Privacy Preservation Guidelines based Substance Distribution in Public Clouds

Mrs.K.Valarmathi[1] N.Ramya[2]
[1]Head of Dept. (MCA, M. Phil., B. Ed)
[1,2]Department of Computer Science
[1,2]Trinity College for Women, Namakkal

Abstract— Present approach to privacy preservation guidelines based data hosted in the public cloud are based on substance distribution of the data. Below such approaches, data owners are in accuse of encrypting the data before uploading them on the cloud and encrypting the data at any time user authorization alerts. Data owners thus increase the communication and estimation costs. An approach must be allot the enforcement of privacy preservation guidelines based substance distribution access control to the cloud, so to decrease the network traffic on public cloud, although assuring data confidentiality from the cloud. We propose an approach; the data owner performs a guideline based encryption; whereas the cloud performs guidelines based encryption on top of the owner encrypted data. While aspects of these characteristics have been realized to a certain extent, cloud environment remainder a task in progress. This publication provides an overview of the privacy challenges pertinent to public cloud computing and points out consideration organizations should take when private source information, application, and communications to a public cloud environment.

Key words: Cloud Computing, Privacy, Information Technology, Private Sourcing

I. INTRODUCTION

Cloud computing have quickly grown-up in recent years outstanding to the compensation of greater flexibility and accessibility of sources the cost cheap. Security is be disturbed for agency delightful into thought migrate application to public cloud background location, and form the strength behind this document.

Cloud computing is the make use of of computing resources that are deliver as a service over a network. For example, the email service is probably the most popular one. Cloud computing is a concept that treats the wealth on the Internet as an included entity, a cloud. Users just use services without being disturbed about how computation is done and storage is managed. This paper focuses 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. Data robustness is a major necessity for storage systems. One way to provide data robustness is to replicate a message such that each storage server stores a copy of the message. It is very robust because the message can be retrieved as long as one storage server survives. Another way is to encode a message of k symbols into a codeword of n symbols by erasure coding. To store a message, each of its codeword symbols is stored in a different storage server. A storage server failure corresponds to an erasure error of the codeword symbol. As long as the number of failure servers is under the tolerance threshold of the erasure code, the message can be recovered from the codeword symbols stored in the Available storage servers by the decoding process. A decentralized erasure code is an erasure code that independently computes each codeword symbol for a message. A decentralized erasure code is suitable for use in a distributed storage system. After the message symbols are sent to storage servers, each storage server independently computes a codeword symbol for the received message symbols and stores it. This finishes the encoding and storing process. Storing data in a third party's cloud system causes serious concern on data confidentiality.

To provide Strong confidentiality for messages in storage servers, a user encrypts 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. This addresses the problem of forwarding data to another user by storage servers directly under the command of the data owner. We consider the system model that consists of distributed storage servers and key servers. Since storing cryptographic keys in a single device is risky, a user distributes his cryptographic key to key servers that shall perform cryptographic function on behalf of the user. These key servers are highly protected by security mechanisms. We use a new threshold Byte Rotation Encryption scheme and integrate it 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.

Our system meets the requirements that storage servers independently perform encoding and re-encryption. There are n distributed storage servers and m key servers in the cloud storage system. A message is divided into k blocks and represented as a vector of k symbols.

The contributions are:

1) We construct a secure cloud storage system that supports the function of secure data forwarding by using a threshold Byte Rotation Encryption scheme. The encryption scheme supports decentralized erasure codes over encrypted messages and forwarding operations over encrypted and encoded messages. Our

system is highly distributed where storage servers independently encode and forward messages and key servers independently perform partial decryption.

2) The storage size in each storage server does not increase because each storage server stores an encoded result a codeword symbol which is a combination of encrypted message symbols.

A. Purpose and Scope

The function of this article is to give an overview of public cloud and the Security and privacy face concerned. The article discusses the threats, technology risks, and safeguard for public cloud environments, and provides the coming needed to make well-versed information technology decisions on their dealing.

II. RELATED WORKS

A cloud storage system, consisting of a collection of storage servers, provides long-term storage services over the Internet. Storing data in a third party's cloud system causes serious concern over data confidentiality. General encryption schemes protect data confidentiality, but also limit the functionality of the storage system because a few operations are supported over encrypted data. Constructing a secure storage system that supports multiple functions is challenging when the storage system is distributed and has no central authority. We propose a threshold proxy re-encryption scheme and integrate it 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 to another user without retrieving the data back. The main technical contribution is that the proxy re-encryption scheme supports encoding operations over encrypted messages as well as forwarding operations over encoded and encrypted messages. Our method fully integrates encrypting, encoding, and forwarding. We analyze and suggest suitable parameters for the number of copies of a message dispatched to storage servers and the number of storage servers queried by a key server. These parameters allow more flexible adjustment between the number of storage servers and robustness. Designing a cloud storage system for robustness, confidentiality and functionality. The proxy re-encryption scheme supports encoding operations over encrypted messages as well as forwarding operations over encoded and encrypted messages. To provide data robustness is to replicate a message such that each Storage server stores a copy of the message. It is very robust because the message can be retrieved as long as one storage server survives. The number of failure servers is under the tolerance threshold of the erasure code, the message can be recovered from the codeword symbols stored in the available Storage servers by the decoding process. This provides a tradeoff between the storage size and the tolerance threshold of failure servers.

A decentralized erasure code is an erasure code that independently computes each codeword symbol for a message. A decentralized erasure code is suitable for use in a distributed storage system. A storage server failure is modeled as an erasure error of the stored codeword symbol. We construct a secure cloud storage system that supports the function of secure Data forwarding by using a threshold proxy re-encryption scheme. The encryption scheme supports decentralized erasure codes over encrypted messages and forwarding operations over encrypted and encoded messages. Our system is highly distributed where storage servers independently encode and forward messages and key servers independently perform partial decryption.

We consider the system model that consists of distributed storage servers and key servers. Since storing cryptographic keys in a single device is risky, a user distributes his cryptographic key to key servers that shall perform cryptographic functions on behalf of the user. These key servers are highly protected by security mechanisms. The distributed systems require independent servers to perform all operations. We propose a new threshold proxy re-encryption scheme and integrate it 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.

A. Distributed Storage Systems

The Network-Attached Storage (NAS) and the Network File System (NFS) provide storage devices over the network such that a user can access the storage devices via network connection. A decentralized architecture for storage systems offers good scalability, because a storage server can join or leave without control of a central authority. To provide robustness against server failures, a simple method is to make replicas of each message and store them in different servers. One way to reduce the expansion rate is to use erasure codes to encode messages. A message is encoded as a codeword, which is a vector of symbols, and each storage server stores a codeword symbol. A storage server failure is modeled as an erasure error of the stored codeword Symbol. To store a message of k blocks, each storage server linearly combines the blocks with randomly chosen coefficients and stores the codeword symbol and coefficients. To retrieve the message, a user queries k storage servers for the stored codeword symbols and coefficients and solves the linear system. The system has light data confidentiality because an attacker can compromise k storage servers to get the message.

B. Proxy Re-Encryption Scheme

In a proxy re-encryption scheme, a proxy server can transfer a cipher text under a private key A to a new one under another public key B. The server does not know the plaintext during transformation. The data is first encrypted with a symmetric data encryption key and then stored in the cloud storage server. The cloud storage server uses a re-encryption algorithm to transfer the encrypted DEK into the format that can be decrypted by the recipient's private key. The recipient then can download the encrypted data from the cloud and use the DEK for decryption. A re-encryption key is generated from the data owner's private key and a recipient's public key. A data owner may share different files with different recipient groups. Therefore, a recipient cannot read data for a group it does not belong to. The cloud, on the other hand, acts as an intermediate proxy. It cannot read the data as it cannot get DEKs. Thus the system has data confidentiality and supports the data forwarding function.

This technique sketches the design of PAST, a large-scale, Internet-based, global storage utility that provides scalability, high availability, persistence and security. PAST is a peer-to-peer Internet application and is entirely self organizing. PAST nodes serve as access points for clients, participate in the routing of client requests, and contribute storage to the system. Nodes are not trusted, they may join the system at any time and may silently leave the system Without warning. Yet, the system is able to provide strong assurances, efficient storage access, load balancing and scalability. Among the most interesting aspects of PAST's design are(1) the Pastry location and routing scheme, which reliably and efficiently routes client requests among the PAST nodes, has good network locality properties and automatically resolves node failures and node additions; (2) the use of randomization to ensure diversity in the set of nodes that store a file's replicas and to provide load balancing; and (3) the optional use of smartcards, which are held by each PAST user and issued by a third party called a broker. The smartcards support a quota system that balances supply and demand of storage in the system.There are currently many projects aimed at constructing peer-to-peer applications and understanding more of the issues and requirements of such applications and systems. Peer-to-peer systems can be characterized as distributed systems in which all nodes have identical capabilities and responsibilities and all communication is symmetric. We are developing PAST, an Internet-based, peer-to-peer global storage utility, which aims to provide strong persistence, high availability, scalability and security. The PAST system is composed of nodes connected to the Internet, where each node is capable of initiating and routing client requests to insert or retrieve files. Optionally, nodes may also contribute storage to the system. The PAST nodes form a self-organizing overlay network. Inserted files are replicated on multiple nodes to ensure persistence and availability.

III. PRIVACY PRESERVATION GUIDELINES BASED SUBSTANCE DISTRIBUTION

These substance distribution and guidelines privacy system have more support for enhancement. This can obtain to be to a huge extent expensive and places and every one client that has accepted right of entry can then get the twisted information from the Cloud and unscramble the information utilizing the supplied key. This guarantee that no unapproved client gets right of entry. To in order regardless of the fact that he figures out how to download the cipher text from the Cloud as he doesn't have the key for decoding. A long time ago the information holder chooses to deny a client from getting to their information; one minor arrangement would be for the information manager to decode the information and the information once more, huge trouble on the information holder when considering gathering sizes in surplus of thousands to a large number of clients. Moreover, as parts of the gathering consistently join and leave, constantly descrambling information and sending re-encryption keys to a gathering of this size gets to be unreasonable for the information manager and infeasible to actualize in this present reality. Right now, there is continuous research on this issue.

A. Advantage

- Minimize cloud environment cost.
- More secure for distributed data sharing in cloud.
- Reduce the communication costs.
- Easy for the cloud environment remainder a task in progress.
- Easy to sharing and distributed the client systems.

IV. EVALUATION

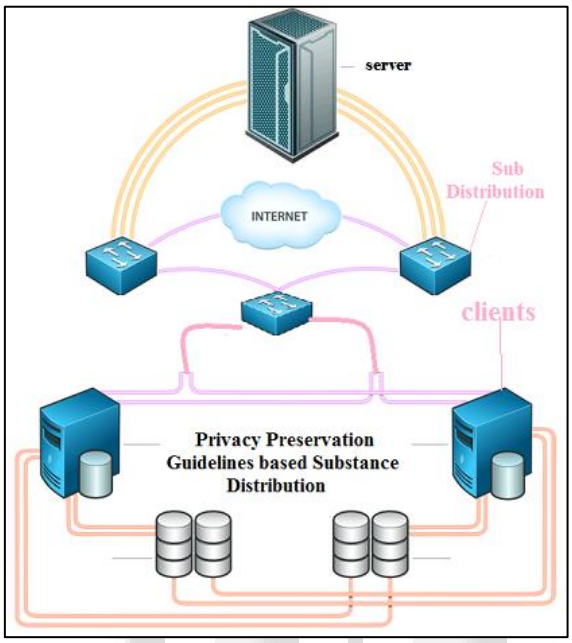

Fig. 1: This Proposed System

This research shows that our move toward is practical and could be used in Privacy preservation guidelines based substance distribution using cloud storage system. The empirical results show that cost reduction, time consuming, provide more security.

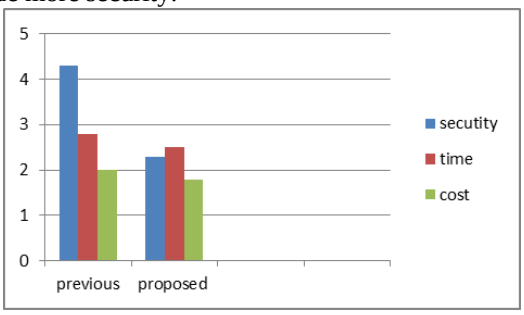

Fig. 2: compare the previous and proposed system analysis the graph

V. CONCLUSION

A number of serious pieces of technology, such as a solution for federate trust, are not yet fully realized, impinging on winning cloud computing deployments. Determining the security of complex computer systems self-assured together is also ancient security issue that plagues large-scale computing in general and cloud computing in particular. Attain high-assurance quality in implementations has been an elusive goal of computer security researchers and practitioners and, as established in the example specified in this report, it's having own compensation and drawback. The developed system ignore the frontage could be used in

any network services for network security. The concept of Privacy preservation guidelines based substance distribution technique enhances the speed of encryption processes and speed distributed application on clouds. Thus the system is justified for its use in securing files.

REFERENCES

[1] Adya, W.J. Bolosky, M. Castro, G. Cermak, R. Chaiken, J.R.Douceur, J. Howell, J.R. Lorch, M. Theimer, and R. Wattenhofer, "Farsite: Federated, Available, and Reliable Storage for an Incompletely Trusted Environment," Proc. Fifth Symp. Operating System Design and Implementation (OSDI), pp. 1-14, 2002.

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

[4] Brownbridge.D.R., L.F. Marshall, and B. Randell, "The Newcastle Connection or Unixes of the World Unite!," Software Practice and Experience, vol. 12, no. 12, pp. 1147-1162, 1982.

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

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

[7] Druschel. P and A. Rowstron, "PAST: A Large-Scale, Persistent Peer-to-Peer Storage Utility," Proc. Eighth Workshop Hot Topics in Operating System (HotOS VIII), pp. 75-80, 2001.

[8] Haeberlen.A, A. Mislove, and P. Druschel, "Glacier: Highly Durable, Decentralized Storage Despite Massive Correlated Failures," Proc. Second Symp. Networked Systems Design and Implementation (NSDI), pp. 143-158, 2005.

[9] Hsiao-Ying Lin, Member, IEEE, and Wen-Guey Tzeng, Member "A Secure Erasure Code-Based Cloud Storage System with Secure Data Forwarding"vol. 23, no. 6, june 2012.

[10] Kubiatowicz. J, D. Bindel, Y. Chen, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. eatherspoon, W. Weimer, C. Wells, and B. Zhao, "Oceanstore: An Architecture for Global-Scale Persistent Storage," Proc. Ninth Int'l Conf. Architectural Support for Programming Languages and Operating Systems (ASPLOS), pp. 190-201, 2000.

[11] Mambo.M and E. Okamoto, "Proxy Cryptosystems: Delegation of the Power to Decrypt Ciphertexts," IEICE Trans. Fundamentals of Electronics, Comm. and Computer Sciences, vol. E80-A, no. 1, pp. 54- 63, 1997.

[12] Shao.J and Z. Cao, "CCA-Secure Proxy Re-Encryption without Pairings," Proc. 12th Int'l Conf. Practice and Theory in Public Key Cryptography (PKC), pp. 357-376, 2009.

