Enhancing Data Confidentiality in Cloud System using Key-Private PRE Scheme

Abhinav.A.Baone, K.R.Jansi

Abstract— Cloud computing is an approach to computing that leverages the efficient pooling of an on-demand, self-managed, virtual infrastructure. It abstracts the business services from complex IT infrastructure by making use of virtualization and pooling of resources. The risks involved on evaluation are that it requires trusting of cloud platform provider for availability and data security. It can also raise legal concerns while storing data outside customer premises. Intentional or unintentional release of secure information to an untrusted environment is the major security issue which is compromised. Data confidentiality is the main threat often experienced by the end users while outsourcing services to third parties. To overcome this scenario encryption schemes are used, but it limits the functionality of storage system since only few operations are supported over encrypted data. The proposed idea is to make use of key private proxy re-encryption scheme over encrypted data so that secured cloud storage system can be established. The model will not only support secure robust data storage and retrieval but it also allows different hosts to forward there data among each other via virtual machines. The main technical work is to implement key-private PRE scheme which supports re-encryption over encrypted message as well as forwarding operations over re-encrypted messages. To add-on system integrates all operations together to provide enhanced security and restrict data breaching. Entire focus is on building a secure cloud system which improves the factors such as robustness, confidentiality, functionality of an application.

Index Terms— Cloud computing, key-private pre scheme, data confidentiality

I. INTRODUCTION

Cloud computing technology is enabling IT to do more with the infrastructure that already exists, as well as adding new ways to expand capacity quickly and economically by using external cloud computing resources. Cloud computing is the long dreamed vision of computing as a utility, where users can remotely store their data into the cloud so as to enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources**.** The many advantages of cloud computing are increasingly attracting individuals and organizations to move their data from local to remote cloud servers [1]. In addition to the major cloud infrastructure providers [2], such as Amazon, Google, and Microsoft, more and more third-party cloud data service providers are emerging with more offerings.

Main concerns on data security with cloud storage are arising due to unreliability of the service. For example, recently more and more events on cloud service outage or server corruption with major cloud service providers are reported [3][4], be it caused by Byzantine failures and/or malicious attacks.

Manuscript received April, 2014.

Mr.Abhinav.A.Baone, Department of Computer Science and Engineering, SRM University ,Tamil Nadu,India.

Asst Prof. K.R.Jansi, Department of Computer Science and Engineering, SRM University, Tamil Nadu, India.

Such a reality demands for reliable data storage to tolerate certain outage/corruption. In particular, the cloud storage service should offer cloud customers with capabilities of: 1) timely detection of any server (and hence data) corruption event, 2) correct retrieval of data even if a limited number of servers are corrupted, and 3) repair of corrupted data from uncorrupted data.

A. Security Issues and Challenges

IaaS (Infrastructure as a Service), PaaS (Platform as a Service), and SaaS (Software as a Service) are three general models of cloud computing. Each of these models possesses a different impact on application security [5].

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 requirement for storage systems. There have been many proposals of storing data over storage servers [6]. 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.

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. To well fit the distributed structure of systems, we require that servers independently perform all operations. With this consideration, we propose a key private proxy re-encryption scheme and integrate it with an encrypted data to form a secure distributed storage system. Our system meets the requirements that storage servers independently perform encoding and re-encryption and key servers independently perform partial decryption.

B. Our contributions

Assume that 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. Our contributions are as follows: We construct a secure cloud storage system that supports the function of secure data forwarding by using a threshold proxy re-encryption scheme. We present a general setting for the parameters of our secure cloud storage system. Our parameter setting of $n = ak^c$ supersedes the previous one of $n = ak\sqrt{k}$, where $c > 1.5$ and $a > \sqrt{2}$ [7]. Our result $n = ak^c$ allows the number of storage servers be much greater than the number of blocks of a message. In practical systems, the number of storage servers is much more than k.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

C. Different Types of Cloud Computing

Companies can leverage cloud computing for access to software, development platforms and physical hardware. These assets become virtualized and available as a service from the host:

1) *Application and Information cloud:* Sometimes these referred to as Software-as-a-Service, this type of cloud is referring to a business-level service. Typically available over the public internet, these clouds are information-based.

2) *Development cloud:* They are also known as Platform-as-a-Service, cloud development platforms enable application authoring and provide runtime environments without hardware investment.

3) *Infrastructure cloud:* These are also referred to as Infrastructure-as-a-Service, this type of cloud enables IT infrastructure to be deployed and used via remote access and made available on an elastic basis.

II. PROPOSED METHOD

The proposed method has a key private proxy re-encryption scheme applied on splitted data which can be stored, retrieved from cloud server securely preventing data breaching and ensures that a secure cloud storage system is formulated.

The distributed cloud 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. The problem of forwarding data to another user by storage servers directly under the command of the data owner.is addressed considering 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. The proposed idea has a new key private proxy re-encryption scheme to form a secure distributed storage system. The encryption scheme supports encoding operations over encrypted messages and forwarding operations over encrypted and encoded messages. [11]

A. Advantages

1) Data confidentiality can be enhanced by using key private proxy re-encryption scheme
2) Tight integration of encoding, encryption, and forwarding makes the storage system efficiently meet the requirements of data robustness, data confidentiality, and data forwarding.
3) The storage servers independently perform encoding and re-encryption process and the key servers independently perform partial decryption process.
4) More flexible adjustment between number of storage server.

III. RELATED WORKS

At the early years, the Network-Attached Storage (NAS) [8] and the Network File System (NFS) [9] provide extra 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. However, this method is expensive as z replicas result in z times of expansion. The Type-Based proxy re-encryption schemes proposed by Tang [10] provide a better granularity on the granted right of a re-encryption key. A user can decide which type of messages and with whom he wants to share in this kind of proxy re- encryption schemes. Key-private proxy re-encryption schemes are proposed by Ateniese et al. [11]. In the key-private proxy re-encryption scheme, given the re-encryption key, a proxy server always cannot determine the identity of the recipient. This kind of proxy re-encryption schemes provides higher privacy guarantee against proxy servers.

A. Key Private PRE Schemes

Proxy re-encryption (PRE) allows a proxy to convert a cipher text encrypted under one key into an encryption of the same message under another key. The main idea is to place as little trust and reveal as little information to the proxy as necessary to allow it to perform its translations. At the very least, the proxy should not be able to learn the keys of the participants or the content of the messages it re-encrypts. However, in all prior PRE schemes, it is easy for the proxy to determine between which participants a re-encryption key can transform cipher texts. In many applications, data protected under one public key pk_i needs to be distributed to a user with a different public key pk_2. It is not always practical for the owner of sk_i to be online to decrypt these cipher texts and then encrypt these contents a new under pk_2.

As a solution to this key management problem, the concept of proxy re-encryption (PRE) was introduced [12]. Proxy re-encryption is a cryptosystem with the special property that a proxy, given special information, can efficiently convert a cipher text for Alice into a cipher text of the same message for Bob. The proxy should not, however, learn either party's secret key or the contents of the messages it re-encrypts. The main idea is to place as little trust in the proxy as possible.

The server might be told to re-encrypt all category one files with key one and category two files with keys two and three, without the proxy being able to deduce the public keys behind these values. This way, if the proxy is compromised, the adversary will not be able to extract a list of "who was speaking privately with whom". This is highly desirable for many encrypted communication scenarios.

This level of privacy for standard encryption schemes was formalized as key-private (or anonymous) encryption. Achieving key-private PRE is only possible

when the underlying encryption scheme is key-private. The main contribution of this work is the first realization of a key-private PRE scheme. Our construction is efficient, reasonably simple, and secure under basic assumptions about bilinear groups in the standard model. Formally, it is a unidirectional, single-hop, CPA-secure PRE with key-privacy. Thus, we show, for the first time, that this natural extension of anonymous encryption is practical and available for many existing PRE applications.

B. Functionality and Applications

PRE has been proposed for use in email-forwarding [12], secure file systems [13]. All these applications can benefit from the key-privacy property in some way. In email-forwarding, Alice may not want the mail server to know to whom she is delegating her decryption rights. This is similar in the real world to a P.O. Box address where mail can be sent to a physical location but neither the sender nor the carrier may know who the actual recipient is. Alice can hide the fact that Bob is a delegate by instructing the server to convert her encrypted emails via a key-private PRE scheme and to forward the results to an anonymous (or group) email address

In a distributed file system, PRE schemes can be used as an access control mechanism to specify who can access and read encrypted files. Alice may want Bob to read some of her encrypted files, thus she instructs the file system to convert those files using a proxy re-encryption key from Alice to Bob. In a distributed file system, anyone can access those files but only Bob can read them. If the PRE scheme employed is key-private, nobody can even tell who can access and read any file in the system.

C. Key-Private PRE Definitions

We build upon the re-encryption definitions of [13] and [14] to introduce the concept of key privacy. We begin by specifying the input/output behavior of a proxy re-encryption scheme. For simplicity, we will only consider a definition for unidirectional, single-hop PREs. By single-hop, we mean that only original ciphertexts (and not re-encrypted ciphertexts) can be re-encrypted.

Definition: (Unidirectional, Single-HopPRE)

A unidirectional, single-hop, proxy re- encryption scheme is a tuple of algorithms.

N = (SETUP, KEYGEN, REKEYGEN, ENC, REENC, DEC) for message space M:

a) Dec (PP, SKI, CI) ^ M. Given a secret key for user i and a ciphertext for i, the decryption algorithm Dec outputs a message m e M.

A unidirectional, single-hop PRE scheme n is correct with respect to domain M if:

b) For all (pk, sk) e KeyGen (PP) and all m e M, it holds that Dec (PP, sk, Enc (PP, pk, m)) = m.

c) For all pairs (pk_i, ski), (pkj, skj) e KeyGen (PP) and keys rk_i^j e ReKeyGen (PP, sk_i, pkj), and m e M, it holds that Dec (PP, skj, ReEnc (PP, rk_i^j, Enc (PP, pki, m))) = m.

Next, we turn to the issue of what it means for a re-encryption key to be key-private. Informally, we want a proxy to be unable to identify either the delegator i or the delegate j when given the re-encryption key rk_i^j and flexible interaction with the system.

D. Key-Private PRE Scheme

1) Construction:

Scheme n = **(**Setup, KeyGen, ReKeyGen, Enc, ReEnc, Dec) is described as follows:

a) Setup (Setup): Run BSetup(1^k) ^ (q,g, G, G_T, e), where (g) = G. Choose a random generator h e G. Compute Z = e(g, h), and set the public parameters PP = (g, h, Z). In the following, we assume that all parties have PP.

b) Key Generation (KeyGen): Choose random values a_1, a2 e Z_Q and set the public key as pk = (Z^{ai}, ga2) with secret key sk = (a_1, a2).

c) Re-Encryption Key Generation (ReKeyGen): A user A with secret key (a_1, a_2) can delegate to a user B with public key (Z^{bl}, gb2) as:

1. Select random values r, w, e, Z_Q.
2.Compute rk_{A^B} = $((g^{b2})^{ai+r}, h^r, e(g^{b2},h)^w, e(g,h)^w) = (g^{b2(ai+r)}, h^r, Z^{b2w}, Z^w)$.

d) Encryption (Enc): To encrypt a message m e G_T under public key pk_A = (Z^{ai}, ga2), do:

1. Select random value k e Z_Q.
2. Compute the ciphertext (g^k, h^k, m ■ Z^{aik}).

e) Decryption (Dec): Given secret key (a_1,a_2), to decrypt a first-level ciphertext (a, ft), compute m = ft/$a^{1/a2}$; and to decrypt a second-level ciphertext (a,ft,Y), output ± if e(a,h) = e(g,ft), otherwise output m = Y/$e(a,h)^{ai}$.

Fortunately, this scheme is practical and multi-purpose. Public keys can be used either for re- encryption purposes .It allows proxy to convert cipher text under one key into encryption of the same message under another key.

The algorithm restricts access to:

1. Sender secret key used during encryption
2. Contents of messages
3. Identity of sender.

Cipher text can be re-encrypted number of times which defines transitivity property. Higher privacy is guaranteed.

IV. SCENARIO

We present the scenario of the storage system, improving confidentiality issue, and a discussion for a straightforward solution.

A. System Model

As shown in Figure1, The cloud setup consists of several virtual machines installed on it.Virtualisation concept is the base for implementing cloud computing technology[15][16]. Number of virtual machine can be deployed on the local host which completely depends on base system configuration.On each VM different types of server can be made to run. Data owner A selects any plaintext file from local system which is to be uploaded on the cloud. File choosen by user A is encrypted by using encryption algorithm like DES.The encrypted file is not readable to a normal user since that file contains cipher text in it. The same file is given to file splitter module which fragments original file into number of smaller parts. Splitted files can be maintained into different servers so that in case of server failure,each replicas of the file can be retrieved back from anothe workable server.

Secure cloud storage and retrieval

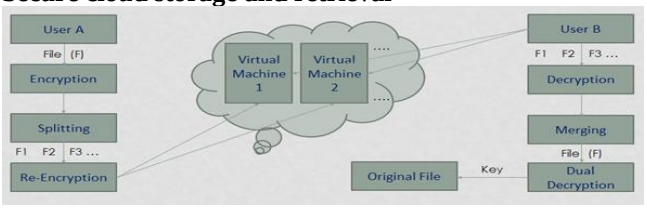

Fig. 1. Detailed working

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Number of splits can be determined by uploader of file.These fragmented files are again encrypted by using Key-private PRE algorithm. Alongwith storage capabilities , many other operations on encrypted message are supported by this algorithm. These new re-encrypted splitted files are send to storage server for storing purpose and further retrieval also.If another User B wishes to access stored file he searches for that file into the cloud servers from local system.Unless search result for requested file is not matched any server will not respond to host B.If the match is found then server will send the desired splitted files to the client B.The decryption algorithm will be applied on splitted files. Afterwards file merger will combine the partitioned files into single file. Partial decryption is done up till now, ciphertext is converted into plaintext which is still not readable.Merged file is brought for dual decryption so that final key is generated.Thus,retrieval is done perfectly alongwith secure data forwarding to the another host.Original file can be retrieved in similar manner and it can viewed easily.

1) Process Encryption: In this module, define the process which will use DES algorithm for encryption. The process can be encrypted by using cryptographic keys. After the process can be encrypted, it can be splitted as different process. Process splitting can also be sectorized by the data owner.
2) Re-encryption-scheme: The Splitted process can be implemented using key private proxy re-encryption scheme. Every splitted process can be re-encrypted in this module. By this proxy functioning, the process can be brought to ready state before sending to storage sector.
3) Secure cloud storage: In this module, the cloud storage is defined with dependable server and splitted re-encrypted data storage in VM.Finite number of server can be declaimed. Data owner can store there process in this secure cloud storage system.
4) Dual-decryption: In this module, process can be again decrypted and then forwarded to the client. The function of this module can be integrated with partial decryption, data retrieval and forwarding. Process retrieval from the cloud storage operation can be performed in this module. Search and select the process which is needed by the user from the secure cloud storage.
5) Data Forwarding: In this module, transfer of merged data to another host takes place. Forwarding process can be claimed as secure data forwarding functioning. Target user must have the retrieval power for the transferring process.

B. Straightforward Solution

A straightforward solution to supporting the data forwarding function in a distributed storage system is as follows: when the owner A wants to forward a message to user B, he downloads the encrypted message and decrypts it by using his secret key. He then encrypts the message by using B's public key and uploads the new ciphertext. When B wants to retrieve the forwarded message from A, he downloads the ciphertext and decrypts it by using his secret key. The communication cost is linear in the length of the forwarded message. The computation cost is the encryption for the owner A, and the decryption for user B.

Key-private proxy re-encryption schemes can significantly decrease communication and computation cost of the owner. The owner sends a re-encryption key to storage servers such that storage servers perform the re-encryption operation for him. Thus, the communication cost of the owner is independent of the length of forwarded message and the computation cost of re-encryption is taken care of by storage servers. Proxy re-encryption schemes significantly reduce the overhead of the data forwarding function in a secure storage system.

V. CONCLUSION

Secure data forwarding in cloud storage system can be made possible. Message can be forwarded securely between end-users through cloud environment. Data confidentiality, robustness, functionality will be improved to the greater extent.Usage of separate key server for key management makes efficient working and data processing rate can be reduced to the greater extent.

ACKNOWLEDGMENT

I am thankful and owe my sincere gratitude to my project guide Mrs.K.R.Jansi along with the entire department of Computer Science & Engineering of SRM University for their overwhelm support to carry out this project successfully.

REFERENCES

1. S. Kamara and K. Lauter,"Cryptographic cloud storage," in RLCPS, 2010.
2. M. Armbrust, A. Fox, and et al., "Above the clouds: A berkeley view of cloud computing," University of California, Berkeley, Tech. Rep. UCB-EECS-2009-28.
3. "Summary of the amazon ec2 and amazon rds service disruption in the us east region," http://aws.amazon.com/message/65648/.
4. "Microsoft cloud data breach heralds things to come," http://www.tech-world.com.au/article/372111/microsoft_cloud_data_breach_heralds_things _come/.
5. John, H.: Security Guidance for Critical Areas of Focus in Cloud Computing (2009), http:// www .cloudsecurity alliance.org/gui- dance/
6. Z. Wilcox-O'Hearn and B. Warner, "Tahoe: The Least-Authority Filesystem," Proc. Fourth ACM Int'l Workshop Storage Security and Survivability (StorageSS), pp. 21-26, 2008.
7. H.-Y. Lin and W.-G. Tzeng, "A Secure Decentralized Erasure codefor Distributed Network Storage," IEEE Trans. Parallel & Distributed Systems, vol. 21, no. 11, pp.1586-1594, Nov. 2010.
8. D.R. Brownbridge, 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.
9. R. Sandberg, D. Goldberg, S. Kleiman, D. Walsh, and B. Lyon, "Design and Implementation of the Sun Network Filesystem," Proc. USENIXAssoc. Conf., 1985
10. Q.Tang,"Type BasedProxy Re-Encryption and Its Construction," Proc. Ninth Int'l Conf. Cryptology in India: Progress in Cryptology (INDOCRYPT), pp. 130-144, 2008.
11. G.Ateniese, K. Benson, and S. Hohenberger, "Key-Private Proxy ReEnc - ryption," Proc. Topics in Cryptology (CT-RSA), pp. 279-294, 2009.
12. Matt Blaze, Gerrit Bleumer, and Martin Strauss. Divertible protocols and atomic proxy cryptography. In EUROCRYPT '98, volume 1403 of LNCS, pages 127-144, 1998.
13. Giuseppe Ateniese, Kevin Fu, Matthew Green, and Susan Hohenberger. Improved Proxy Re- encryption Schemes with Applications to Secure Distributed Storage. In NDSS, pages 29-43, 2005.
14. Ran Canetti and Susan Hohenberger. Chosen-ciphertext secure proxy re-encryption. In ACM CCS, pages 185-194. ACM, 2007.
15. ShengmeiLuo; ZTECorp.,Shenzhen,China; ZhaojiLin ; Xiaohua Chen ; Zhuolin Yang .“Virtualization security for cloud computing service”.In Cloud and Service Computing (CSC), 2011
16. Volokyta, A. ; Nat. Tech. Univ. of Ukraine Kyiv Polytech. Inst., Kiev, Ukraine ; Kokhanevych, I. ; “Secure virtualization in Cloud computing”. In Modern Problems of Radio Engineering Telecommunications and Computer Science (TCSET), 2012