Secure Data Forwarding In Cloud Storage System With Cryptographic Scheme

1.S KARTHIKA,
ASSISTANT PROFESSOR
Department of Computer science and Engineering,
East Point College Of Engineering and Technology, Karnataka, Bengaluru,India
karthu_suresh@yahoo.com

2.M.BHUVANESHWARI,
ASSISTANT PROFESSOR
Department of Computer Science And Business Systems,
M.Kumarasamy College of Engineering , Tamilnadu, Karur, India
bhuvanamanoharan03@gmail.com

ABSTRACT

Storing data in a third party's cloud system causes serious concern on data confidentiality. In order to provide strong confidentiality for messages in storage servers, a user can encrypt messages by a cryptographic method before applying an code method to encode and store messages.

The threshold proxy re encryption scheme supports forwarding, and perfect decryption operations in a distributed way. The user forward his data in the storage servers to another user without retrieving the data back. The proxy re-encryption scheme supports encrypted messages as well as forwarding operations over encoded messages. This method fully integrates encrypting and forwarding. The tight integration of encryption, and forwarding makes the storage system efficiently meet the requirements of data robustness, data confidentiality, and data forwarding.

Accomplishing the integration with consideration of a distributed structure is challenging. This system meets the requirements that storage servers independently perform re-encryption and key servers independently perform perfect decryption. This setting allows more flexible adjustment between the number of storage servers and robustness. By using the cryptographic scheme data confidentiality of the user's information is achieved. It also allows various operations over the encrypted data. In future revocation can be performed by making the entire application web based.

Index Terms:- Proxy re-encryption, Revocation,Robustness,Confidentiality,Encryption, Data Forwarding,Decryption.

1 INTRODUCTION

Cloud computing is a concept that treats the resources on the Internet as a unified entity, a cloud. In this paper, the focus is on designing a cloud storage system for robustness, confidentiality, and functionality. A cloud storage system is considered as a large-scale distributed storage system that consists of many independent storage servers.A decentralized erasure code is suitable for use in a distributed storage system.In order to provide strong confidentiality for messages in storage servers, a user can encrypt messages by a cryptographic method before applying an erasure code method to encode and store messages. When he wants to use a message, he needs to retrieve the codeword symbols from storage servers, decode them, and then decrypt them by using cryptographic keys. There are three problems in the above straightforward integration of encryption and encoding.

First, the user has to do most computation and the communication traffic between the user and storage servers is high. Second, the user has to manage his cryptographic keys. If the user's device of storing the keys is lost or compromised, the security is broken. Finally, besides data storing and retrieving, it is hard for storage servers to directly support other functions. For example, storage servers cannot directly forward a user's messages to another one. The owner of messages has to retrieve, decode, decrypt and then forward them to another user.

A new threshold proxy re-encryption scheme is proposed and integrated with a secure decentralized code to form a secure distributed storage system. The encryption scheme supports encoding operations over encrypted messages and forwarding operations over encrypted and encoded messages. The tight integration of encoding, encryption, and forwarding makes the storage system efficiently meet the requirements of data robustness, data confidentiality, and data forwarding.

2 LITERATURE REVIEW

In the previous work proposed by Wenjing Lou, an effective and flexible distributed scheme with explicit dynamic data support, including block update, delete, and append is performed. Curtmola et al. aimed to ensure data possession of multiple replicas across the distributed storage system. They extended the PDP scheme to cover multiple replicas without encoding each replica separately, providing guarantee that multiple copies of data are actually maintained. Juels et al. described a formal "proof of retrievability" (POR) model for ensuring the remote data integrity.

Their scheme combines spot-cheking and error-correcting code to ensure both possession and retrievability of files on archive service systems. Shacham et al.built on this model and constructed a random linear function based homomorphic authenticator which enables unlimited number of queries and requires less communication overhead. Ateniese et al defined the "provable data possession" (PDP) model for ensuring possession of file on untrusted storages.

Their scheme utilized public key based homomorphic tags for auditing the data file, thus providing public verifia-bility. However, their scheme requires sufficient computation overhead that can be expensive for an entire file. In their subsequent work, Ateniese et al described a PDP scheme that uses only symmetric key cryptography.

This method has lower-overhead than their previous scheme and allows for block updates, deletions and appends to the stored file, which has also been supported in our work. However, their scheme focuses on single server scenario and does not address small data corruptions, leaving both the distributed scenario and data error recovery issue unexplored.

3 PROPOSED SYSTEM

3.1 Proxy re-encryption

proxy re-encryption scheme is proposed and integrated with a decentralized erasure code such that a secure distributed storage system is formulated.

The distributed storage system not only supports secure and robust data storage and retrieval, but also lets a user forward his data in the storage servers without retrieving the data back.

The main technical contribution is that the proxy re-encryption scheme supports encoding operations over encrypted messages. The threshold proxy reencryption scheme supports encoding, forwarding, and partial decryption operations in a distributed way. To decrypt a message of k blocks that are encrypted and encoded to n codeword symbols.

The storage system and some newly proposed content addressable file systems and storage system are highly compatible. The storage servers act as storage nodes in a content addressable storage system for storing content addressable blocks.

The key servers act as access nodes for providing a front-end layer such as a traditional file system interface.

3.2 Algorithm

$HMAC\,(K,m) = H\,((K \oplus opad) \parallel H\,((K \oplus ipad) \parallel m))$

where

H is a cryptographic hash function,

K is a secret key padded to the right with extra zeros to the input block size of the hash function, or the hash of the original key if it's longer than that block size,

m is the message to be authenticated,

$\parallel$ denotes concatenation,

$\oplus$ denotes exclusive or (XOR),

opad is the outer padding (0x5c5c5c…5c5c, one-block-long hexadecimal constant),

and *ipad* is the inner padding (0x363636…3636, one-block-long hexadecimal constant).

4 Proposed Methodology

4.1 MAC Based Solution

In cryptography, a **hash-based message authentication code** (**HMAC**) is a specific construction for calculating a message authentication code (MAC) involving a cryptographic hash function in combination with a secret cryptographic key. As with any MAC, it may be used to simultaneously verify both the *data integrity* and the *authenticity* of a message. Any cryptographic hash function, such as MD5 or SHA-1, may be used in the calculation of an HMAC; the resulting MAC algorithm is termed HMAC-MD5 or HMAC-SHA1 accordingly.

The cryptographic strength of the HMAC depends upon the cryptographic strength of the underlying hash function, the size of its hash output length in bits, and on the size and quality of the key.

An iterative hash function breaks up a message into blocks of a fixed size and iterates over them with a compression function.MAC-based solution which suffers from undesirable systematic demerits bounded usage and stateful verification, which may pose additional online burden to users, in a public auditing setting.

There are two possible ways to make use of MAC to authenticate the data. A trivial way is just uploading the data blocks with their MACs to the server. Later, the **USER** can randomly retrieve blocks with their MACs and check the correctness.

4.2 IP- Based Solution

Cloud Server will add the Restricted IP Address (i.e. Server address). When cloud server is creating the file (locker) for the registered user one more added information is allowed Internet Protocol Address.

The user will be able to access the file only if he/she is entered in the mentioned allowed ip address. If the user entered through the allowed ip address then it will ask for MAC Key. If the MAC key is valid then it views the data to the user and the he can perform the dynamic operation.

If the MAC key is not valid then the data will be encrypted by (Proxy –RE Encryption) and displayed to the user at the same time the user will be blocked.If the User is not entered in the given the ip address then it will ask for Password maximum time.

Then all these details (File Name, User Name, Date, and Password, from which IP address is trying to access) will be stored as HACKER.Then the same result will be displayed in the MOBILE Panel (J2ME Wireless Tool Kit).

5 Advantage

Proxy Re-Encryption (PRE) is a cryptographic primitive in which a semi-trusted proxy is able to convert a cipher text encrypted into another cipher text that can be opened without seeing the underlying plaintext.A secure cloud storage system implies that an unauthorized user or server cannot get the content of stored messages**.**

The Unauthorized user will be blocked.Achieves the integration of storage correction insurance and data error localization. Can also guarantee the simultaneous identification of misbehaving servers.

6 Conclusion

The storage system and some newly proposed content addressable file systems and storage system are highly compatible. The storage servers act as storage nodes in a content addressable storage system for storing content addressable blocks.

The key servers act as access nodes for providing a front-end layer such as a traditional file system interface.

Cloud Server will add the Restricted IP Address (i.e. Server address). When cloud server is creating the file (locker) for the registered user one more added information is allowed Internet Protocol Address.

The user will be able to access the file only if he/she is entered in the mentioned allowed ip address. If the user entered through the allowed ip address then

it will ask for MAC Key. If the MAC key is valid then it views the data to the user and the he can perform the dynamic operation.If the MAC key is not valid then the data will be encrypted by (Proxy –RE Encryption) and displayed to the user at the same time the user will be blocked.

If the User is not entered in the given the ip address then it will ask for Password maximum time.Then all these details (File Name, User Name, Date, and Password, from which IP address is trying to access) will be stored as HACKER.

Then the same result will be displayed in the MOBILE Panel (J2ME Wireless Tool Kit). In future revocation can be performed by making the entire application web based.

REFERENCES

[1]W.Dong,F.Douglis,K.Li,H.Patterson,S.Reddy,and P.Shilane,(2011) 'Tradeoffs in Scalabel Data Routing for Deduplication Clusters',Proc.Ninth USENIXConf.File and Storage Technologies(FAST).

[2]C.Ungureanu,B.Atkin,A.Aranya,S.Gokhale,S.Rago,G.Calkowski,C.Dubnicki,and A.Bohra, (2010) 'Hydrafts: A High-Throughput File System for the Hydrastor Content-Addressable Storage System',Proc.Eighth USENIX Conf.File and Storage Technologies(FAST),p.17.

[3]C.Wang,Q.Wang,K.Ren,and W.Lou, (2010) 'Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing',Proc.IEEE 29th Int'lConf.ComputerComm.(INFOCOM) ,pp.525-533.

[4]C. Dubnicki, L. Gryz, L. Heldt, M. Kaczmarczyk, W. Kilian, P.Strzelczak, J. Szczepkowski, C. Ungureanu, and M. Welnicki, (2009) 'Hydrastor: A Scalable Secondary Storage, ' Proc. Seventh Conf. File and Storage Technologies (FAST), pp. 197-210.

[5]K.D. Bowers, A. Juels, and A. Oprea,(2009) 'HAIL: A High-Availability and Integrity Layer for Cloud Storage', Proc. 16th ACM Conf. Computer and Comm. Security (CCS), pp. 187-198.

[6]G. Ateniese, S. Kamara, and J. Katz, (2009) 'Proofs of Storage from Homomorphic Identification Protocols, ' Proc. 15th Int'l Conf. Theory and Application of Cryptology and Information Security (ASIACRYPT), pp. 319-333.