A Secure Cloud Storage System with increased Availability and Robustness

Premkumar M.

Department of Pervasive Computing Technology, University College of Engineering,
BIT Campus, Tiruchirappalli.
shreeprem4u@gmail.com

Abstract- A cloud storage system is a collection of storage servers capable of storing huge volumes of data. Security is the main issue in this system of storing a data in third party cloud storage area. To overcome this issue by untrusted third party data being stored should be secured while storing data in the storage server. This paper explains the way of storing data which increases the availability of data and thus robustness of the system. It also describes the additional functionalities performed by servers like encoding of data and proxy re-encryption when data forwarding required which minimizes interaction of user on processing data. This method integrates an encryption, encoding and proxy re-encryption scheme which achieves data confidentiality and integrity. The availability of data is promised till the single storage server is available and thus robustness of the system will be increased. The implementation will be done using the Google app - engine.

Keywords- encryption, encoding, proxy re-encryption, storage servers, key servers

I. INTRODUCTION

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. Some important characteristics of cloud computing are on-demand self service, ubiquitous network access, resource pooling- location independence, rapid elasticity, measured service. It is an Internet based computing in which common resources will be shared among several users. The various types of cloud are private cloud, public cloud and hybrid cloud. A private cloud is implemented inside an organization or selected business groups and access is restricted within that organization or groups. Public cloud can be accessed by any user. Hybrid cloud is the combination of both private and public clouds. By the nature cloud environment capital expenditure is converted into operational expenditure which reduces cost. Servers and storage devices are shared and utilization is increased using Virtualization. Also applications can be migrated from one server to another. Cloud computing is multitenant that is it supported sharing of resources and costs among a large pool of resources [5].

Cloud providers offer their services among three models namely infrastructure as a service (IaaS), platform as a service (PaaS) and software as a service (SaaS). In IaaS model, service offers as computers, as physical and virtual machines. In PaaS model, providers offer a computing environment which includes an operating system, programming language, database and web

server. In the SaaS model, any user can use any software without installing it on their computer but working as installed in their personal computer. The main advantage of cloud application over other applications is its scalability. It is achieved through dynamic replication of tasks on multiple virtual machines. Though cloud computing achieved increased popularity, concerns is being voiced over security issues. The traditional security mechanisms are being re-constructed as a new deployment model is differing from traditional models. Security issues are categorized into sensitive data access, privacy, accountability, recovery, account control and multi-tenancy issues. Solutions to these security issues vary from cryptography, to use of multiple cloud providers, standardization of APIs and improving virtual machine support and legal support [2] [3] [4]. An optimized approach to secure the user data, minimal user interaction in processing data and method of storing will be discussed.

II. Related Works

In recent advanced computing environment, all kinds of machines and devices are networked and communicate with each other. Since they are storing and sharing data both sensitive and non-sensitive among them the need for security is essential. In cloud computing which is the most recent developing technology in a network related computing environment bundled almost any type of business into it, security is the most concern.

Mahesh Kallahalla et al. [6] introduced a file system called Plutus provided high security in an untrusted server. All data stored in encrypted format and also applied de-centralized key management technique. In specific that file system focused on prevention and detection of un-authorized modification of data, read/write controls and user's access privileges.

There is a variety of encoding techniques are already available in networking concept. Especially our concentration focused on encoding techniques which are de-centralized in nature. Alexandros G.Dimakis et al. [7] proposed a de-centralized erasure code for networked storage which is distributed. It was a mathematical approach of random linear codes over a finite field with the specific randomized structure of their generator matrix, also many other approaches have already proposed to secure and encoding schemes [8] [9] [10].

Proxy re-encryption scheme in general is the process of encrypting ciphertext by a proxy on behalf of any user. There are multiple proxy re-encryption schemes are proposed. Matt Blaze et al. [11] have proposed proxy cryptography in which un-trusted third party converts cipher- text without accessing either original plaintext or the secret component of the keys. The key-pair holders generate and publish proxy key such that the original message can be gotten by the receiver using decryption key.

Qiang Tang [12] proposed a type-based proxy re-encryption scheme in which user can categorize his data into different types and apply proxy cryptography according to the type of message he wants to send to different type of users. Jun Shao and Zhenfu Cao [13] proposed proxy re-encryption scheme without pairings which is a unidirectional approach and achieved security over Chosen Ciphertext Attack (CCA).

Adi Shamir [14] proposed a mathematical approach to key management. In his method he divided the data into a number of pieces and user can retrieve the original data from k pieces of the data even when k-1 pieces means no information about the message. A user can construct robust key management technique by using his method.

III. System Architecture

Cloud computing is purely an internet-based computing environment involves sharing of resources and services among different users. In order to reduce capitalization cost, now-a-days many large business sectors turns their interest in storing their huge data in the cloud. Their data contain both sensitive and non-

sensitive information. Security becomes most concern in this case while storing sensitive data in the cloud.

The system architecture in fig.1consists of 'p' number of storage servers (SS_1, SS_2... SS_p) for storing data and 'q' number of key servers (KS_1, KS_2, .., KS_q) for key management functionality. There are four modules present in this architecture namely system setup, data storage, data forwarding and data retrieval.

In system setup phase, every users has assigned their private key (PK) and public key (UK). Users stored their keys in any one of available key servers. Key server which receives a key pair replicated it to all other available key servers. Now all key servers contain private and public key of every user.

In data storage phase, the message 'M' is encrypted using an encryption technique and forwards it to any one storage server. The storage server which receives the encrypted message converts it into codewords using an encoding technique and stores it. Also it replicates codewords to all other available storage servers.

In data forwarding phase, when a user A wants to forward his data to another user B; user A sends a re-encryption key to all storage servers. Using re-encryption key storage servers do proxy re-encryption on codewords and stores the re-encrypted codewords.

In data retrieval phase, data retrieval request is sent to any one key server by the user who wants to retrieve data. When the user is authenticated by key server, code words in storage server are partially decrypted by key server and sends to the user. Upon receiving the decrypted message user can get the original message by proper combining process.

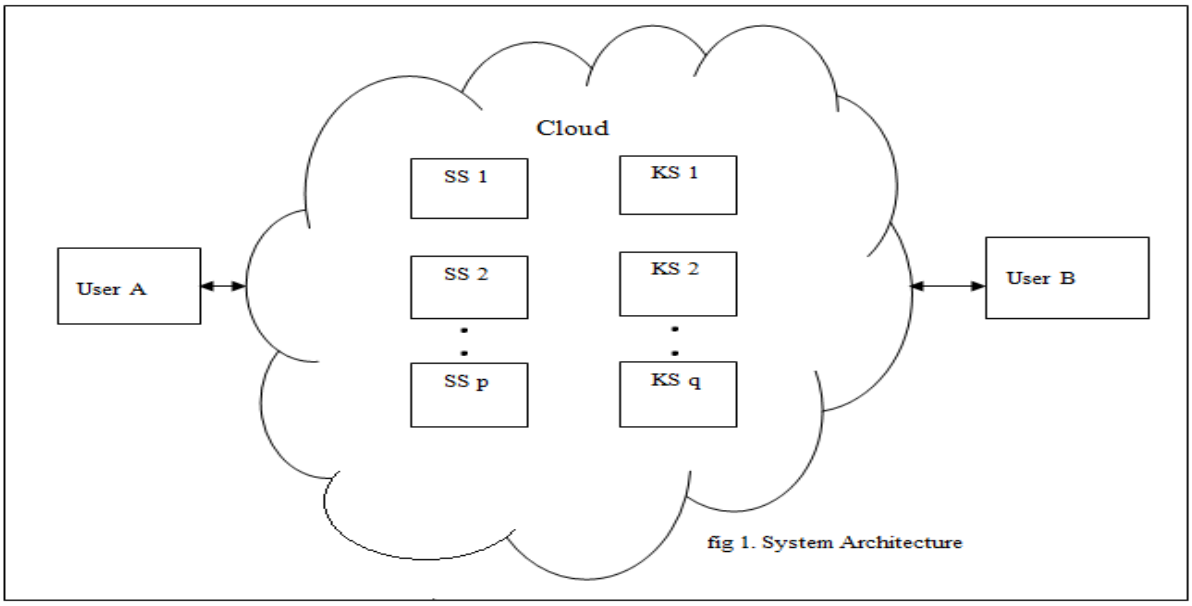

fig 1. System Architecture

IV. Implementation

In system setup module, system parameters are generated and also users are assigned to their private and public keys. The private key of every users are shared with all key servers.

- **setUp( )** function generates system parameters.
- **keyGenerator( )** function generates private key (PK) and public key (UK) for a user.
- **Share Key ()** function sends a private key of user to key servers KS.

In data storage module, initially user A divides the plaintext P into 'k' blocks P_1, P_2, …, P_k. Each block is identified with unique identifiers and encrypted using

public key of the user UK_A and ciphertext $C_1, C_2, ..., C_k$ with unique ids. The ciphertext blocks are sent to anyone available storage server for example SS_1. Upon receiving ciphertext, the storage server SS_1 encodes it and stores the code words (CW). Also SS_1 replicates the codewords it stored on other available storage servers. The algorithms involved in this module are

- **encryptor(UK, P_1, P_2, … , P_k)** converts plaintext into ciphertext blocks $C_1, C_2, ..., C_k$.
- **encoder (C_1,C_2,…, C_k)** encodes ciphertext into codewords.

In data forwarding module, if user A wants to forward his message to user B he needs to generate re-encryption key to perform encryption on codewords. The re - encryption key is generated and sends to storage server by example SS_1. Storage server SS_1 performs re-encrypts and stores the resultant codewords (CW'). The algorithms functioned here are

- **reKeyGenerator(PK_A,UK_A,ID,UK_B)** generates re-encryption key $REK_{A\to B}$ by user A.
- **reEncryptor($REK_{A\to B}$,CW)** performs encryption on stored codewords (CW) and stores codewords (CW').

In data retrieval phase, the process may be user A wants his own message to retrieve or user B wants the message forwarded by user A. In case of user B wants message sent by user A to him, he sends a request to any one key server by example KS_1. After executing proper authentication process, key server forwards the request any one storage server SS_1. On receiving codewords (CW) from storage server it performs decryption process and sends to user B. A new user can get original plaintext by required combine process. The algorithms in this module are

- **mesRequest(PK_B,UK_B,ID)** from user B to key server KS_1.
- **decryptor(CW')** does partial decryption by key server KS_1.
- **combine()** used to get original plaintext.

The data flow in the system is diagrammatically explained in fig. 2 data flow diagram.

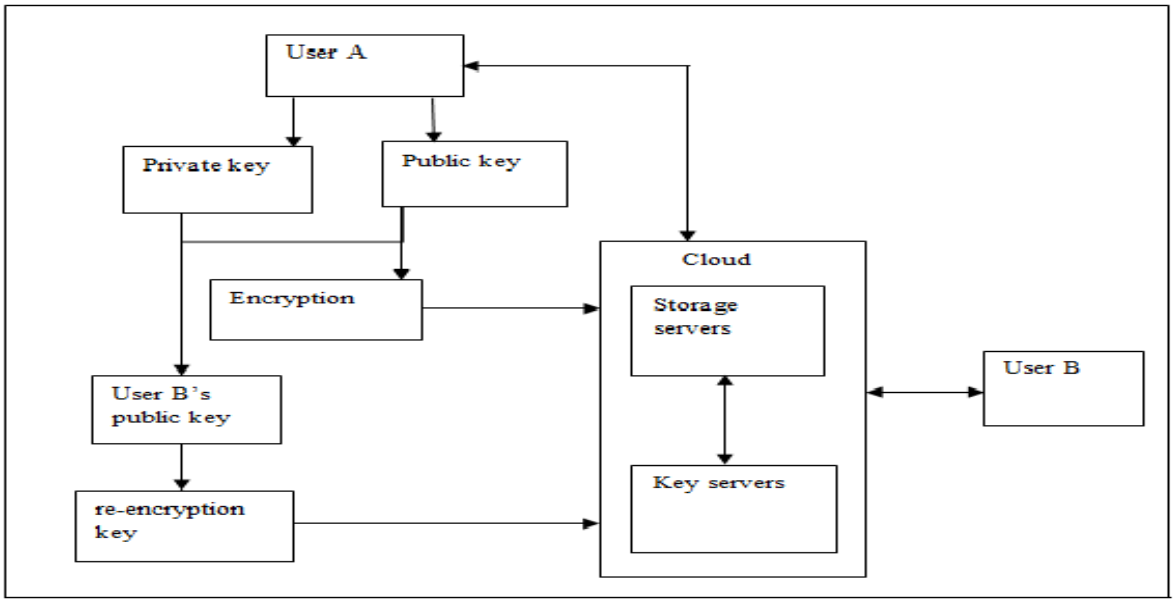

fig. 2 Dataflow diagram

V. Conclusion And Future Work

This paper proposed a way of storing data in a cloud storage system such that the computation cost of storing data is reduced when compared with other previous approaches. Also the availability of data is sustained until one storage server is available. This increases the robustness of the storage system. Though this approach has several advantages still there is some drawbacks present. Key servers meant for key management is not maintained well in terms of security. The private keys of all users are available with those key servers and security breach can be occurred when hacking those key servers. Future works will be focussed on strengthening key server security and also designing different approaches for key management functionality.

References

[1] S. M. Metev and V. P. Veiko, *Laser Assisted Microtechnology*, 2nd ed., R. M. Osgood, Jr., Ed. Berlin, Germany: Springer-Verlag, 1998.

[2] Hsiao-Ying Lin and Wen-Guey Tzeng"A secure erasure code-based cloud storage system with secure data forwarding" *IEEE transactions on parallel and distributed systems*, vol 23, No. 6, june 2012.

[3] Zissis, Dimitrios; Lekkas (2010). "Addressing cloud computing security issues". *Future Generation Computer Systems*.

[4] Armbrust, M; Fox, A., Griffith, R., Joseph, A., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Zaharia, (2010). "A view of cloud computing.". *Communication of the ACM* **53** (4): 50–58.

[5] Anthens, G. "Security in the cloud". *Communications of the ACM***53** (11). http:// en.wikipedia.org/wiki/Cloud_computing.

[6] M. Kallahalla, E. Reidel, R. Swaminathan, Q. Wang, and K. Fu, "Plutus:Scalable Secure File Sharing on Untrusted Storage," *Proc. Second USENIX Conf. File and Storage Technologies (FAST)*, pp. 29-42, 2003.

[7] A.G. Dimakis, 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.

[8] S.C. Rhea, P.R. Eaton, D. Geels, H. Weatherspoon, B.Y. Zhao, and J. Kubiatowicz, "Pond: The Oceanstore Prototype" *Proc. Second USENIX Conf. File and Storage Technologies (FAST)*, pp. 1-14, 2003.

[9] R. Bhagwan, K. Tati, Y.C. Cheng, S. Savage, and G.M. Voelker, "Total Recall: System Support for Automateed Availability Management," *Proc. First Symp. Networked Systems Design and Implementation (NSDI)*, pp. 337-350, 2004.

[10] A.G. Dimakis, 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.

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

[12] Q. Tang, "Type-Based Proxy Re-Encryption and its Construction," *Proc. Ninth Int'l Conf. Cryptography in India: Progress in Cryptography (INDOCRYPT)*, pp. 130-144, 2008.

[13] J. Shao 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.

[14] A. Shamir, "How to Share a Secret," *ACM Comm*, vol. 22, pp. 612-613, 1979.

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

[16] P. Druschel 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.

[17] A. 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.

[18] Z. Wilcox-O'earn and B. Warner, "Tahoe: The Least-Authority Filesystem," Proc. Fourth ACM Int'l Workshop Storage Security and Survivability (StorageSS), pp. 21-26, 2008.

[19] H.Y. Lin and W.G. Tzeng, "A Secure Decentrealized Erasure Code for Distributed Systems," vol. 21, no. 11, pp. 1586-1594, Nov. 2010.

[20] R. Sandberg, D. Goldberg, S. Kleiman, D. Walsh, and B. Lyon, "Design and Implementation of the Sun Network Filesystem," Proc. USENIX Assoc. Conf., 1985.