INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN COMPUTER SCIENCE AND ENGINEERING TECHNOLOGIES

ISSN 2454-9924 Volume: 1 Issue: 1(2015)

Providing Security for Public Data Auditability in Regeneration based Cloud Storage

R. Saranya #1, S. Vanakovarayan *1

Mailam Engineering College, Mailam #1, *1

Saransweety08@gmail.com #1

Abstract: Storing data in a third party's cloud system provides serious anxiety across data privacy. To make sure remote data truth and fault tolerance often lacks the support of either public auditability or changing data operations. Task of allowing a trusted party auditor (TPA), on behalf of the cloud client, to verify the integrity of the dynamic data stored in the cloud. To provide efficient data dynamics, we improve the existing proof of storage models by implementing block tag authentication. Additionally, implement the technique of signature to extend our main result into a multi-user setting, where TPA can perform multiple auditing tasks simultaneously. And also we use a proxy agent to verify the signature technique and to determine the regeneration problem in case of failed authenticators and data owner.

Keywords: Third Party Auditor, Anxiety, Signature, Tolerance

Introduction

Cloud storage is now obtaining popularity because it offers a flexible on-demand data outsourcing service with providing benefits: relief of the load expenses for storage management, universal data access with location independence, and avoidance of capital expenditure on hardware, software, and personal maintenances, etc. However, this new paradigm of data hosting service also brings new security threats toward user's data, thus making individuals or enterprisers still feel hesitant. It is noted that data owners lose ultimate control over the fate of their outsourced data; thus, the correctness, availability and integrity of the data are being put at risk. On the one hand, the cloud service is usually faced with a broad range of internal/external adversaries, who would maliciously delete or corrupt users' data; on the other hand, the cloud service providers may act dishonestly, attempting to hide data loss or corruption and claiming that the files are still correctly stored in the cloud for reputation or monetary reasons. Thus it makes great sense for users to implement an efficient protocol to perform periodical verifications of their outsourced data to ensure that the cloud indeed maintains their data correctly. Many mechanisms dealing with the integrity of outsourced data without a local copy have been proposed under different system and security models up to now. The most significant work among these studies is the PDP (provable data possession) model and

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN COMPUTER SCIENCE AND ENGINEERING TECHNOLOGIES

ISSN 2454-9924 Volume: 1 Issue: 1(2015)

POR (proof of irretrievability) model, which were originally proposed for the single-server scenario. Considering that files are usually striped and redundantly stored across multi-servers or multi-clouds, explore integrity verification schemes suitable for such multi-servers or multi clouds setting with different redundancy schemes, such as replication, erasure codes, and, more recently, regenerating codes. In this paper, we focus on the integrity verification problem in regenerating-code-based cloud storage, especially with the functional repair strategy. Similar studies have been performed separately and independently. Extended the single-server CPOR scheme(private version in to the regenerating code-scenario; designed and implemented a data integrity protection(DIP) scheme for FMSR based cloud storage and the scheme is adapted to the thin-cloud setting. However, both of them are designed for private audit, only the data owner is allowed to verify the integrity and repair the faulty servers. Considering the large size of the outsourced data and the user's constrained resource capability, the tasks of auditing and reparation in the cloud can be formidable and expensive. 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 their outsourced data (in additional to retrieving it). In particular, users may not want to go through the complexity in verifying and reparation. The auditing schemes in imply the problem that users need to always stay online, which may impede its adoption in practice, especially for long-term archival storage. To fully ensure the data integrity and save the users' computation resources as well as online burden, we propose a public auditing scheme for the regenerating-code-based cloud storage, in which the integrity checking and regeneration (of failed data blocks and authenticators) are implemented by a third party auditor and a semi-trusted proxy separately on behalf of the data owner. Instead of directly adapting the existing public auditing scheme to the multi-server setting, we design a novel authenticator, which is more appropriate for regenerating codes. Besides, we "encrypt" the coefficients to protect data privacy against the auditor, which is more lightweight than applying the proof blind technique and data blind method. Several challenges and threats spontaneously arise in our new system model with a proxy, and security analysis shows that our scheme works well with these problems.

Related Work

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. It does not support fault tolerance in case of failed authenticators. 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.

Drawbacks

- Especially to support block insertion, which is missing in most existing schemes.

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN COMPUTER SCIENCE AND ENGINEERING TECHNOLOGIES

ISSN 2454-9924 Volume: 1 Issue: 1(2015)

- Authentication was provided only at the time of upload not at the time of download.
- Data integrity is not maintained.
- Fault tolerance is not provided.
- Only single copy of data is stored.

Proposed Work

Here we are providing better security in owner's upload side as well as on the download side. For better security client splitting that single file into different blocks and providing a unique identification number for each block. 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. 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. Trusted Party Auditor (TPA): 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. Proxy agent: semi-trusted and acts on behalf of the data owner to regenerate authenticators and data blocks on the failed servers during the repair procedure

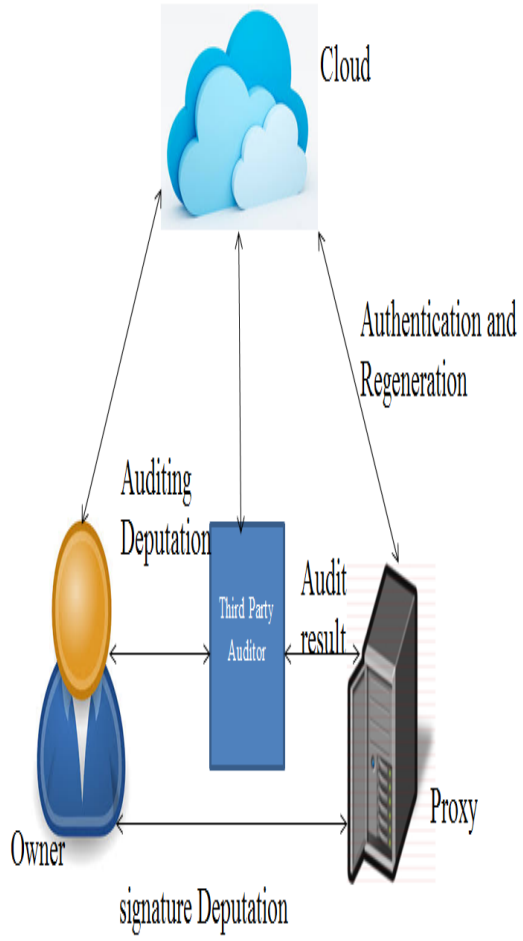
Benefits

- 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.
- We extend our scheme to support scalable and efficient public auditing in Cloud Computing. In particular, our scheme achieves auditing tasks from different users can be performed simultaneously by the TPA.
- Spitted into blocks and it is uploaded in the cloud in a encrypted format for better security.
- We prove the security of our proposed construction and justify the performance of our scheme through concrete implementation and comparisons.

System Design

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN COMPUTER SCIENCE AND ENGINEERING TECHNOLOGIES

ISSN 2454-9924 Volume: 1 Issue: 1(2015)



Components involved in system design

- Data owner
- Cloud
- Third party Auditor(tpa)
- Proxy agent

The data owner owns large amounts of data files to be stored in the cloud. And the cloud which is managed by the cloud service provider, provide storage service and have significant computational resources. The third party auditor who has expertise capabilities to conduct public audits on the coded data in the cloud, the TPA is trusted and its audit result is unbiased for both data owners and cloud servers. And a proxy agent who is semi trusted and acts on behalf of the data owner to regenerate authenticators and data blocks on the failed servers during the repair procedure. The data owner is restricted in computational and storage resources compared to other entities and may become offline even after the upload procedure. The TPA and proxy are much more powerful than the data owner but less than the cloud servers in terms of computation and memory capacity. To save resources as well as the online burden potentially brought by the periodic auditing and accidental repairing, the data owners resort to the TPA for integrity verification and delegate the reparation to the proxy.

Algorithm and description

The user can also encrypt data before outsourcing it into the cloud server with encryption techniques .As a significant research area for system protection, data access control has been evolving in the past thirty years and various techniques have

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN COMPUTER SCIENCE AND ENGINEERING TECHNOLOGIES

ISSN 2454-9924 Volume: 1 Issue: 1(2015)

been developed to effectively implement fine-grained access control, which allows flexibility in specifying differential access rights of individual users. Traditional access control architectures usually assume the data owner and the servers storing the data are in the same trusted domain, where the servers are fully entrusted as an omniscient reference monitor responsible for defining and enforcing access control policies. This assumption however no longer holds in cloud computing since the data owner and cloud servers are very likely to be in two different domains. With the character of low maintenance, cloud computing provides an economical and efficient solution for sharing group resource among cloud users. Unfortunately, sharing data in a multi-owner manner while preserving data and identity privacy from an untrusted cloud is still a challenging issue, due to the frequent change of the membership. In this paper, we propose a secure multi-owner data sharing scheme, named Mona, for dynamic groups in the cloud. By leveraging group signature and dynamic broadcast encryption techniques, any cloud user can anonymously share data with others. Meanwhile, the storage overhead and encryption computation cost of our scheme are independent with the number of revoked users. In addition, we analyze the security of our scheme with rigorous proofs, and demonstrate the efficiency of our scheme in experiments.

Ring signature:

Holomorphic Authenticators Ring Signatures Rijndael Managed, Suppose that a group of entities each have public/private key pairs, (PK1, SK1), (PK2, SK2), ... ,(PKn, SKn). Party i can compute a ring signature σ on a message m, on input (m, SKi, PK1, ... , PKn). Anyone can check the validity of a ring signature given σ, m, and the public keys involved, PK1, ... , PKn. If a ring signature is properly computed, it should pass the check. On the other hand, it should be hard for anyone to create a valid ring signature on any message for any group without knowing any of the secret keys for that group. Ring signatures, rst introduced by Rivest, Shamir, and Tauman, enable a user to sign a message so that a ring of possible signers (of which the user is a member) is identified, without revealing exactly which member of that ring actually generated the signature. In contrast to group

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN COMPUTER SCIENCE AND ENGINEERING TECHNOLOGIES

ISSN 2454-9924 Volume: 1 Issue: 1(2015)

signatures, ring signatures are completely ad-hoc" and do not require any central authority or coordination among the various users (indeed, users do not even need to be aware of each other); furthermore, ring signature schemes grant users ne-grained control over the level of anonymity associated with any particular signature. This paper has two main areas of focus. First, we examine previous definitions of security for ring signature schemes and suggest that most of these prior definitions are too weak, in the sense that they do not take into account certain realistic attacks. We propose new definitions anonymity and un-forget ability which address these threats, and give separation results proving that our new notions are strictly stronger than previous ones. Second, we show the rst constructions of ring signature schemes in the standard model. One scheme is based on generic assumptions and satisfies our strongest definitions of security. Two additional schemes are more recent, but achieve weaker security guarantees and more limited functionality.

Methodology

- User registration
- Mail alert process
- Block insertion or block tag

1. User registration

Users can use them from anywhere at any time. For example, the email service is probably the most popular one. Cloud computing is a concept that treats the resources on the Internet as a unified entity, a cloud. Users just use services without being concerned about how computation is done and storage is managed. In this paper, we focus on designing a cloud storage system for robustness, confidentiality, and functionality. A cloud storage system is considered user interface entry level creation in this module. A user should register their details first such as their name, email id, mobile no, and an id for that particular client this is used to avoid the duplication of entries in this application.

2. Mail alert process

The uploading process of the user is first get the secret key in the corresponding user email id and then apply the secret key to encrypted data to send the server storage and

ISSN 2454-9924 Volume: 1 Issue: 1(2015)

decrypts it by using his secret key to download the corresponding data file in the server storage system's the secret key conversion using the Share Key Gen (SKA, t, m). This algorithm shares the secret key SKA of a user to a set of key servers, this key was generated and sent to the registered client's mail id as per the notifications which I have received from your end.

3. Block insertion or Block tag

Splitting the source content into 9 different blocks and have to give a tag for that all blocks for client identification, here that user defined tags are going to convert in to a binary value that's via ASCII table using Horner method. Dividing of these blocks makes difficult for attacker to predict the combinations also and the generated ASCII value acts as a metadata key value and each block is send along with the metadata key.

Conclusion

From this the providing preserving public auditing for regenerating code based cloud storage has been implemented. Where the data owners are privileged to delegate TPA for checking entitled their data validity. To protect the original data privacy against the TPA, we randomize the coefficients in the beginning rather than applying the blind technique during the auditing process. Determining that the data owner cannot always stay online in practice, in order to keep the storage available and verifiable after a malicious corruption, we introduce a semi-trusted proxy into the system model and provide a privilege for the proxy to handle the reparation of the coded blocks and authenticators. To better exact for the regenerating-code-scenario, we design our authenticator based on the BLS signature. This authenticator can be efficiently generated by the data owner simultaneously with the encoding procedure. Additional analysis shows that our scheme is provable secure, and the performance evaluation shows that our scheme is highly effective t and can be feasibly integrated into a regenerating-code-based cloud storage system.

References

[1] A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, and I. Stoica, "Above the clouds: A

ISSN 2454-9924 Volume: 1 Issue: 1(2015)

Berkeley view of cloud computing," Dept. Electrical Eng. and Comput. Sciences, University of California, Berkeley, Rep. UCB/EECS, vol. 28, p. 13, 2009.

[2] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in Proceedings of the 14th ACM Conference on Computer and Communications Security, ser. CCS '07. New York, NY, USA: ACM, 2007, pp. 598–609.[3] A. Juels and B. S. Kaliski Jr, "Pors: Proofs of retrievability for large files," in Proceedings of the 14th ACM conference on Computer and communications security. ACM, 2007, pp. 584–597.

[4] R. Curtmola, O. Khan, R. Burns, and G. Ateniese, "Mr-pdp: Multiplereplica provable data possession," in Distributed Computing Systems, 2008. ICDCS'08. The 28th International Conference on. IEEE, 2008, pp. 411–420.

[5] K. D. Bowers, A. Juels, and A. Oprea, "Hail: a high-availability and integrity layer for cloud storage," in Proceedings of the 16th ACMconference on Computer and communications security. ACM, 2009, pp. 187–198.

[6] J. He, Y. Zhang, G. Huang, Y. Shi, and J. Cao, "Distributed data possession checking for securing multiple replicas in geographically dispersed clouds," Journal of Computer and System Sciences, vol. 78, no. 5, pp. 1345–1358, 2012.

[7] B. Chen, R. Curtmola, G. Ateniese, and R. Burns, "Remote data checking for network coding-based distributed storage systems," in Proceedings of the 2010 ACM workshop on Cloud computing security workshop. ACM, 2010, pp. 31–42.

[8] H. Chen and P. Lee, "Enabling data integrity protection in regenerating coding-based cloud storage: Theory and implementation," Parallel and Distributed Systems, IEEE Transactions on, vol. 25, no. 2, pp. 407–416, Feb 2014.