International Journal of Computer Science
Oddity...Probe...Reviste...
http://www.ijcsjournal.com Volume 1, Issue 2, No 4, 2013. ISSN: 2348-6600
Reference ID: IJCS-024 PAGE NO: 128-131

Cloud Storage System Using Secure Data Forwarding

S.SELVANADHI ,
Research scholar,

M.phil(computer science) ,
Tamil university,
Thanjavur-10.

K.RAVIKUMAR,
Asst.Professor
Dept.of computer science,
Tamil university,
Thanjavur-10.
ravikasi2001@yahoo.com

Email-selvanathimca@gmail.com

Abstract— 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. 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. 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.

Keywords: Decentralized erasure code, proxy re-encryption, threshold cryptography, secure storage system.

[1]INTRODUCTION

As high-speed networks and ubiquitous Internet access become available in recent years, many services are provided on the Internet such that users can use them from anywhere at any time. For example, the email service is probably the most popular one. Cloud computing is a concept that treats the resources on the Internet as a unified entity, a cloud. Users just use services without being concerned about how computation is done and storage is managed. Focus 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. 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. 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. Thus, the encoding process for a message can be split into n parallel tasks of generating codeword symbols. 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. The recovery process is the same.

International Journal of Computer Science
Oddity...Probe...Reviste...
http://www.ijcsjournal.com Volume 1, Issue 2, No 4, 2013. ISSN: 2348-6600
Reference ID: IJCS-024 PAGE NO: 128-131

[2] 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 and the notion of proof of storage are proposed. Later, public audit ability of stored data is addressed in. Nevertheless all of them consider the messages in the clear text form.

2.1 Requirements Monitoring: This article does not propose an approach for runtime requirements monitoring, and there are many other research papers. To simplify the discussion and our experimental setup, we have used a trace-based requirements monitor; however, our ideas are applicable to other types of requirement monitors as well.

2.2 Functional Requirements: This work does not propose a heavy weight requirements monitoring for validating functional requirements, as they can very well be monitored by verifying the externally visible interface.

2.3 Notion of Integrity: Intentionally not restricted ourselves to a specific notion of integrity in this article. Any existing notions, along with a corresponding verification mechanism, can be used. In the examples presented in this article, used a notion based on checksum. Briefly, in these examples, consider that a monitor's integrity has not been violated if its checksum as computed by the trust analyzer and signed by the TPM matches the clean room measurements. Our approach can be adapted to use more sophisticated models based on functional equivalence; however, for the proof of concept, consider a checksum based notion of integrity to be sufficient.

2.4 Secrecy and Authenticity Issues: This work is orthogonal and complementary to the secrecy and authenticity research in the Web services security community. Do not focus on securing the interaction between service providers, brokers, and clients, which has been the main focus of many existing approaches, e.g., current standards such as WS-Security and WS-Trust. These approaches address the issue of security-token interoperability and secure transactions. They do not address the integrity issues for components services and they cannot be used directly to certify indisputable trust in an untrusted environment. This approach builds upon existing work on secrecy and authenticity to develop a mechanism for trusting loosely-coupled components in a service-oriented computing environment. The rest of this article is organized as follows: trusted platform modules, which form the basis of our proposed architecture. Describes key ideas of this work and evaluate these contributions. In particular, the former evaluates the feasibility claim and the later evaluates the utility claims, compares and contrasts this work with related approaches. Potential adoption paths for our work in the current service-oriented computing research and practice. Future work and concludes. Now describe key parts of the trust platform module.

[3] CRYPTOGRAPHIC COPROCESSOR

The cryptographic coprocessor implements cryptographic functions executed within the TPM hardware. Hardware or software entities outside the TPM have no access to the execution of these functions. A TPM also contains an RSA accelerator to perform 2,048 bit RSA encryption and decryption. The TPM uses RSA algorithm for signature operations on internal and external items. There is also an engine for computing SHA1 hash for small pieces of data within the TPM. This SHA1 interface is exposed to the software entities outside the TPM to support measurements during the platform boot phases.

3.1 Random Number Generator (RNG)

An RNG is the source of randomness in TPM. It is provided for key generation, nonce generation, and for randomness in signatures. This capability is protected from external access.

3.2 Platform Configuration Registers (PCRs)

PCRs are set of registers that can be used to store the 160-bit hash values obtained using the SHA1 hashing algorithm of the TPM. The hardware ensures that the hash value of any PCR can be changed only by encrypting the new data over the previous hash value of the PCR. Thus, PCRs can be used to indelibly record the history of the machine since the last reboot. The PCRs are cleared off at the time of system reboot.

3.3 Cryptographic Keys

Published by SK Research Group of Companies (SKRGC).

International Journal of Computer Science
Oddity...Probe...Reviste...
http://www.ijcsjournal.com Volume 1, Issue 2, No 4, 2013. ISSN: 2348-6600
Reference ID: IJCS-024 PAGE NO: 128-131

Every TPM is identified by a built-in key called the Endorsement Key, which is included in it by the manufacturer. The key size is 2,048 bits. The trust that one reposes in a TPM comes from the fact that this key is unique and is protected at all times in the TPM. An Endorsement Certificate, which contains the public key of the Endorsement Key, certifies this property. This key can be used by the owner to anonymously confirm that the identity keys were generated by the TPM in their system. In essence, every computer has a unique identity which cannot be repudiated. This can serve to be a fool-proof identity for every user. The TPM manufacturer provides a certificate for the Endorsement Key.

3.4 Attestation Identity Keys (AIKs)

AIKs are used by a privacy certification authority to present different keys to different remote parties to enable the system to hide its platform identity from other systems.

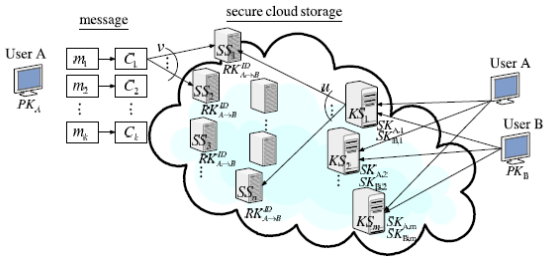
3.5 Certificates

The TPM is also equipped with three kinds of certificates endorsement, platform, and conformance. An endorsement certificate attests that a particular platform configuration is genuine. This contains the public part of the endorsement key. The platform certificate attests that the security components of the platform are genuine. This is provided by the platform vendor and the conformance certificate can be provided by a third party to certify the security properties of the platform.

3.6 TPM Usage Models

Describes three usage models of the TPM. First, hardware protected storage, where TPM is employed to protect sensitive data of the user by encrypting the secret data in such a way that it can only be decoded on a specific hardware that contains the necessary private key. Second, information binding, where critical data is bound to a platform such that it is accessible only if the conditions specified during the binding are met and rendered inaccessible if migrated to a different platform. Third, platform authentication, where attestation identity keys are always bound to the platform. These can be used to authenticate the user and the platform. Our technique uses the third model to authenticate the service implementation platform, including the requirements monitor. Critics of TPM claim that TPMs will have a huge impact on user privacy. Service providers with commercial interest will try to misuse the power of TPM by introducing stricter controls and by eliminating user-anonymity. Although the Jury is still out on the social aspects of TPMs, their wide availability and advantages combine to warrant research on the use of these mechanisms for trusted service-oriented architectures.

[4] INFRASTRUCTURE



4.3 Extensibility for various target services

In this work, THEMIS is focused on SLA-monitoring, especially for IaaS services (in terms of availability and performance) and related billing transactions for mutual verifiability. From the perspective of extensibility, THEMIS should be naturally applicable to various target services as well to improve the accountability of each service. For instance, by applying monitoring techniques S-Mon, we believe that THEMIS can facilitate the cloud-based services with accountability. Examples include PaaS, SaaS, and a cloud storage service. This type of facilitation is possible as long as the monitoring techniques can be plugged into the internal monitoring module of S-Mon. As a result, we believe that the complementarities of THEMIS and the existing monitoring techniques significantly improve the extensibility of this work.

4.4 Multi-CNA support

If different users subscribe to different CNAs on a single physical resource, it becomes necessary for multiple SMons to be deployable for the multiple CNAs. Implementing this support requires multiple S-Mons to be invoked on a single physical resource. In addition, S-Mon needs to implement locking primitives to synchronize access to its global data of a physical TPM because multiple S-Mons shares the physical TPM. By equipping each S-Mon with a virtual TPM (vTPM) device involving the virtualization of the hardware TPM, SMon can overcome this restriction. The

Published by SK Research Group of Companies (SKRGC).

International Journal of Computer Science
Oddity...Probe...Reviste...
http://www.ijcsjournal.com Volume 1, Issue 2, No 4, 2013. ISSN: 2348-6600
Reference ID: IJCS-024 PAGE NO: 128-131

vTPM component proposes a method of virtual zing the hardware TPM. It provides the illusion of a physical TPM to SMons running on a single physical resource. This facility enables the multiple invocation of S-Mon on a physical TPM. Each S-Mon can be mapped to a different CAN because each S-Mon can have a different public key of CNA. Thus, the multiple CNA support would be a promising augmentation of this paper.

[5] CONCLUSION

In this paper, consider a cloud storage system consists of storage servers and key servers. Integrate a newly proposed threshold proxy re-encryption scheme and erasure codes over exponents. The threshold proxy 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, 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 perform partial decryption. Our storage system and some newly proposed content addressable file systems and storage system 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.

[6] REFERENCES

[1] A. C. Ltd., "Amazon elastic compute cloud ec2, simple storage service," Amazon, http://aws.amazon.com/ec2/, http://aws.amazon.com/s32/, April 2011.
[2] Microsoft, "Microsoft, windows azure platform," 2010. [Online]. Available: http://www.microsoft.com/windowsazure/
[3] M. Armbrust and A. E. Fox, "Above the clouds: A Berkeley view of cloud computing," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2009-28, Feb 2009.
[4] N. Santos, K. P. Gummadi, and R. Rodrigues, "Towards trusted cloud computing," in Proc. USENIX Hot Cloud 2009.
[5] Z. Wilcox-O'Hearn and B. Warner, "Tahoe: The Least-Authority File system," Proc. Fourth ACM Int'l Workshop Storage Security and Survivability (StorageSS), pp. 21-26, 2008.
[6] H.-Y. Lin and W.-G. Tzeng, "A Secure Decentralized Erasure Code for Distributed Network Storage," IEEE Trans. Parallel and Distributed Systems, vol. 21, no. 11, pp. 1586-1594, Nov. 2010.
[7] C. Dubnicki, L. Gryz, L. Heldt, M. Kaczmarczyk, W. Kilian, P. Strzelczak, J. Szczepkowski, C. Ungureanu, and M. Welnicki,"Hydrastor: A Scalable Secondary Storage," Proc. Seventh Conf. File and Storage Technologies (FAST), pp. 197-210, 2009.
[8] C. Ungureanu, B. Atkin, A. Aranya, S. Gokhale, S. Rago, G. Calkowski, C. Dubnicki, and A. Bohra, "Hydrafs: A High- Throughput File System for the Hydrastor Content-Addressable Storage System," Proc. Eighth USENIX Conf. File and Storage Technologies (FAST), p. 17, 2010.
[9] W. Dong, F. Douglis, K. Li, H. Patterson, S. Reddy, and P. Shilane, "Tradeoffs in Scalable Data Routing for Deduplication Clusters," Proc. Ninth USENIX Conf. File and Storage Technologies (FAST), p. 2,2011.

Published by SK Research Group of Companies (SKRGC).