal the data contents to the public auditor.
Our first scheme removes the burden of verification from the customer along with it, our second scheme reduces the number of keys, which alleviates both the customer’s and storage service’s fear of data leakage, and provides a method for independent arbitration of data retention contracts. Recently, the notion of public auditability has been proposed in the context of ensuring remotely stored data integrity under different system and security models [8], [12], [10], [7]. They do not consider the privacy protection of users’ data against external auditors and malicious users inside. Indeed, they may potentially reveal user’s data to auditors. This severe drawback greatly affects the security of these protocols in cloud computing [5]. There are legal regulations, such as the US Health Insurance Portability and accountability Act (HIPAA) [15], further demanding the outsourced data not to be leaked to external parties [9]. Our work is among the first few ones to support privacy-preserving public auditing along with strong data confidentiality using threshold cryptography in cloud computing, with a focus on data storage [5]. Considering all the security problems for privacy preserving our technique uses public key-based Homomorphic Linear Authenticator (HLA) scheme [8], [12], [7], 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 [5]. To achieve fine-grained data access control, we have used capability list [16]. In this paper, the approach has used the modified Diffie-Hellman algorithm to generate one time shared session-key between CSP and user to protect the data from outsiders. Our protocol guaranties that third party auditor will not gain any knowledge about the owner’s data stored on cloud server. Our contribution can be stated as, first of all we are providing privacy preserving public auditing system for secure cloud storage. Secondly our scheme enables third party auditor to audit the user’s data without gaining the knowledge about the data. Our scheme achieves batch auditing where multiple delegated auditing tasks from different users can be performed simultaneously by the TPA in a privacy-preserving manner.

The rest of the paper is organized as follows: Section 2 overviews the related work, section 3 introduces the proposed method. Finally, Section 4 gives the concluding remark of the whole paper.

II. RELATED WORK

Anthony Bisong et al. [3] have discussed security risks and concerns in cloud computing and enlightened steps that an enterprise can take to reduce security risks and protect their resources. They have also discussed the cloud computing security concerns and the security risk associated with enterprise cloud computing including its threats, risk and vulnerability. Mohammed A. et al. [6] has provided the survey of solutions for security risks like using a hash function for data integrity by keeping a short hash in local memory, if the amount of data is large, then a hash tree is the solution. Although the previous methods allow data owner to ensure the integrity of their data which has been returned by servers, they do not guarantee that the server will answer a query without knowing what that query is and whether the data is stored correctly in the server or not. In accordance with our first scheme of public auditing, Juels et al. [10] has described a “proof of retrievability” (PoR) model. In this model spot-checking and error-correcting codes are used for ensuring both “possession” and “retrievability” of data files stored on remote service systems. One of the most important drawback of this scheme is that the number of audit challenges a user can perform is fixed a priori, and public auditability is not supported in their main scheme. They also have described a straightforward Merkle-tree construction for public PoRs, this approach only works with encrypted data [5]. Similarly Bowers et al. [9] propose an improved protocol for POR that generalizes Juels’ work. In this protocol a server/archive proves to a client that a target file is intact, the client can retrieve all of target file from the server with high probability. In POR, a client might simply download target file itself and check an accompanying digital signature. Juels and Kaliski and related constructions adopt a challenge-response format that achieves much lower (nearly constant) communication complexity—as little as tens of bytes per round in practice. Their most practical constructions, though, support only a limited number of POR challenges. Y. Dodis, et al. [11] has also given a study of different variants of PoR with private auditability. Shacham and Waters [12] have provided an improved PoR scheme with full proofs of security against arbitrary adversaries in the strongest model, that of Juels and Kaliski. Their first scheme built from BLS signatures and secure in the random oracle model. It has the shortest query and response of any proof-of-retrievability with public verifiability. Second scheme, which builds elegantly on pseudorandom functions (PRFs) and is secure in the standard model, has the shortest response of any proof-of-retrievability scheme with private verifiability (but a longer query). Both schemes rely on Homomorphic properties to aggregate a proof into one small authenticator value. Again, their approach is not privacy preserving as their public auditability scheme exposes the linear combination of sampled blocks to external auditor. When used directly, their protocol is not provably privacy preserving, and thus may leak user data information to the external auditor. Shah et al. [14], [9] propose a TPA to keep online storage honest. It is done by first
encrypting the data then sending a number of precomputed symmetric-keyed hashes over the encrypted data to the auditor. The auditor verifies the integrity of the data file and the server’s possession of a previously committed decryption key. This scheme only works for encrypted files, requires the auditor to maintain state, and suffers from bounded usage, which potentially brings in online burden to users when the keyed hashes are used up. Ateniese et al. [8] have formalized a model called provable data possession (PDP). In this model, data is preprocessed by the client, and metadata used for verification purposes is produced. The file is then sent to an untrusted server for storage, and the client may delete the local copy of the file. The client keeps some (possibly secret) information to check server’s responses later. The server proves the data has not been tampered with by responding to challenges sent by the client. Again, their approach is not privacy preserving as their public auditability scheme exposes the linear combination of sampled blocks to external auditor. When used directly, their protocol is not provably privacy preserving, and thus may leak user data information to the external auditor. However, PDP apply only to the case of static, archival storage, i.e., a file that is outsourced and never changes. C. Chris Erway et al. [13] further provided Dynamic Provable Data Possession (DPDP) scheme which extends the PDP model to support provable updates on the stored data. DPDP solution is based on a new variant of authenticated dictionaries, where they use rank information to organize dictionary entries. Thus enabling users to support efficient authenticated operations on files at the block level.

In accordance with our second scheme of threshold cryptography a quantitative risk and impact assessment framework (QUIRC) is presented in [17], to assess the security risks associated with cloud computing platforms, which defines risk as a combination of the Probability of a security threat event and it's Severity measured as its impact. This approach gives vendors, customers and regulation agencies the ability to comparatively assess the relative robustness of different cloud vendor offerings and approaches in a defensible manner. It also can be helpful in alleviating the considerable Fear, Uncertainty and Doubt associated with cloud platform security issues and helping that they are dealt in an effective way. Limitations of the approach include that it requires the meticulous collection of input data for Probabilities of events which requires collective industry SME (Subject Matter Experts) inputs. The scheme proposed in [18] have used group-key scheme, there is a single key corresponding to each group of users for decryption process and all users of the group know that key. Here, number of keys is reduced but there is a problem of collusion attack of CSP and a user because a single malicious user can leak whole data of the group to CSP. Sunil Sanka et al. proposed a capability based access control technique [19] that ensures only valid users will access the outsourced data. In this scheme, data are encrypted by symmetric keys and symmetric keys are known only to data owner and corresponding data users. The encrypted data are stored at CSP. CSP can’t see data stored at it as data are encrypted. Data are further encrypted by one time secret session-key shared between CSP and user by the modified Diffie-Hellman protocol to protect data from outsiders during the transmission between CSP and user. This scheme provides whole data security but there is associated a key corresponding to each user and users may be large in number in some applications. So, number of keys may increase. Hence, increases the maintenance and security concerns of keys consistency and complexity of the proposed method can be considered for improvement.

III. PROPOSED METHOD

To ensure the integrity, availability, confidentiality of the data stored on cloud we are proposing two techniques first one is a public auditing and second one is threshold cryptography. In our system we are considering four core entities of cloud storage system such as Data owner who has large amount of data which he needs to be store on cloud server; cloud server has a storage space for the user’s data, which is managed by cloud service provider. The third party auditor (TPA), who has expertise and capabilities that users do not, can periodically check the integrity of all the data stored in the cloud on behalf of the users, which provides a much more easier and affordable way for the users to ensure their storage correctness in the cloud [15]. And fourth one is cloud data users which uses data owner’s stored data. Architecture of our proposed method is shown in figure 1. Using third-party auditing service provides a cost-effective method for users to gain trust in cloud [5]. Our first scheme uses technique of public key-based homomorphic linear authenticator (HLA), it enables TPA to perform the public auditing without need of local copy of data and thus minimizes 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 will not gain any knowledge about the user’s data content stored in the cloud server (CS) during the efficient auditing process. Our second scheme uses capability list to achieve fine-grained data access control. Our approach has used the modified Diffie-Hellman algorithm to generate one time shared session-key between CSP and user to protect the data from outsiders. To ensure data integrity the approach has used MD5 [16].

http://www.ijctjournal.org  Page 89

ISSN: 2231-2803
Advantages of proposed scheme

1. In our first proposed scheme we motivate the public auditing system of data storage security in Cloud Computing and provide a privacy-preserving auditing protocol.
2. Our scheme enables an external auditor to audit user’s cloud data without learning the data content.
3. To the best of our knowledge, our scheme is the first to support scalable and efficient privacy preserving public storage auditing in Cloud along with threshold cryptography.
4. Our second scheme not only provides the strong data confidentiality but also reduces the number of keys required to access the cloud data securely.
5. By employing the threshold cryptography at the user side, we protect outsourced data from collusion attack. Since, DO stores its data at CSP in encrypted form and, keys are known only to DO and respected users group, data confidentiality is ensured.

Basic Schemes proposed in paper:

a. Third Party Auditor
To achieve privacy-preserving public auditing, we propose to uniquely integrate the homomorphic linear authenticator with random masking technique. Our design makes use of a public key-based HLA, to equip the auditing protocol with public auditability. Specifically, we use the HLA proposed in [5], which is based on the short signature scheme. The auditing process will be carry out in two phases which consists of four algorithms.

i. Setup Phase
The data owner initializes the public and secret parameters of the system by executing KeyGen, and preprocesses the data file F by using SigGen to generate the verification metadata. The data owner then stores the data file F and the verification metadata at the cloud server, and delete its local copy. As part of preprocessing, the data owner may alter the data file F by expanding it or including additional metadata to be stored at server.

ii. Audit Phase:
The TPA issues an audit message or challenge to the cloud server to make sure that the cloud server has retained the data file F properly at the time of the audit. The cloud server will derive a response message by executing GenProof using F and its verification metadata as inputs. The TPA then verifies the response via VerifyProof.

b. Threshold Cryptography
In this scheme we are presenting a model for secure communication between different entities and secure access to data. There are four algorithms in the proposed scheme. Algorithm 1 describes secure communication of data between DO and CSP moreover this algorithm insures data confidentiality and, authentication of DO and CSP. Algorithm 2 describes procedures which DO and CSP apply after a new file creation in respect. Algorithm 3 describes about secure communication of data between CSP and user. In this algorithm user’s authorization is also checked. Algorithm 4 describes the threshold cryptography technique for decryption of a user’s file. Algorithm 4 is applied at user side where number of keys is reduced (one key corresponding to one group) and no threat of collusion attack as in group-key scheme.

IV. CONCLUSION
In this paper we have proposed two schemes in order to achieve cloud data security, availability and integrity. Our first scheme is public auditing system where have used Homomorphic Linear Authenticator as third party authenticator. TPA ensures data integrity without gaining knowledge about data during auditing process. In our second scheme we have introduced threshold cryptography technique. By employing the threshold cryptography at the user side, we protect outsourced data from collusion attack. Since, DO stores its data at CSP in encrypted form and, keys are known only to DO and respected users group, data confidentiality is ensured. To ensure fine-grained access control of outsourced data, the scheme has used capability list. Public key cryptography and MD5 ensure the entity
authentication and data integrity respectively. In this scheme we have reduced number of keys required so as its maintenance overhead. We can further extend our privacy-preserving public auditing protocol into a multiuser setting, where the TPA can perform multiple auditing tasks in a batch manner for better efficiency.

ACKNOWLEDGMENT

Reshma Suryawanshi would like to thank to her guide Assistant Prof. S.N.Shelke for his guidance and instructive comments in this paper work. Lastly, authors would like to offer regards to all of those who supported in any respect during the completion of this paper.

REFERENCES