ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

A NEW FRACTAL MERKLE HASH TREE CONSTRUCTION FOR PROOF OF RETRIEVABILITY IN CLOUD

D.Abinaya[1], Dr.P.Vivekanandan[2],D.Ramesh[3]
PG scholar[1], Head of the Department[2], Assistant Professor[3]
[1,2,3] Department of Computer Science and Engineering,
[1,2,3] Park college of Engineering and Technology,
d.abinaya01@gmail.com

Abstract:

Cloud computing moves the application software and databases to the central large data centers, in which the management of data and services may not be fully trusted. In particular, we consider the problem of having a third party auditor (TPA), to check saved data on behalf of Cloud Client for the integrity of dynamic data in the cloud. The support for information about the general dynamism of the data operation, such as block modification, insertion and deletion, is also an important step toward practicality. While earlier works on ensuring data integrity often lack the support of both public verifiability or dynamic data operations, this paper reaches both. We have to first identify the difficulties and potential security issues direct extensions with fully dynamic data updates from the prior works and then show you how to build an elegant verification system for the seamless integration of these two outstanding features in our protocol design. To achieve efficient data dynamics, the existing evidence storage models are manipulated by the classic fractal merkle hash tree structure for block tag authentication. In order to support efficient handling of multiple test objects, we have continued to explore the technique of bilinear aggregate signature to extend our main result in a multiuser environment where TPA can perform multiple tasks at the same examination.

Keywords: cloud computing, fractal merkle hash tree, public auditability, proof of retrievability

Introduction:

Cloud computing has emerged as the next generation of information technology (IT) architecture. It has been introduced for companies because of its long list of unprecedented advantages in the IT history: on-demand self-service, ubiquitous network access, location-independent resource pooling, rapid resource elasticity , usage-based pricing and transfer of risk.

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

An essential aspect of this paradigm shift is that data stored centrally or in the out cloud. It brings new and challenging security threats toward outsourced data of users. Since Cloud Service Provider (CSP) separate administrative units, data outsourcing actually waiving ultimate user control of the fate of their data. As a result, the accuracy of data in the cloud will be questioned.

As user no longer physically possesses the storage of our data, traditional cryptographic primitives can be taken over protection not directly for the purpose of data security. To fully ensure the integrity of data and store computing means of cloud users and online exposure, it is of crucial importance for the public auditing service for cloud data storage, so users can use an independent third party auditor (TPA) to check the outsourced data when needed.

A cryptographic module known as proof of retrievability (POR) allows a user (verifier) to determine that an archive (proofer) "owns" a file or data object F. Specifically successfully executed POR ensures a verifier that the auditor provides a protocol interface, through which the reviewer can retrieve file F in its entirety. In this paper, the inspection body be the TPA or client.

Another important concern is to provide support for dynamic operations while maintaining data integrity and allow the TPA for batch auditing to the multi-client applications. Our contribution in this paper can be summarized as improving the existing storage models using the classic fractal hash tree for tag generation and integrate the bilinear aggregate signature for batch auditing support.

Related Work

Ateniese et al. defines the "provable data possession" (PDP) model for guaranteeing ownership of files on untrusted storage. Their suggestion also supports the first proof-of-storage scheme, the public verifiability. The system uses RSA-based homomorphic tags for auditing outsourced data so that a linear combination of file blocks can be combined in a single block and verified through the use of homomorphic property of the RSA. However, the owner of the data has to be calculated, a large number of tags for the data to be outsourced, which usually involves potentiation and multiplication operations.

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

Juel and Kaliski Jr. was a "proof of retrievability" (POR) model, with sampling and error correction codes are adopted to ensure both "possession" and "retrievability" of files in the archive service systems. However, verifiability is not publicly supported in their system and the data owner also has many computational effort to produce tags for the making are outsourced this data.

The POR scheme offers the user the assurance that the file can be accessed by the clients and integrity is maintained. While OPoR scheme of Jin Li Xiao Tan, Chen Xiaofeng, Duncan S. Wong, and Fatos Xhafa uses public auditing and ensures data integrity. The scheme provides support for dynamic data operations in resisting the reset attack. The robustness against reset attack ensures that a malicious server memory can never gain an advantage of passing the examination of a false saved file by resetting the client (or the audit server) in the upload phase.

Cloud Model

Representative network architecture for cloud data storage is in Figure 1. Three different network units can be determined as follows.:

Cloud user: A company that large files must be stored in the cloud, and relies on the cloud for data maintenance and the calculation can either be individual consumers or organizations.

Cloud storage server: An entity that is managed by a cloud service provider, has significant storage and computing resources to obtain the customer's data. The CSS is required to provide the integrity proof to the customer or cloud Audit server during the integrity check phase.

Cloud Audit Server: A TPA, the expertise and capabilities that customers do not have, trustworthy must be assessed, and put risk of cloud storage services on behalf of clients on request. In this system, the Cloud Audit server also generates all the tags of the files for the user before we upload them to the cloud storage servers.

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

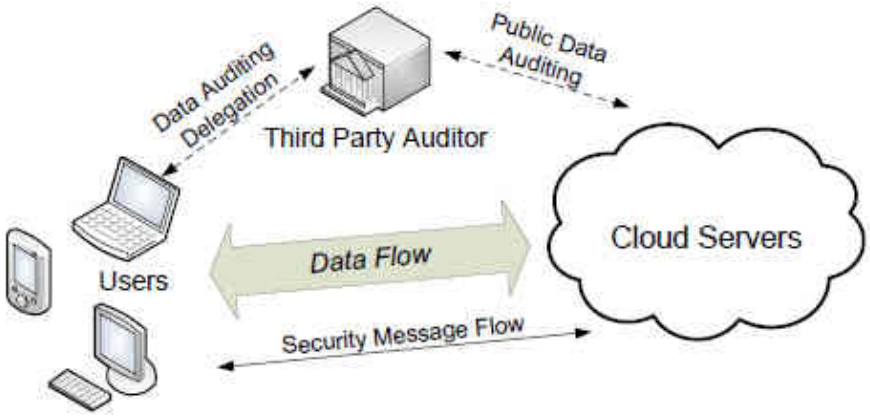

FIG 1: CLOUD MODEL

In the cloud paradigm by storing large files on the remote server, the client, can be free from the burden of memory and computing. When cloud client no longer own their data locally, it is crucial for customers to ensure that their data is stored properly and maintained. In other words, customers should be equipped with certain safety measures so that it checks at regular intervals the accuracy of remote data even without the existence of local copies. If the customers do not have the time or resources feasibility to monitor their data, they can delegate the monitoring task to a trusted cloud audit server or Third party Auditor (TPA) of their respective decisions.

Construction Of POR Schemes Using Fractal Merkle Hash Tree

Notations And Preliminaries

Merkle hash tree (MHT) tree is a well-studied authentication structure, which is intended to efficiently and securely prove that a number of elements are intact and unchanged. It is formed as a binary tree, where the leaves in the MHT are the hash values of the authentic data values.

Fractal merkle tree traversal is to output the leaf pre-images and authentication paths, sequentially.

Pebble: A pebble on a node in the tree T when we value P (n) associated with this node.

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

Exist subtrees are a stacked series of h-subtrees, indexed $\{\{Existig_i\}^{L}_{i=1}$, which include the authentication path for the current signature.

Desired trees are a stacked series of h-subtrees, indexed $\{Desired_i\}^{L}_{i=1}$. Each Desired trees is next to a tree Exist part at the same level. If Exist subtree no longer contains the next authentication path, it is replaced with its Desired trees counterpart. The desire trees are built gradually according to each output of the algorithm, thus amortizing the operations required to evaluate the subtree.

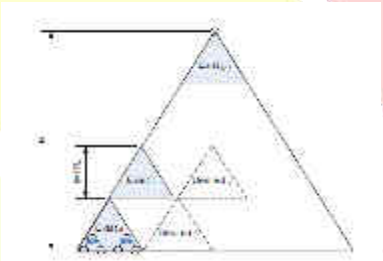

Fig 2 Fractal Merkle Tree Notations

Fractal Merkle Construction

The files are stored with cloud audit server by using the OPoR schemes generating the merkle hash tree. When the audit server starts verification process it uses fractal merkle hash tree traversal algorithm which involves three phases: the key generation phase; the output phase; and the verification phase.

The key generation phase (which can be performed offline by a relatively powerful computer) calculates the root of the tree and output, taking the role of a public key. Moreover, the iterative output phase needs some setup, namely the calculation of the pebbles on the initial existing subtrees. These are stored on the computer that stores the output phase.

The output phase consists of a number of rounds. During the round j, the (previously unreleased) pre-image of the j-th leaf is issued, along with its authentication path. In addition, a number of pebbles is discarded and some number of pebbles is calculated in order to prepare for future spending.

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

The verification phase is identical to the conventional verification phase for Merkle trees.

Key Generation and Setup

1 Initial Authentication Nodes: For each $h \in \{0,1,...,H-1\}$:

Calculate $Auth_h = P(n_{h,1})$.

2 Initial Next Nodes: For each $h \in \{0,1,...,H-1\}$: Set up $Stack_h$ with the value of the sibling of $Auth_h - P(n_{h,0})$ (left nodes).

3 Public Key: Calculate and publish tree root value P(n root)

Output and Update:

1 Set leaf =0

2 Output:

• Compute and output $P(n_{leaf}) = LEAFCALC(leaf)$.

• For each $h \in [0,H-1]$ output $\{Auth_h\}$.

3 Refresh Authentication Nodes:

For all h such that 2^h/ leaf +1:

• Let $Auth_h$ become equal to the only node value in $Stack_h$. Empty the stack.

• Set startnode=$(leaf+1+2^h) \oplus 2^h$.

• $Stack_h$. initialize(startnode, h) .

4 Build Stacks:

For all $h \in [0,H-1]$:

• $Stack_h$. update(2)

5 Loop:

• Set leaf =leaf +1.

• If leaf $< 2^H - 1$ go to step 2, otherwise stop.

One time signatures

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

FMTseq - Fractal Merkle Tree sequential signatures combines Merkle's one-time signatures with Jakobsson et al.'s algorithm for hash tree traversal. Our scheme differs by providing many more one-time signatures with the same hash tree. The secrets of each one-time signatures are generated by a pseudorandom number generator. The value of each leaf of the fractal Merkle tree is a hash over all the commitments of a single one-time signature. Therefore, each leaf serves as a public commitment to a one-time signature. For each one-time signature, the signer regenerates the next unused leaf, reveals the required secrets, and outputs the commitments of the unrevealed secrets and the authentication path.

Performance Evaluation

For a selection of the parameters is the total space requirement by 1.5 log2 N / log (log N) hash values, and in the worst case computational complexity 2 log N / log (log N) hash function evaluations per output is limited. By reducing the time or space costs, we find that for medium - size trees, the computational effort can be made efficient enough for practical use. This reinforces the belief that practical, secure signature / authentication protocols can be implemented.

Conclusion

To ensure cloud data storage security, it is important to enable TPA to evaluate the service quality of an objective and independent perspective. Public auditability also enables customers to delegate auditing tasks to TPA for the integrity verification. This paper proposes the proof of retrievability scheme with public auditability using the Third Party Auditor which supports dynamic operations. in order to reduce the space and time during the verification and auditing process instead of traditional merkle tree traversal , the classical fractal merkle tree traversal is introduced.

There are several interesting issues that do along this line of research. For example, we can (1) reduce the reliance on the cloud audit server for more generic applications, (2) strengthening security model against restoring attacks in the data integrity verification protocol, and (3), the proposed approach persistent to other data Programs such as NoSQL databases. We leave the study of these problems, as our future work.

References

- G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in Proc. 14th ACM Conf. Comput. Commun. Security, 2007, pp. 598–609.
- A. Juels and B. S. Kaliski Jr, "Pors: Proofs of retrievability for large files," in Proc. 14th ACM Conf. Comput. Commun. Security, 2007,pp. 584–597**.**
- H. Shacham and B. Waters, "Compact proofs of retrievability," inProc. 14th Int. Conf. Theory Appl. Cryptol. Inf. Security, 2008, pp. 90–107.
- K. D. Bowers, A. Juels, and A. Oprea, "Proofs of retrievability:Theory and implementation," in Proc. ACM Workshop Cloud Comput. Security, 2009, pp. 43–54.
- T. S. J. Schwarz and E. L. Miller, "Store, forget, and check: Using algebraic signatures to check remotely administered storage," in Proc. 26th IEEE Int. Conf. Distrib. Comput. Syst., 2006, p. 12.
- C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in Proc. IEEE Conf. Comput. Commun., 2010, pp. 525–533.
- Y. Zhu, H. Hu, G.-J. Ahn, and M. Yu, "Cooperative provable data possession for integrity verification in multicloud storage," IEEE Trans. Parallel Distrib. Syst., vol. 23, no. 12, pp. 2231–2244, Dec. 2012.
- C. Wang, Q. Wang, and K. Ren, "Ensuring data storage security incloud computing," in Proc. 17th Int. Workshop Quality Serv., 2009,pp. 1–9.
- Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," in Proc. 14th Eur. Symp. Res. Comput. Security, 2009,pp. 355–370.
- J. Li, X. Tan, X. Chen, and D. S. Wong, "An efficient proof of retrievability with public auditing in cloud computing," in Proc. 5th Int. Conf. Intell. Netw. Collaborative Syst., 2013, pp. 93–98.
- D. Boneh and C. Gentry, "Aggregate and verifiably encrypted signatures from bilinear maps," in Proc. 22nd Int. Conf. Theory Appl. Cryptograph. Tech., 2003, pp. 416–432.

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

- D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," in Proc. 7th Int. Conf. Theory Appl. Cryptol. Inf. Security,2001, pp. 514–532.
- Markus Jakobsson, Tom Leighton, Silvio Micali, and Michael Szydlo "Fractal Merkle Tree Representation and Traversal" in Proc. The Cryptographers' Track at the RSA Conference 2003, pp 314-326