ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 3, March 2018

Data Integrity Checking in Dynamic Cloud

[1] R. Vinoth, [2]R.Vimalraj, [3]R. Sathishkumar, [4]R. Sarathkumar
[1][2][3][4] Department of Computer Science and Engineering,VSB Engineering College, Karur, Tamil Nadu, INDIA

***Abstract* – Cloud storage is a model of data storage in which the digital data is stored in logical pools, the cloud providers are responsible for keeping data available, accessible with security. A public auditing protocol allows a TPA (Third Party Auditor) to check the integrity protection in cloud computing a formidable task. In fact, the end devices may have low computational capabilities. The trusted third party auditing process should take in no new vulnerabilities towards user data privacy. The proposed work for improving the data integrity and data security by implementing the double encryption algorithm to encrypt the data twice and stored in the cloud server. The trusted third party auditing for the data modification happened on the first level encrypted layer of the file. The first level encryption key would be secure on the user side. In this project work, double encryption approach with public auditing protocol, we can enhance the data privacy preserving in the public cloud without leakage of data.**

Key Words: Cloud Storage, Public Auditing, Online/Offline signature, Double Encryption

1. INTRODUCTION

1.1 CLOUD COMPUTING

Cloud computing is a computing paradigm, where a large pool of systems are connected in private or public networks, to provide dynamically scalable infrastructure for application, data and file storage. With the advent of this technology, the cost of computation, application hosting, content storage and delivery reduced significantly. It is a practical approach to experience direct cost benefits and it has the potential to transform a data center from a capital-intensive set up to a variable priced environment. The idea of cloud computing is based on very fundamental principles of reusability of IT capabilities. The difference that cloud computing brings compared to traditional concepts of "grid computing", "distributed computing", "utility computing", or "autonomic computing" is to broaden horizons across organizational boundaries. Forrester defines cloud computing as "A pool of abstracted, highly scalable, and managed to compute infrastructure capable of hosting end-customer applications and billed by consumption". A technology uses the internet and central remote servers to maintain data and applications and allows consumers and businesses to use applications without installation and access their personal files at any computer with internet access. This technology allows for much more efficient computing by centralizing data storage, processing, and bandwidth. Cloud computing examples are Yahoo email, Gmail, or Hotmail.

1.2 CLOUD COMPUTING CHALLENGES

Despite its growing influence, concerns regarding cloud computing still remain. In our opinion, the benefits outweigh the drawbacks and the model is worth exploring. Some common challenges are:

- ✓ Data Protection
- ✓ Data Recovery and Availability
- ✓ Management Capabilities
- ✓ Regulatory and Compliance Restrictions

1.3 PRIVACY PRESERVING

While the storage of corporate data on remote servers is not a new development, the current expansion of cloud computing justifies a more careful look at its actual consequences involving privacy and confidentiality issues. As users no longer physically possess the storage of their data, traditional cryptographic primitives for the purpose of data security protection cannot be directly adopted. To fully ensure the data integrity and save the cloud users' computation resources as well as an online burden, it is of critical importance to enable public auditing service for cloud data storage, so that users may resort to an independent third party auditor (TPA) to audit the outsourced data when needed. The 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.

1.4 CLOUD DATA SECURITY

Data security is a crucial element that warrants security. Enterprises are reluctant to buy an assurance of business data security from vendors. They fear losing data to competition and the data confidentiality of consumer. In many instances, the actual storage location is not disclosed, adding to the security concerns of enterprises. In the existing models, firewall across data centers (owned by enterprises) protects this sensitive information.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 3, March 2018

2. RELATED WORK:

G. Ateniese, A. Faonio, and S. Kamara (2015) said In this paper provide a framework for constructing leakage-resilient ID protocols in the BRM from publicly verifiable proofs of storage (PoS) that are computationally zero-knowledge (ZK). PoS is interactive protocols allowing a client to verify that a server faithfully stores its file [1]. Z. Fu, K. RenA. (2016) a popular way to search over encrypted data is searchable encryption and many constructive schemes have been forward under different applications. One is that most of the existing schemes follow the model of "one size fits all" and ignore individual users' experience due to their different hobbies, interests or cultural backgrounds [2]. Z. Hao, S. Zhong (2011) In remote data integrity checking protocols, the client can challenge the server about the integrity of a certain data file, and the server generates responses proving that it has access to the complete and uncorrupted data [3]. H. Liu, L. Chen. (2015) In this paper, they have shown the construction not secure in their security model or in a correct security model. To be specific, with the aid of signature queries, a malicious cloud server could generate a valid response to a challenge from a third party auditor (TPA) even the server has deleted all the files of a user or has corrupted the file [4]. J. K. Liu, M. H. Au. (2016). As sensitive data may be stored in the cloud for sharing a purpose or convenient access; and eligible users may also access the cloud system for various applications and services, user authentication has become a critical component for any cloud system [5]. F. Sebe, J. Domingo-Ferrer. (2008) the protection of critical infrastructures is a priority for governments and companies. In the Dependable Intrusion Tolerance architecture (DIT), an integrity check is just one among the various building blocks used to detect corruption of remote data [6]. A. Shamir and Y. Tauman (2013)To fully ensure the data integrity and save the cloud users' computation resources as well as online burden, it is of critical importance to enable public auditing service for cloud data storage, so that users may resort to an independent third party auditor (TPA) to audit the outsourced data when needed [7]. It is often insufficient to detect the data corruption when accessing the data, as it might be too late for recover the data loss or damage [8]. L. Zhang, Q. WuGKA. (2015) is inefficient since the sender may change frequently. Further, with the standard round notion, the best-known GKA protocols require two or more rounds to establish a secret key [9]. Z. Xia, X. Wang. (2017)In order to obtain high search efficiency, we construct a tree-based index structure and propose a "Greedy Depth-first Search (GDFS)" algorithm based on this index tree [10].

3. SYSTEM ANALYSIS

3.1 Existing:

While cloud computing makes various advantages and also have challenging task in data security threats toward users's outsourced data. Since cloud service providers (CSP) are separate administrative entities, data outsourcing is actually relinquishing user's ultimate control over the fate of their data. As a result, the correctness of the data in the cloud was put at risk due to the following reasons. In cloud, although the infrastructures under the cloud are much more powerful and reliable than personal computing devices, they are still facing the broad range of both internal and external threats for data integrity. Examples of outages and security breaches of noteworthy cloud services appear from time to time. Second, there do exist various motivations for CSP to behave unfaithfully toward the cloud users regarding their outsourced data status. CSP might reclaim storage for monetary reasons by discarding data that have not been or were rarely accessed, or even hide data loss incidents to maintain a reputation. In short, although outsourcing data to the cloud is economically attractive for long-term large-scale storage, it does not immediately offer any guarantee on data integrity and availability. This problem, if not properly addressed, may impede the success of cloud architecture. As users no longer physically possess the storage of their data, traditional cryptographic primitives for the purpose of data security protection cannot be directly adopted. In particular, simply downloading all the data for its integrity verification is not a practical solution due to the expensiveness in I/O and transmission cost across the network. Besides, it is often insufficient to detect the data corruption only when accessing the data, as it does not give users [2] correctness assurance for those un-accessed data and might be too late to recover the data loss or damage.

3.1.1 Disadvantages:

Leak users' data to external auditor
Can extract the original data of a user during the auditing process
Existing system provide insecurity scheme for data auditing
Provide Computational overheads

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 3, March 2018

3.2 Proposed:

The system model in this project involves three parties: the cloud server, a group of users and a public verifier. There are two types of users in a group: the original user and a number of group users. The original user initially creates shared data in the cloud, and shares it with group users. Both the original user and group users are members of the group. Every member of the group is allowed to access and modify shared data. Shared data and its verification metadata (i.e. signatures) are both stored in the cloud server. A public verifier, such as a third-party auditor (TPA) providing expert data auditing services or a data user outside the group intending to utilize shared data, is able to publicly verify the integrity of shared data stored in the cloud server. When a public verifier wishes to check the integrity of shared data, it first sends an auditing challenge to the cloud server. After receiving the auditing challenge, the cloud server responds to the public verifier with an auditing proof of the possession of shared data. Then, this public verifier checks the correctness of the entire data by verifying the correctness of the auditing proof. Essentially, the process of public auditing is a challenge-and-response protocol between a public verifier and the cloud server.

Public Auditing A public verifier is able to publicly verify the integrity of shared data without retrieving the entire data from the cloud.
Correctness A public verifier is able to correctly verify shared data integrity.
Unforgeability Only a user in the group can generate valid verification metadata (i.e., signatures) on shared data.
Identity Privacy A public verifier cannot distinguish the identity of the signer on each block in shared data during the process of auditing.

With cloud computing and storage, users are able to access and to share resources offered by cloud service providers at a lower marginal cost. It is routine for users to leverage cloud storage services to share data with others in a group, as data sharing becomes a standard feature in most cloud storage offerings, including Dropbox, cloud and Google Drive. The integrity of data in cloud storage, however, is subject to skepticism and scrutiny, as data stored in the cloud can easily be lost or corrupted due to the inevitable hardware/software failures and human errors. The traditional approach for checking data correctness is to retrieve the entire data from the cloud, and then verify data integrity by checking the correctness of signatures or hash values of the entire data. Certainly, this conventional approachable to successfully check the correctness of cloud data. However, the efficiency of using this traditional approach to cloud data is in doubt. The main reason is that the size of cloud data is large in general. Downloading the entire cloud data to verify data integrity will cost or even waste user's amounts of computation and communication resources, especially when data have been corrupted in the cloud. Recently, many mechanisms have been proposed to allow not only a data owner itself but also a public verifier to efficiently perform integrity checking without downloading the entire data from the cloud, which is referred to as public auditing. In these mechanisms, data is divided into many small blocks, where each block is independently signed by the owner; and a random combination of all the blocks instead of the whole data is retrieved during integrity checking. A public verifier could be a data user (e.g. researcher) who would like to utilize the owner's data via the cloud or a third-party auditor (TPA) who can provide expert integrity checking service. In this proposed system we can implement Merkle Hash Tree to splitted the files into various parts and to provide double encryption concept to encrypt the data first at owner side and again encrypt the data based on TPA provided keys. Finally, provide batch auditing schemes to perform multiple tasks at a time and user-level privacy can be implemented to share the data without any leakages.

3.2.1 Advantages:

- Improved Public auditability and privacy-preserving
- Fully data dynamics
- Fast auditing and low-performance protocols
- End device friendliness

4. SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE DIAGRAM:

In the proposed system, we can implement public auditing scheme to monitor the data from modifying attacks. Cloud owner can be authenticated by a trusted third party. An authorized owner can be uploading the files in encrypted format. First encryption can be done using symmetric encryption algorithm. And then spilled the files into chunks.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 3, March 2018

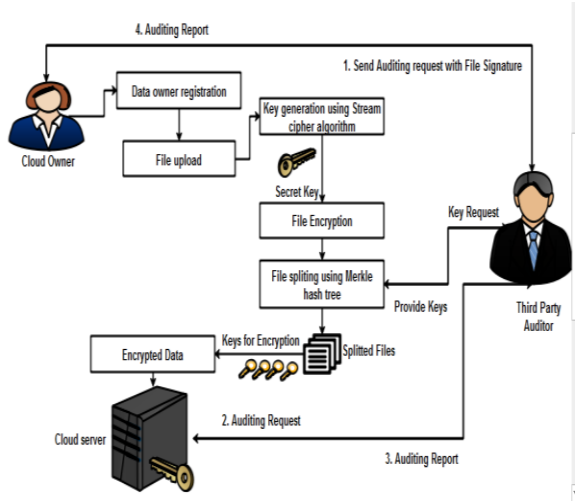


Fig 4.1.1: Overall Architecture of the System.

The following fig.5.1.1 have shown that Chunks are encrypted using Merkle hash tree algorithm. Encrypted files are uploaded to cloud storage and maintained by cloud server. The cloud owner can send the auditing messages to a cloud server through TPA. The messages can be sent is in the form of online signature. The cloud server can be audit the data and to provide proofs to the owner about the status of storage. The TPA can be performing batch auditing scheme. Finally, cloud users can access the data from a cloud with the permission of cloud owners.

4.2 MODULES DESCRIPTION

4.2.1 Cloud Registration

In the public cloud, the user has to register with his own identity using username and password. In order to avoid the unauthorized user, we have to register with the public cloud. After that, some space will be allocated to the user for storing the data.

4.2.2 Symmetric Encryption

The symmetric encryption uses a single key for both encryption and decryption. The key is only known to the user. The symmetric encryption we are going to use is stream cipher. The stream cipher model the encryption will be done bit by bit.

4.2.3 Merkle Hash Tree

Merkle Hash Tree will do a secondary encryption. The whole document is split into leaf nodes. Each leaf nodes labeled with the hash of a data block and every non-leaf node is labeled with the cryptographic hash of the labels of its child nodes.

4.2.4 Auditing

The auditing is done by the Third Party Auditor. The auditing is done on the first level encrypted document, This ensures the privacy of the user document.

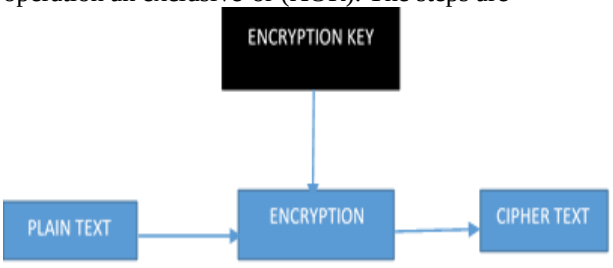
5. SYSTEM IMPLEMENTATION

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective. The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods. The implementation can be preceded by cloud storage in .NET but it will be considered as cloud communication.For dynamic cloud storage, the need is dynamic allocation. So .NET (C#) will be more suitable for platform independence and networking concepts. For maintaining route information the proposed scheme for SQL Server as database backend.

5.2 IMPLEMENTATION STAGES

5.2.1 Symmetric key algorithm:

A stream cipher is a symmetric key cipher where plaintext digits are combined with a pseudorandom cipher digit stream (key-stream). In a stream cipher, each plaintext digit is encrypted one at a time with the corresponding digit of the key-stream, to give a digit of the cipher-text stream. Since encryption of each digit is dependent on the current state of the cipher, it is also known as state cipher. In practice, a digit is typically a bit and the combining operation an exclusive-or (XOR). The steps are



5.2.2 Merkle Hash Tree (MHT)

To achieve privacy-preserving public auditing, propose to uniquely integrate the linear authenticator with binary tree technique. In our protocol, the linear combination of sampled blocks in the server's response is masked with randomness generated by the server. With random masking, the TPA no longer has all the necessary

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 3, March 2018

information to build up a correct group of linear equations and therefore cannot derive the user's data content, no matter how many linear combinations of the same set of file blocks can be collected. An MHT Encryption scheme is comprised of a tuple of algorithms (Gen, E, D, Eval), and is defined with respect to a circuit C with t inputs. Though an MHT scheme can be either a public-key or symmetric-key system, we will define it as a public-key system here. The key generation algorithm Gen takes the security parameter 1k as input and outputs the public key and private key for the system (Notation: (pk, SK) ← Gen(1k)).

Assume that messages M ∈ {0, 1} l (k).

The encryption algorithm E takes a public key and a message as input and outputs a ciphertext C, (Notation: C ← E(pk,M) for M ∈ {0, 1}l(k)).

The decryption algorithm D takes a secret key and a ciphertext, and returns a message, (Notation: M ← D(SK, C) and M ∈ {0, 1}l).

Finally, the evaluation algorithm Eval takes as input a public key, a description of at-input circuit C, and t ciphertexts C1, . . . ,Ct such that Ci← E(pk,Mi), and produces as output C*, (Notation: C* ← Eval(pk,C, C1, . . . ,Ct)).

We add a new correctness property to the standard correctness requirement for an encryption scheme as follows. We say that an encryption scheme is homomorphic with respect to a t-input circuit C if ∀k, ∀M1, . . . ,Mt, Pr[(pk, sk) ← Gen(1k); C1, . . . Ct ← E(pk,M1), . . . , E(pk,Mt); C* ← Eval(pk,C, C1, . . . ,Ct) : D(sk,C*) = C(M1, . . . ,Mt)] = 1.

Similarly, a scheme with respect to a family of circuits {Ci} if the correctness property holds for any circuit C ∈ {Ci}. Note that so far, our definition makes no requirement that the output C* of Eval should look like a standard ciphertext. Indeed, without some additional restriction on C*, every standard encryption scheme (Gen, E, D) can be trivially modified to yield a homomorphic encryption scheme (Gen', E', D', Eval') with respect to all circuits as follows.

Gen' runs as Gen.

E' runs as E.

The Eval' is constructed to take a public key, a circuit description, and up to ciphertexts, and then output the circuit description concatenated with each of the ciphertexts, as C* ← Eval"(pk, C, C1, . . . , Ct) = C|C1| . . . |Ct, with | used to denote concatenation.

On special ciphertexts, C* containing a circuit description, D' parses its input into C, C1, . . . , Ct, runs the original decryption algorithm D on the ciphertexts to obtain messages Mi ← D(SK, Ci), and runs the circuit C on these messages, to obtain D'(SK, C*) = C(M1, . . . , Mt), satisfying the homomorphic correctness property. On ciphertexts without circuit descriptions, D'(SK, C) simply returns D(SK, C).

5.2.3 Batch auditing:

With the establishment of privacy-preserving public auditing, the TPA may concurrently handle multiple auditing upon different users' delegation. The individual auditing of these tasks for the TPA can be tedious and very inefficient. Given K auditing delegations on K distinct data files from K different users, it is more advantageous for the TPA to batch these multiple tasks together and audit at one time. Keeping this natural demand in mind, we slightly modify the protocol in a single user case and achieve the aggregation of K verification equations (for K auditing tasks) into a single one. As a result, a secure batch auditing protocol for simultaneous auditing of multiple tasks is obtained.

1. Verify file tag for each user k, and quit if fail

For each user k (1≤k≤K)

2. Generate a random challenge

3. Compute μk, σk, Rkas single user case;

Chal = {(I, Vi)} i∈I

4. Compute R=R1, R2,….Rk

L = vk1||vk2||…..||vkk

5. Compute μk=rk+γkμ^' mode

6. Compute γk=h(R|(|Vk|)|L) for each user k and do batch auditing

6. RESULT ANALYSIS:

The proposed system can be compared in terms of computational time and space complexity. The proposed system provide improved accuracy rate, improved public auditing and privacy-preserving. Fully data dynamics, fast auditing and high performance and end device friendliness

7. CONCLUSION:

In this paper, we have analyzed data storage correctness issue in reference of cloud computing. We have provided the mechanism for trusted and secure data storage model with new scheme with integrity verification. The features of algorithm are useful to reduce computational cost for the client who may not have much security processing power. Using TPA we can audit the data on the server, and can preserve the privacy in data communication. The data owners have an assurity of validity of data due to the implementation of the Audit Mechanism. Thus we can

secure our data on the cloud servers using this Mechanism.

REFERENCES:

[1] Ateniese, Giuseppe, Antonio Faonio, and Seny Kamara. "Leakage-resilient identification schemes from zero-knowledge proofs of storage."

[2] Cisco visual networking index: Global mobile data traffic forecast update, 2013–2018, February 5, 2014.

[3] Hao, Zhuo, Sheng Zhong, and Nenghai Yu. "A privacy-preserving remote data integrity checking protocol with data dynamics and public verifiability." [4] Secure Hash Standard (SHS), National Institute of Standards and Technology (NIST), FIPS PUB 180

[5] Liu, Joseph K., et al. "Fine-grained two-factor access control for web-based cloud computing services

[6] Sebé, F., Domingo-Ferrer, J., Martinez-Balleste, A., Deswarte, Y., & Quisquater, J. J. (2008). Efficient remote data possession checking in critical information infrastructures. [7] Wang, Q., Wang, C., Li, J., Ren, K., & Lou, W. (2009, September). Enabling public verifiability and data dynamics for storage security in cloud computing.

[8] PauloS.L.M. Barreto and Michael Naehrig. Pairing-friendly elliptic-curves of prime order. In Bart Preneel and Stafford Tavares, editors, Selected Areas in Cryptography, volume 3897 of Lecture Notes in Computer Science, pages 319–331. Springer Berlin Heidelberg,2006.

[9] A. Barsoum and A. Hasan. Enabling dynamic data and indirectmutual trust for cloud computing storage systems. Parallel andDistributed Systems, IEEE Transactions on, 24(12):2375–2385, Dec2013.

[10] Matt Blaze, GerritBleumer, and Martin Strauss. Divertible protocols and atomic proxy cryptography. In Kaisa Nyberg, editor, Advances in Cryptology - EUROCRYPT'98, volume 1403 of LNCS,pages 127–144. Springer Berlin / Heidelberg, 1998.

[11] Dan Boneh and Xavier Boyen. Efficient selective identity-based encryption without random oracles. Journal of Cryptology (JoC), 24(4):659–693, 2011. early version in Eurocrypt 2004.

[12] Dan Boneh, Ben Lynn, and HovavShacham. Short signatures from the Weil pairing. In Colin Boyd, editor, Advances in Cryptology– ASIACRYPT 2001, volume 2248 of Lecture Notes in Computer Science, pages 514–532. Springer Berlin Heidelberg, 2001.

[13] Towards Secure Data Distribution Systems inMobile Cloud ComputingJiang Zhang, Zhenfeng Zhang, and Hui Guo

[14]A Privacy-Preserving Remote Data Integrity Checking Protocol with Data Dynamics and Public Verifiability Author: Z. Hao, S. Zhong

[15] Enabling Personalized Search over Encrypted Outsourced Data with Efficiency Improvement Author: Z. Fu, K. Ren

[16]Leakage-Resilient Identification Schemes from Zero-Knowledge Proofs of Storage Author: G. Ateniese, A. Faonio, and S. Kamara