ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

Dynamic Big Data Storage on Cloud with Efficient Verifiable Fine-grained Updates using Secure Erasure Code-Based Cloud Storage System

Suganya .S[*1], Mrs.Sumathi .D[2]
[*1] ME , [2] Assistant Professor Computer Science and Engineering, Jayam College of Engineering and Technology,Dharmapuri, Tamilnadu, India
suganyathendral@gmail.com

Abstract

Cloud computing opens a new era in IT as it can provide various elastic and scalable IT services in a pay-as-you-go fashion, where the users can reduce the huge capital investments in their own IT infrastructure. The users of cloud storage services have no control over their data, which makes data security, one of the major concerns of using cloud. The existing system allows data integrity to be verified without possession of the actual data file. The verification done by a trusted third party is called data auditing, and the third party is called an auditor. The main drawback in auditing is the any person can challenge with the service provider for integrity of data. Based on this scheme, an enhancement can dramatically reduce communication overheads for verifying small updates. This scheme can offer not only enhanced security and flexibility, but also significantly lower overhead for big data applications with a large number of frequent small updates.

Keywords: Cloud Computing, Big Data, Data Security, Authorized Auditing, Decentralized Erasure code, Proxy Re-encryption, Secure storage System.

Introduction

CLOUD computing is being intensively referred to as one of the most influential innovations in information technology in recent years. With resource virtualization, cloud can deliver computing resources and services in a pay-as-you-go mode, which is envisioned to become as convenient to use similar to daily-life utilities such as electricity, gas, water and telephone in the near future. These computing services can be categorized into Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service SaaS). Many international IT corporations now offer powerful public cloud services to users on a scale from individual to enterprise all over the world; examples are Amazon AWS, Microsoft Azure, and IBM Smart Cloud. Although current development and proliferation of cloud computing is rapid, debates and hesitations on the usage of cloud still exist. Data security/privacy is one of the major concerns in the adoption of cloud computing. Compared to conventional systems, users will lose their direct control over their data. We investigate how to improve the efficiency in verifying frequent small updates which exist in many popular cloud and big data contexts such as social media. Compared to existing schemes, both theoretical analysis and experimental results demonstrate that our modified scheme can significantly lower communication overheads. The proposed system uses a threshold proxy re-encryption scheme and integrates it with a decentralized erasure code such that a secure distributed storage system is formulated. Using the threshold proxy re-encryption scheme, a secure cloud storage system provides secure data storage and secure data forwarding functionality in a decentralized structure. Data Storing in a third party's cloud system causes serious problems on data confidentiality. So to provide strong confidentiality for messages in storage servers, a user has to encrypt messages by a cryptographic method before applying an erasure code method in order to encode and store messages. When the user needs to use a message, the user needs to retrieve the codeword symbols from storage servers. After getting the codeword from the server the user decode them, and then decrypt them by using cryptographic keys. The encryption scheme supports encoding and forwarding operations over encrypted and encoded messages. The message can be retrieved as long as one storage server survives. The way of encoding a message of k symbols is by erasure code. In order to store a message, the codeword symbols are stored at different storage server. A storage server failure will lead to an error in the codeword symbol. The message can be recovered from the codeword symbols by the decoding process until the number of failure servers is under the tolerance threshold of the erasure code.

In the existing system, the storage system is distributed and has no central authority, in this case constructing a secure storage system that supports multiple functions is challenging.

Existing System

In the early years, the Network-Attached Storage (NAS) and the Network File System (NFS) provide storage devices over the network so that user can access the storage devices through network connection.

Drawbacks of an existing system

Encryption schemes supports confidentiality of the data, but the functionality of the storage system is limited because only certain operations are supported.

- Data robustness is the important need for storage systems.
- Participating parties in the auditing scheme
- Rank-based Merkle hash tree

Proposed system

The proposed system focus on constructing a cloud storage system for confidentiality and functionality.

A cloud storage system is a large scale distributed storage system that has large number of many independent storage servers.

Methods in proposed system:

- Secured erasure code
- Proxy Re-encryption server
- Fine grained Access (Improved)
- Data Validation
- Secured threshold proxy re-encryption server
- One time Data access

Related Work

Constructing a Secure Server

Decentralized erasure code formulates a secure distributed storage system by integrating with a secure threshold proxy re-encryption server for processing big data. This method allows multiple users interact with the storage system and upload their data in to the distributed storage system. This system supports secure and robust data storage and retrieval and also lets a user forward the data in the storage servers to another user without retrieving the data. This causes the ownership data unused and so the data is secured at the time of retrieval. The database content is in decrypted format. So that the hackers cannot able to access the big data even if they access the database. Because the encrypted data will become unused even the data is obtained by the hacker . So the system becomes so stronger. The project fully deals with encryption, encoding, and forwarding. The application will be shown in both cloud servers as well as in local host as per the available environment. The storage is very flexible with the users and the user can authorize the sender request for generating the key. The key is a onetime key, using this one time key the sender can access the encrypted file in decrypted format only once. The key will be invalid after one use. This is method is applied for secured data forwarding. when the data is forwarded a proxy server will be created virtually in order to view the encrypted data from the sender side. We use a threshold proxy re-encryption scheme with multiplicative homomorphism property. An encryption scheme is multiplicative homomorphic if it supports a group operation where E is the encryption function, D is the decryption function and (PK, SK) is a pair of public key and secret key. This encryption scheme provides the encoding operation for encrypted messages. This proxy re-encryption scheme with homomorphic property is converted into a threshold version. A secret key is sent to key servers that has a threshold value t. To decrypt a message of k symbols, each of the key servers queries two storage servers and partially decrypts the encrypted codeword symbols. Codeword symbols are obtained from the partially decrypted cipher texts from the t key servers.

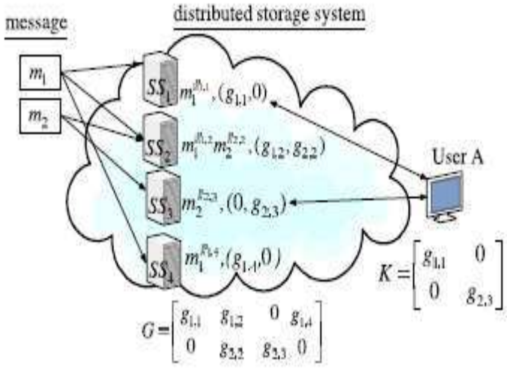
message
distributed storage system
m_1
m_2
SS_1 $m_1^{g_{1,1}}, (g_{1,1}, 0)$
SS_2 $m_1^{g_{1,2}} m_2^{g_{2,2}}, (g_{1,2}, g_{2,2})$
SS_3 $m_2^{g_{2,3}}, (0, g_{2,3})$
SS_4 $m_1^{g_{1,4}}, (g_{1,4}, 0)$
User A
$K = \begin{bmatrix} g_{1,1} & 0 \\ 0 & g_{2,3} \end{bmatrix}$
$G = \begin{bmatrix} g_{1,1} & g_{1,2} & 0 & g_{1,4} \\ 0 & g_{2,2} & g_{2,3} & 0 \end{bmatrix}$

Fig.1. A Storage Systems with Random Linear Code

Proxy server

The main advantages of using Proxy server are

- Time to access the resources is increased. Proxies are normally used to cache the web pages from a web server.
- Prevent downloading the redundant content(and save bandwidth).
- Log / audit usage, e.g. to provide company employee Internet usage reporting.
- Before delivering the content the cache the scan transmitted content for malware.
- Scan outbound content, e.g., for data loss prevention.

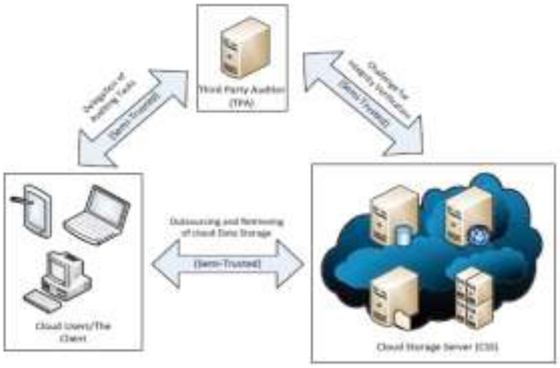
Delegation of Auditing Tasks
(Semi-Trusted)
Third Party Auditor (TPA)
Challenge for Integrity Verification
(Semi-Trusted)
Outsourcing and Retrieving of cloud Data Storage
(Semi-Trusted)
Cloud Users/The Client
Cloud Storage Server (CSS)

Fig.2. Relationship between Parties in Public Auditing Scheme

Distributed Erasure Code

Consider the problem of constructing an erasure code for storage over a network when the data sources are distributed in the cloud server. Assume that there are n storage nodes with limited memory and k < n sources generating the data. A data collector can appear anywhere in the network for accessing the data, used to query any k storage nodes and be able to retrieve the data. A Decentralized Erasure Codes is introduced which linear codes with a specific randomized structure inspired by network are coding on random bipartite graphs with encrypted format. Communication, storage and computation cost is reduced by Decentralized erasure codes .

Erasure code over a cloud Network

Decentralized erasure codes are random linear codes over a finite field Fq with a specific randomized structure on their generator matrix. Each data packet Di is considered as a vector of elements of a finite field fi. We denote the set of data nodes by V1 with |V1| = k and storage nodes by V2, |V2| = n. We will now give a description of a randomized construction of a bipartite graph that corresponds to the creation of a decentralized erasure code. Every data node i ∈ V1 is assigned a random set of storage nodes N (i). This set is created as follows: a storage node is selected uniformly and independently from V2 and added in N(i) and this procedure is repeated d(k) times. Therefore N (i) will be smaller than d(k) if the same storage node is selected twice. In fact, the size of the set N (i) is exactly the number of coupons a coupon collector would have after purchasing d(k) coupons from a set of n coupons . It is not hard to see that when d (k) ≪ n, N(i) will be approximately equal to d(k) with high probability. Denote N (j) = {i ∈ V1 : j ∈ N(i)} the set of data nodes that connect to a storage node.

Sj = X ∀i:∈N(j) fij Di where the coefficients fij are selected uniformly and independently. The storage node also stores the fij coefficients, which requires an overhead storage of N(j)(log2 (q) + log2 (k)) bits. This can be generalized into s = mG where s is a 1 × n vector of stored data, m is 1 × k data vector and G is a k × n matrix with non-zero entries corresponding to the adjacency matrix of the random bipartite graph we described. The main property which allows the decentralized construction of the code is the data node in choosing its neighbours independently and uniformly and has N(i) = O(d(k)) nonzero elements. This is called decentralized property. The results are compared with random linear coding for distributed networked storage. To reconstruct, the data collector must invert a k × k sub matrix G′ of G. Any selection of G′ forms a full rank matrix with high probability is the key property for successful decoding . Clearly d(k) is measuring the sparsely of G. Making d(k) as small as possible is very important since it is directly related with overhead storage, decoding complexity and communication cost. Our main contribution is identifying how small d(k) can be made for matrices that have the decentralized property to ensure that their sub matrices have full rank. The following theorems are the main results of this correspondence:

Theorem 1

Let G be a random matrix with independent rows constructed as described. Then, d(k) = c ln (k) is sufficient for a random k ×k sub matrix G′ of G to be non-singular with high probability. More specifically, P r[det(G′) = 0] ≤ k q + o(1) for any c > 5 n k.

Theorem 2

(Converse) If each row of G is generated independently (Decentralized property), at least d(k) = Ω(ln(k)) is necessary to have G′ invertible with high probability. From the two theorems it follows that d(k) = c ln(k). Therefore, decentralized erasure

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

codes have minimal data node degree and logarithmically many nonzero elements in every row. Decentralized erasure codes can be decoded using Maximum Likelihood (ML) decoding, which corresponds to solving a linear system of k equations in GF(q). This has a decoding complexity of O(k 3).
Note: However that one can use the sparsely of the linear equations and have faster decoding. Using the Wiedemann algorithm one can decode in O(k 2 log(k)) time on average with negligible extra memory requirements.

System Architecture

The system consist of following phases

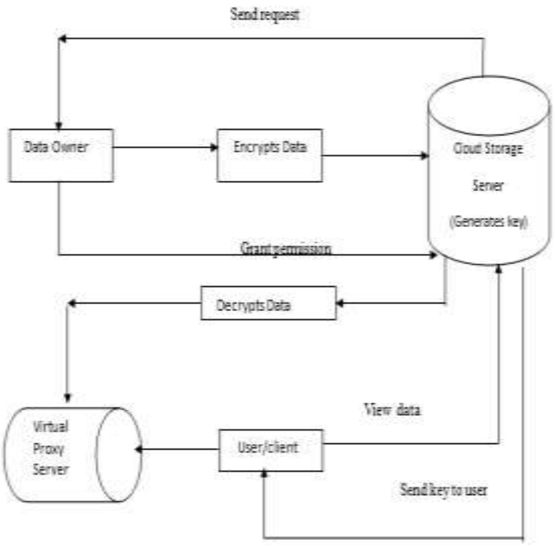

Fig.3.Construction of Proxy Server

Data Owner key Generation

This is the initial module of this project. Data owner is the possession of and responsibility for information. Data Owner has the power as well as control over the data. The control of information is not just the ability to access, create, modify, sell or remove data but has the act of legal rights and complete control over a single piece or set of data elements. The Data owner defines and provides information about the rightful owner of data assets and the distribution policy implemented by the data owner. Here Data owners can create a login and they can upload their own files in the cloud Storage.

Construction of Secure Cloud Storage

Here the construction of the cloud will be more secure. The two biggest concerns about cloud storage are reliability and security. Basically, a cloud storage system is considered as a network of distributed data centres which typically uses cloud computing technologies offers interface for storing data and increase the availability of the data, which is stored at different locations. This is not visible to the user because the data will be in the encrypted format. Many cloud storage providers offers various kinds of services to their customers. But here a personalized cloud store server for both big data and secured erasure code is used. The storage server will be unique which has been distributed into much system for easy access of data. It contains only the encrypted data of the data owners

Proxy Re-Encryption

It is one of the advanced encryption model which works on both real system and virtual systems. This works more efficient on cloud systems. Proxy re-encryption schemes are cryptosystems which allow third-parties (proxies) to alter a cipher text which has been encrypted for one party. Proxy re-encryption schemes are similar to traditional symmetric or asymmetric encryption schemes. It allows a message recipient (key holder) to generate a re-encryption key based on his secret key. The re-encryption key is used and is executed by the proxy to translate cipher texts to the delegated user's key. Asymmetric proxy re-encryption schemes will be in bi-directional and unidirectional .Proxy re-encryption schemes allows a cipher text to re-encrypted an unlimited number of times.

Double Encryption Technique

The proposed system works based on the following algorithm

- Encryption type = 64 bit key
- Key Length = 23 char
- Key type = Alpha numerical with special characters.
- Key character = Encryption and decryption
- Key Limitation = Caps alphabets = 26, Small alphabets = 26, 0 -9 numbers = 10, Special Characters = 10:
- Result = **3.8483294074100064e+135** of key combinations can be produced.
- It is equivalent to 30000 trillion and above combination

Conclusion

Thus we are concluding that all the result obtained according to the committed abstract. In this paper, we consider a cloud storage system consists of storage servers and key servers. We integrate a newly proposed threshold proxy re-encryption scheme and

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

erasure codes over exponents. The threshold proxy re-encryption scheme supports encoding, forwarding, and partial decryption operations in a distributed way. To decrypt a message of k blocks that are encrypted and encoded ton code word symbols, each key server only has to partially decrypt two codeword symbols in our system. By using the threshold proxy re-encryption scheme, we present a secure cloud storage system that provides secure data storage and secure data forwarding functionality in a decentralized structure. Moreover, each storage server independently performs encoding and re-encryption and each key server independently perform partial decryption. Our storage system and some newly proposed content addressable file systems and storage systems are highly compatible. Our storage servers act as storage nodes in a content addressable storage system for storing content addressable blocks. Our key servers act as access nodes for providing a front-end layer such as a traditional file system interface. Further study on detailed cooperation is required.

References

[1] G. Ateniese, K. Benson, and S. Hohenberger, "Key-Private Proxy Re-Encryption, "Proc. Topics in Cryptology (CT-RSA),pp. 279-294, 2009.

[2] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage,"ACM Trans. Information and System Security, vol. 9, no. 1, pp. 1-30, 2006.

[3] M. Blaze, G. Bleumer, and M. Strauss, "Divertible Protocols and Atomic Proxy Cryptography,"Proc. Int'l Conf. Theory and Applica-tion of Cryptographic Techniques (EUROCRYPT),pp. 127-144, 1998.

[4] A.G. Dimakis, V. Prabhakaran, and K. Ramchandran, "Ubiqui-tous Access to Distributed Data in Large-Scale Sensor Net-works through Decentralized Erasure Codes,"Proc. Fourth Int'l Symp. Information Processing in Sensor Networks (IPSN),pp. 111-117, 2005.

[5] A.G. Dimakis, V. Prabhakaran, and K. Ramchandran, "Decen-tralized Erasure Codes for Distributed Networked Storage,"IEEE Trans. Information Theory,vol. 52, no. 6 pp. 2809-2816, June 2006.