IJCSEC- International Journal of Computer Science and Engineering Communications.
Vol.3, Issue 2, 2015, Page.889-892, ISSN: 2347–8586

Scalable Identity-Based Distributed Provable Data Possession in Multi-Cloud Storage

A.Keerthana[1] S.JayaPrakash[2]

[1]PG Student [2]H.O.D of CSE,
[1&2] Dept. of Computer Science and Engineering
Idhaya Engineering College for Women
sarakeerthi.arthi@gmail.com, sjpme1981@gmail.com

Abstract: Currently the client stores his data on multi-cloud servers, in which the distributed storage and integrity checking are indispensable. In multi-cloud environment, distributed provable data possession is an important element to secure the remote data. Based on the bilinear pairings, a concrete ID-DPDP protocol is designed. The proposed ID-DPDP protocol is provably secure under the hardness assumption of the standard CDH (Computational Diffie-Hellman) problem. In addition to the structural advantage of elimination of certificate management, proposed ID-DPDP protocol is also efficient and flexible. The proposed scheme is extended to support scalable and efficient public auditing in Cloud Computing. The scheme achieves batch auditing where large data auditing tasks from users can be performed simultaneously by the TPA by splitting them into batches.

***Index Terms**: Cloud computing, Provable data possession, Identity-based cryptography, Distributed computing, Bilinear pairings*

Introduction

Cloud computing has become an important theme in the computer field. Essentially, it takes the information processing as a service, such as storage, computing. It relieves of the burden for storage management, universal data access with independent geographical locations. At the same time, it avoids of capital expenditure on hardware, software, and personnel maintenances, etc. Thus, cloud computing attracts more intention from the enterprise. The foundations of cloud computing lie in the outsourcing of computing tasks to the third party. It entails the security risks in terms of confidentiality, integrity and availability of data and service. The issue to convince the cloud clients that their data are kept intact is especially vital since the clients do not store these data locally. Remote data integrity checking is a primitive to address this issue.For the general case, when the client stores his data on multi-cloud servers, the distributed storage and integrity checking are indispensable. On the other hand, the integrity checking protocol must be efficient in order to make it suitable for capacity-limited end devices. Thus, based on distributed computation, we will study distributed remote data integrity checking model and present the corresponding concrete protocol in multi-cloud storage.

Proposed System

An ID-DPDP protocol comprises four different entities which are Client, CS (Cloud Server), Combiner, and PKG (Private Key Generator). This protocol comprises four procedures: Setup, Extract, TagGen, and Proof. 1. In the phase Extract, PKG creates the private key for the client. 2. The client creates the block-tag pair and uploads it to combiner. The combiner distributes the block-tag pairs to the different cloud servers according to the storage metadata. 3. The verifier sends the challenge to combiner and the combiner distributes the challenge query to the corresponding cloud servers according to the storage metadata. 4. The cloud servers respond the challenge and the combiner aggregates these responses from the cloud servers. The combiner sends the aggregated response to the verifier. Finally, the verifier checks whether the aggregated response is valid. The concrete ID-DPDP construction mainly comes from the signature, provable data possession and distributed computing. The signature relates the client's identity with his private key. Distributed computing is used to store the client's data on multi-cloud servers. At the same time, distributed computing is also used to combine the multi-cloud servers' responses to respond the verifier's challenge. Based on the provable data possession protocol, the ID-DPDP protocol is constructed by making use of the signature and distributed computing.

User key generation

New user obtains the token from Identity Provider by submitting their details. Users send their ID to Private Key Generator (PKG). In the phase Extract, PKG creates the private key for the client. PKG is an entity, when receiving the identity; it outputs the corresponding private key to user. PKG sends the private key $sk_{ID} = (R, \sigma)$ to the client by the secure channel. The client can verify the correctness of the received private key.

Block-tag pair creation and distribution

User has massive data to be stored on the multi-cloud for maintenance and computation, can be either individual consumer or corporation. The client creates the block-tag pair and uploads it to combiner. The combiner distributes the block-tag pairs to the different cloud servers according to the storage metadata. Split the whole file F into n blocks, i.e., F = (F1, F2 … Fn). The client prepares to store the block F_i in the cloud server $CS_{li.}$

Challenge query distribution

The verifier sends the challenge to combiner and the combiner distributes the challenge query to the corresponding cloud servers according to the storage metadata. First-Phase Queries are Extract, Hash, TagGen queries to the challenger C. Extract queries - The adversary A queries the private key of the identity ID. By running Extract (params, mpk, msk, ID), the challenger C gets the private key skID and forwards it to A. Let S1 denote the extracted identity set in the first phase. Hash queries - The adversary A queries hash function adaptively. C responds the *hash* values to A. TagGen queries - The adversary A makes block-tag pair queries adaptively. For a block tag query Fi, the challenger calculates the tag Ti and sends it back to the adversary.

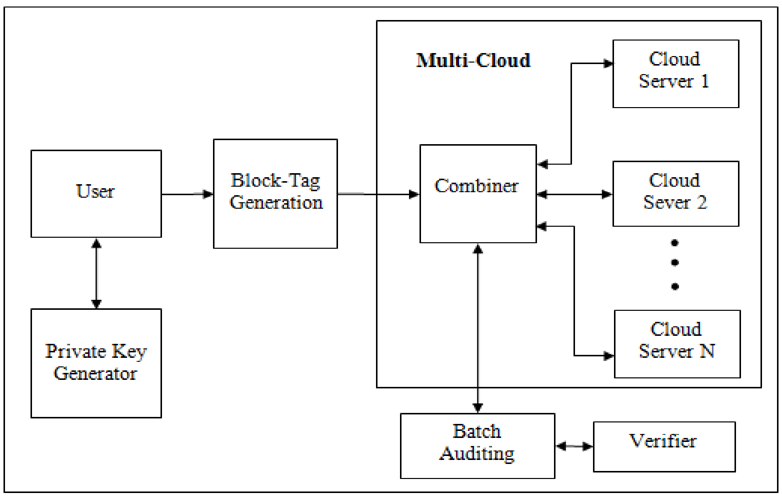
Batch Auditing

Batch auditing is TPA may concurrently handle multiple auditing upon different user's delegation. The individual auditing of these tasks for the TPA can be tedious and very inefficient. For instance, given K auditing delegations on K distinct data files from K different users, it is more advantageous for the TPA to batch these multiple tasks together and audit at one time.

Response verification

The cloud servers respond the challenge and the combiner aggregates these responses from the cloud servers. The combiner sends the aggregated response to the verifier. Finally, the verifier checks whether the aggregated response is valid. Combiner when receives the storage request and distributes the block-tag pairs to the corresponding cloud servers. When receiving the challenge, it splits the challenge and distributes them to the different cloud servers.

Architecture diagram of proposed system



When receiving the responses from the cloud servers, it combines them and sends the combined response to the verifier.

Related work

Susheel George Joseph [1] PDP supports private verifiability, i.e., only the data owner can check data possession. **Yan Zhu [2]** Communication cost is high. **Ayad F. Barsoum [3]** Computation cost is high due to certificate verification. **Qian Wang [4]** Public Auditability does not verify multi-tasks at the same time. **Giuseppe Ateniese [7]** provably secure PDP does not support insert operation. **Francesc Sebe [8]** Remote data possession drastically reduces I/O costs due to without retrieving and downloading of data during verification.

Conclusion

Considering TPA may concurrently handle multiple audit sessions from different users for their outsourced data files, ID-DPDP protocol into a multi-user setting, where the TPA can perform multiple auditing tasks in a batch manner for better efficiency. Extensive analysis shows that the proposed schemes are provably secure and highly efficient.

Auditing System

A public auditing scheme consists of four algorithms (KeyGen, SigGen, GenProof, and VerifyProof).

i) KeyGen is a key generation algorithm that is run by the user to setup the scheme.

ii) SigGen is used by the user to generate verification metadata, which may consist of MAC, signatures, or other related information that will be used for auditing.

iii) GenProof is run by the cloud server to generate a proof of data storage correctness.

iv) VerifyProof is run by the TPA to audit the proof from the cloud server.

Running a public auditing system consists of two phases, Setup and Audit.

a) Setup: The user initializes the public and secret parameters of the system by executing KeyGen, and pre-processes the data file F by using SigGen to generate the verification metadata.

The user stores the data file F and the verification metadata at the cloud server, and deletes its local copy. As part of pre-processing, the user may alter the data file F by expanding it or including additional metadata to be stored at server.

b) Audit: The TPA issues an audit message or challenge to the cloud server to make sure that the cloud server has retained the data file F properly at the time of the audit.

The cloud server will derive a response message from a function of the stored data file F and its verification metadata by executing GenProof. The TPA then verifies the response via VerifyProof.

Future Work

The proposed protocol can be extended to work on secure data sharing. The client can set access control policy for the data’s they are uploading on multi cloud servers. When any other user tries to access the client’s data means they can access only if the particular user has the access rights to that data. These access rights can be designed by the client.

References

[1] Susheel George Joseph, “Co-Operative Multiple Replica Provable Data Possession for Integrity Verification in Multi-Cloud Storage”, International Journal of Engineering And Science, Vol.4, PP 26-31, 2014.

[2] Y. Zhu, G.J. Ahn, H. Hu, S.S. Yau, H.G. An, and S. Chen, “Dynamic Audit Services for Outsourced Storages in Clouds,” IEEE Transactions on Services Computing, 2011.

[3] A. F. Barsoum, and M. A. Hasan, “Integrity Verification of Multiple Data Copies over Untrusted Cloud Servers,” 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012), pp.829-834, 2012.

[4] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, “Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing”, IEEE Transactions on Parallel And Distributed Systems, Vol.22, No.5, pp.847-859, 2011.

[5] Zhen Mo, Yian Zhou, and Shigang Chen, “A Dynamic Proof of Retrievability (PoR) scheme with O(logn) Complexity”, 2012.

[6] A. F. Barsoum, M. A. Hasan, “On Verifying Dynamic Multiple Data Copies over Cloud Servers”, IACR eprint report 447, 2011.

[7] G. Ateniese, R. DiPietro, L. V. Mancini, and G. Tsudik, “Scalable and Efficient Provable Data Possession”, Secure Comm, 2008.

[8] F. Sebe, J. Domingo-Ferrer, A. Martınez-Ballest, Y.Deswarte, and J.Quisquater, “Efficient Remote Data Integrity checking in Critical Information Infrastructures”, IEEE Transactions on Knowledge and Data Engineering, Vol.20, No.8, pp.1-6, 2008.

[9] A. F. Barsoum, M. A. Hasan, “Integrity Verification of Multiple Data Copies over Untrusted Cloud Servers,” 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012), pp. 829-834, 2012.

[10] Y. Zhu, H. Hu, G.J. Ahn, M. Yu, “Cooperative Provable Data Possession for Integrity Verification in Multicloud Storage”, IEEE Transactions on Parallel and Distributed Systems, 23(12), pp. 2231-2244, 2012.