ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol. 3 , No.1, Pages : 563– 568 (2014)
Special Issue of ICETETS 2014 - Held on 24-25 February, 2014 in Malla Reddy Institute of Engineering and Technology, Secunderabad– 14, AP, India

A Secure Cloud Storage System with Secure Data Forwarding

B.H Chandana Reddy[1], S.Thulasi krishna [2].
M.Tech student, Dept of CSE, Kuppam Engineering college[1],
Asso.prof, Dept of CSE, Kuppam Engineering College[2].

Abstract:

CLOUD computing presents a new way to enhancement the current utilize and delivery model for IT services based on the Internet, by providing for dynamically scalable and often virtualized resources as a service over the Internet. To date, there are a number of notable commercial and individual cloud computing services, including Amazon, Google, Microsoft, Yahoo, and Sales force [19]. Details of the services provided are abstracted from the users who no longer need to be experts of technology infrastructure. A novel highly decentralized information accountability framework to keep track of the actual usage of the users' data in the cloud. In particular, an object-centered approach that enables enclosing our logging mechanism together with users' data and policies. We leverage the JAR programmable capabilities to both create a dynamic and traveling object, and to ensure that any access to users' data will trigger authentication and automated logging local to the JARs. In our proposed system we address 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 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.

Index Terms—Decentralized erasure code, proxy re-encryption, threshold cryptography, secure storage system.

I. INTRODUCTION

In this paper, we address 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 crypto-graphic functions on behalf of the user. These key servers are highly protected

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol. 3 , No.1, Pages : 563– 568 (2014)
Special Issue of ICETETS 2014 - Held on 24-25 February, 2014 in Malla Reddy Institute of Engineering and Technology, Secunderabad– 14, AP, India

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 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. The tight integration of encoding, 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. Our system meets the requirements that storage servers independently perform encoding and re-encryption and key servers independently perform partial decryption. Moreover we consider the system in a more general setting than previous works. This setting allows more flexible adjustment between the number of storage servers and robustness.

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:

1. 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.
2. 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* $c \geq 1.5$ *and* $a > \sqrt{2[6]}$
3. encoding, that is, each codeword symbol is independently computed. 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

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. The sacrifice is to slightly increase the total copies of an encrypted message symbol sent to storage servers. Nevertheless, 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.

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol. 3 , No.1, Pages : 563– 568 (2014)
Special Issue of ICETETS 2014 - Held on 24-25 February, 2014 in Malla Reddy Institute of Engineering and Technology, Secunderabad– 14, AP, India

2 RELATED WORKS

We briefly review distributed storage systems, proxy re-encryption schemes, and integrity checking mechanisms.

2.1 Distributed Storage Systems

At the early years, the Network-Attached Storage (NAS) [7] and the Network File System (NFS) [8] provide extra storage devices over the network such that a user can access the storage devices via network connection. Afterward, many improvements on scalability, robustness, efficiency, and security were proposed [1], [2], [9]. 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.

2.2 Proxy Re-Encryption Schemes

Proxy re-encryption schemes are proposed by Mambo and Okamoto [14] and Blaze et al. [15]. In a proxy re-encryption scheme, a proxy server can transfer a ciphertext under a public key PKA to a new one under another public key PKB bv using the re-encryption key $RK_{A\rightarrow B}$. The server does not know the plaintext during transformation. Ateniese et al. [16] proposed some proxy re-encryption schemes and applied them to the sharing function of secure storage systems. In their work, messages are first encrypted by the owner and then stored in a storage server. When a user wants to share his messages, he sends a re-encryption key to the storage server. The storage server re-encrypts the encrypted messages for the authorized user. Thus, their system has data confidentiality and supports the data forwarding function. Our work further integrates encryption, re-encryption, and encoding such that storage robustness is strengthened.

2.3 Integrity Checking Functionality

Another important functionality about cloud storage is the function of integrity checking. After a user stores data into the storage system, he no longer possesses the data at hand. The user may want to check whether the data are properly stored in storage servers. The concept of provable data possession [20], [21] and the notion of proof of storage [22], [23], [24] are proposed. Later, public auditability of stored data is addressed in [25]. Nevertheless all of them consider the messages in the clear text form.

3 SCENARIO

We present the scenario of the storage system, the threat model that we consider for the confidentiality issue, and a discussion for a 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 his secret key. The whole data forwarding process needs three communication rounds for A's downloading and uploading and B's downloading. The communication cost

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol. 3 , No.1, Pages : 563– 568 (2014)
Special Issue of ICETETS 2014 - Held on 24-25 February, 2014 in Malla Reddy Institute of Engineering and Technology, Secunderabad– 14, AP, India

is linear in the length of the forwarded message. The computation cost is the decryption and encryption for the owner A, and the decryption for user B.

4 CONSTRUCTION OF SECURE CLOUD STORAGE SYSTEMS

Before presenting our storage system, we briefly introduce the algebraic setting, the hardness assumption, an erasure code over exponents, and our approach is

System setup: The algorithm setup (1^T) generates the system parameters μ. A user uses key Gen (μ) to generate his public and secret key pair and share key gen (-) to share his secret key to a set of m key serves with a threshold t, where $k \le t \le m$. The user locally stores the third component of his secret key.

- Setup (1^λ) Run Gen (1^λ) to obtain (g,h,e, G_1,G_2,p), where
- Key Gen (μ) for a user A, the algorithm selects a_1,a_2,a_3εRZp and sets
- Share key Gen (SK_A, t,m), this algorithm shares the secret key
- Enc (PKA,ë,m1,m2…mk) for this algorithm computes

Where and 0 is the leading bit indicating an original ciphertext.

- Encode (C_1,C_2,…. C_k) for each ciphertext C_i, the algorithm randomly selects a coefficient g_i. If some cipher text C_1 is (0,1,ë,1) the coefficient g_i is set to 0. Let Ci=(0, α_i,β,χ_i). the encoding process is to compute an original codeword symbol
- Keyrecover ($SK_{A,i1}$, $SK_{A,i2}$… $SK_{A,it}$). Let T= {i_1,i_2,….i_t}. this algorithm recovers a1 via lagrange interpolation as follows:

Rekey Gen (PK_A, SK_A, ID, PK_B). This algorithm selects and computes

5 DISCUSSION AND CONCLUSION

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 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 to n codeword 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 performs partial decryption. Our storage system and some newly proposed content addressable file systems and storage system [27], [28], [29] 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] 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

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol. 3 , No.1, Pages : 563– 568 (2014)
Special Issue of ICETETS 2014 - Held on 24-25 February, 2014 in Malla Reddy Institute of Engineering and Technology, Secunderabad– 14, AP, India

for Global-scale persistent programming Languages and Operating systems (ASPLOS), pp.190-201, 2000.

[2 P. Druschel and A. Rowstron,

[3] A. Adya, W.J. Bolosky, M. Castro, G. Cermak, R. Chaiken, J.R. Douceur, J. Howell, J.R. Lorch, M. Theimer, and R. Wateenhofer, "Farsite: Federated, Available and reliable storage for an incompletely trusted environment," Proc. Fifth Symp. Operating system design and implementation (OSDI), pp.1-14, 2002.

[4] A. Haeberlen, 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.

[5] Z. Wilcox-O' Heam and B. Warner, "Tahoe: The Least-Authority Filesystem," Proc. Fourth ACM Intel workshop storage security and survivability (Storage SS), pp. 21-26, 2008.

[6] H. Y. Lin and W.G. T zeng, "A Secure decentralized erasure code for distributed systems, vol.21, no.11, pp. 1586-1594, Nov.2010.

[7] 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.

[8] R. Sandberg, D. Goldberg, S. Kleiman, D. Walsh, and B. Lyon "Design and Implementation of the sun network filesystem", Proc. USENIX Assoc. Conf., 1985.

"PAST: A Large-scale, persistent peer-to-peer storage utility", Proc. Eighth Workshop Hot topics in operating system (Hot OS VIII), pp.75-80, 200

[9] M. Kallahalla, E. Riedel, 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.

[10] 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.

[11] R. Bhagwan, K. Tati, Y.C. Cheng S. Savage, and G. M. Voelker, "Total Recall: System Support for Automated availability management", Proc. First Symp. Networked systems design and implementation (NSDI), pp. 337-350, 2004.

[12] A.G. Dimakis, V. Prabhakaran, and K. Ramchandran, "Ubiquitous Access to Distributed data in large scale sensor networks through decentralized erasure codes", Proc. Fourth Intel Symp. Information processing in sensor networks (IPSN), pp.111-117, 2005.

[13] A.G. Dimakis, V. Prabhakaran, and K. Ramchandran. "Decentralised Erasure codes for distributed networked storage", IEEE Tans. Information Theory, Vol.52, no.6, pp. 2809-2816, June 2006.

[14] M. Mambo and E. Okamoto, "Proxy Cryptosystems :

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol. 3 , No.1, Pages : 563– 568 (2014)
Special Issue of ICETETS 2014 - Held on 24-25 February, 2014 in Malla Reddy Institute of Engineering and Technology, Secunderabad– 14, AP, India

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.

[15] M. Blaze, G. Bleumer and M. Strauss, "Divertible Protocols and Atomic Proxy cryptography", Proc. Int Conf. Theory and Application of Cryptographic Techniques (Eurocrypt) pp. 127-144, 1998.

[16] G. Ateniese, K. Fur, and M. Strauss, "Divertible protocols and atomic proxy re-encryption schemes with applications to secure distributed storage" ACM Trans. Information and system security, vol. 9, no.1, pp. 1-30, 2006.

[17] Q. Tang, "Type- based proxy re-encryption and its construction", Proc. Ninth Int Conf. Cryptology in India: Progress in Cryptology (Indocrypt), pp.130-144, 2008.

[18] G. Ateniese, K. Benson, and S. Hohenberger, "Key-private proxy re-encryption", Proc. Topcis in cryptology (CT-RSA), pp.279-294, 2009.

[19] J. Shao and Z. Cao, "CCA-Secure proxy re-encryption without pairings," Proc. 12th Int Conference. Practice and theory in public key cryptography (PKC), pp.357-376, 2009.

[20] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson and D. Song, "Provable Data possession at Untrusted stores", Proc. 14th ACM Conf. computer and comm.. security (CCS),pp. 598-609, 2007.

[21] G. Ateniese, R.D. Pietro, L.V. Mancini and G. Tsudik, "Scalable Peterson and D. Song, "Provable data possession", Proc. Fourth Int Conf. Security and Privacy in comm.. networks (Secure comm.), pp.1-10, 2008.

[22] H. Shacham and B. Waters, "Compact proofs of retrievability", Proc. 14th Int Conf. theory and application of cryptology and information security (ASIACRYPT), pp.90-107, 2008.

[23] G. Ateniese, S. Kamara and J. Kalz, "Proofs of storage from homomorphic identification protocols", Proc. 15th Int Conf. theory and application of cryptology and information security (ASIACRYPT) pp. 319-333, 2009.

[24] K.D. Bowers, A. Juels, and A. Opera, "HAIL: High availability and integrity layer for cloud storage", Proc. 16th ACM Conf. Computer and Comm Security (CCS), pp. 187-198, 2009.

[25] C. Wang, Q. Wang, K. Ren and W. Lou, "Privacy- preserving public auditing for data storage security in cloud computing", Proc. IEEE 29th Int Conf computer Comm. (INFOCOM), pp. 525-533, 2010.

[26] A. Shamir, "How to share a secret", ACM comm.. Vol.22, pp.612-613, 1979.

.

[27] C. Dubnicki, L. Gryz, L. Heldt, M. Kacmarczyk, W. Kilian, P. Strzelack, J. Szczepkowski, C. Ungureanu, and M. Welnicki, "Hydrastor: A scalable secondary storage," Proc. Seventh Conf. File and Storage technologies (FAST), pp. 197-210, 2009.