Probable Data Distribution Possession in Multi Cloud storage Using Client Authentication

1 V.ROHINI 2 D.RAJA REDDY

Abstract— Identity-Based Distributed Provable Data Possession in Multi-Cloud Storage Remote data integrity checking is of crucial importance in cloud storage. It can make the clients verify whether their outsourced data is kept intact without downloading the whole data. In some application scenarios, the clients have to store their data on multi-cloud servers. At the same time, the integrity checking protocol must be efficient in order to save the verifier's cost. From the two points, we propose a novel remote data integrity checking model: ID-DPDP (identity-based distributed provable data possession) in multi-cloud storage. The formal system model and security model are given. 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, our ID-DPDP protocol is also efficient and flexible. Based on the client's authorization, the proposed ID-DPDP protocol can realize private verification, delegated verification and public verification.

I. INTRODUCTION

Over the last years, 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.

II. Motivation

We consider an ocean information service corporation Cor in the cloud computing environment. Cor can provide the following services: ocean measurement data, ocean environmentmonitoring data, hydrological data, marine biological data, GIS information, etc. Besides of the above services, Cor has also some private information and some public information, such as the corporation's advertisement. Cor will store these different ocean data on multiple cloud servers. Different cloud service providers have different reputation and charging standard.Of course, these cloud service providers need differentcharges according to the different security-levels. Usually,more secure and more expensive. Thus, Cor will select different cloud service providers to store its different data. For some sensitive ocean data, it will copy these data many times and store these copies on different cloud servers. For the private data, it will store them on the private cloud server.For the public advertisement data, it will store them on the cheap public cloud server. At last, Cor stores its whole dataon the different cloud servers according to their importance and sensitivity. Of course, the storage selection will takeaccount into the Cor's profits and losses. Thus, the distributed cloud storage is indispensable. In multi-cloud environment, distributed provable data possession is an important element to secure the remote data. In PKI (public key infrastructure), provable data possession protocol needs public key certificate distribution and management. It will incur considerable overheads since the verifier will check the certificate when it checks the remote data integrity. In addition to the heavy certificate verification, the system also suffers from the other complicated certificates management such as certificates generation, delivery, revocation, renewals, etc. In cloud computing, most verifiers only have low computation capacity. Identity-based public key cryptography can eliminate the complicated certificate management. In order to increase the efficiency, identity-based provable data possession is more attractive. Thus, it will be very meaningful to study the ID-DPDP.

III Related work

In cloud computing, remote data integrity checking is an important security problem. The clients' massive data is outside his control. The malicious cloud server may corrupt the clients' data in order to gain more benefits. Many researchers proposed the corresponding system model and security model. In 2007,provable data possession (PDP) paradigm was proposed by Ateniese *et al.* [1]. In the PDP model, the verifier can check remote data integrity with a high probability. Based on the RSA, they designed two provably secure PDP schemes. After that, Ateniese *et al*. proposed dynamic PDP model and concrete scheme [2] although it does not support insert operation. In order to support the insert operation, in 2009, Erway *et al.* proposed a full-dynamic PDP scheme based on the authenticated lip table [3]. The similar work has also been done by F.

Seb´e *et al.* [4]. PDP allows a verifier to verify the remote data integrity without retrieving or downloading the whole data. It is a probabilistic proof of possession by sampling random set of blocks from the server, which drastically reduces I/O costs. The verifier only maintains small metadata to perform the integrity checking. PDP is an interesting remote data integrity checking model. In 2012, Wang proposed the security model and concrete scheme of proxy PDP in public clouds [5]. At the same time, Zhu *et al.* proposed the cooperative PDP in the multi-cloud storage [6]. Following Ateniese *et al.*'s pioneering work, many remote data integrity checking models and protocols have been proposed [7], [8], [9], [10], [11], [12]. In 2008, Shacham presented the first proof of retrievability (POR) scheme with provable security [13]. In POR, the verifier can check the remote data integrity and retrieve the remote data at any time. The state of the art can be found in [14]. On some cases, the client may delegate the remote data integrity checking task to the third party. It results in the third party auditing in cloud computing. One of benefits of cloud storage is to enable universal data access with independent geographical locations. This implies that the end devices may be mobile and limited in computation and storage.Efficient integrity checking protocols are more suitable forcloud clients equipped with mobile end devices.

IV SECURITY MODEL OF ID-DPDP

The ID-DPDP system model and security definition are presented in this section. An ID-DPDP protocol comprises four different entities. We describe them below:

1) *Client*: an entity, which has massive data to be stored on the multi-cloud for maintenance and computation, can be either individual consumer or corporation.

2) *CS* (Cloud Server): an entity, which is managed by cloud service provider, has significant storage space and computation resource to maintain the clients' data.

3) *Combiner*: an entity, which 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. When receiving the responses from the cloud servers, it combines them and sends the combined response to the verifier.

4) *PKG* (Private Key Generator): an entity, when receiving the identity, it outputs the corresponding private key.

First, we give the definition of interactive proof system. It will be used in the definition of ID-DPDP. Then, we present the definition and security model of ID-DPDP protocol.

Definition 1 (Interactive Proof System): Let $c, s : N \rightarrow R$ be functions satisfying $c(n) > s(n) + 1p(n)$ for some polynomial $p(\cdot)$. An interactive pair (P, V) is called a interactive proof system for the language L, with completeness bound $c(\cdot)$ and soundness bound $s(\cdot)$, if

1) Completeness: for every $x \in L$, $\Pr[< P, V > (x) = 1] \geq c(|x|)$.

2) Soundness: for every $x\ 6\in L$ and every interactive machine B, $\Pr[< B, V > (x) = 1] \leq s(|x|)$. Interactive proof system is used in the definition of IDDPDP,

i.e., Definition 2.

Definition 2 (ID-DPDP): An ID-DPDP protocol is a collection of three algorithms (Setup, Extract, TagGen) and an interactive proof system (Proof). They are described in detail
below.

1) Setup(1k): Input the security parameter k, it outputs the system public parameters *params*, the master public key *mpk* and the master secret key *msk*.

2) Extract(1k, params,mpk,msk, ID): Input the public parameters params, the master public key mpk, the master secret key msk, and the identity ID of a client, it outputs the private key skID that corresponds to the client with the identity *ID*.

3) TagGen(skID, Fi,P): Input the private key skID, theblock Fi and a set of *CS* P = {CSj}, it outputs the tuple {φi, (Fi, Ti)}, where φi denotes the i-th record of metadata, (Fi, Ti) denotes the i-th block-tag pair. Denote all the metadata {φi} as φ.

4) Proof(P,C(Combiner), V (V erifier)): is a protocol among P, C and V . At the end of the interactive protocol, V outputs a bit {0|1} denoting false or true. Besides of the high efficiency based on the communication and computation overheads, a practical ID-DPDP protocol must satisfy the following security requirements:

1) The verifier can perform the ID-DPDP protocol without the local copy of the file(s) to be checked.

2) If some challenged block-tag pairs are modified or lost, the response can not pass the ID-DPDP protocol even if P and C collude. To capture the above security requirements, we define the security of an ID-DPDP protocol as follows.

Definition 3 (Unforgeability): An ID-DPDP protocol is unforgeable if for any (probabilistic polynomial) adversary A (malicious CS and combiner) the probability that A wins the ID-DPDP game on a set of file blocks is negligible. The IDDPDP game between the adversary A and the challenger C can be described as follows:

1) Setup: The challenger C runs Setup(1k) and gets (params,mpk,msk). It sends the public parameters and master public key (params,mpk) to A while it keeps confidential the master secret key msk.

2) First-Phase Queries: The adversary A adaptively makes Extract, Hash, TagGen queries to the challenger C as follows:

• 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 firstphase.

• 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. Let (Fi, Ti) be the queried block-tag pair for index i ∈ I1, where I1 is a set of indices that the corresponding block tags have been queried in the first-phase.

3) Challenge: C generates a challenge chal which defines a ordered collection {ID∗, i1,

i2, · · · , ic}, where ID∗ 6∈ S1, {i1, i2, · · · , ic} * I1, and c is a positive integer. The adversary is required to provide the data possession proof for the blocks Fi1 , · · · , Fic .

4) Second-Phase Queries: Similar to the First-Phase Queries. Let the Extract query identity set be S2 and the TagGen query index set be I2. The restriction is that {i1, i2, · · · , ic} * (I1 ∪ I2) and ID∗ 6∈ (S1 ∪ S2).

5) Forge: The adversary A responses θ for the challenge chal.

V CONCLUSION

In multi-cloud storage, this paper formalizes the ID-DPDP system model and security model. At the same time, we propose the first ID-DPDP protocol which is provably secure under the assumption that the CDH problem is hard. Besides of the elimination of certificate management, our ID-DPDP protocol has also flexibility and high efficiency. At the same time, the proposed ID-DPDP protocol can realize private verification, delegated verification and public verification based on the client's authorization.

REFERENCES

[1] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, D. Song, "Provable Data Possession at Untrusted Stores", *CCS'07*, pp. 598-609, 2007.

[2] G. Ateniese, R. DiPietro, L. V. Mancini, G. Tsudik, "Scalable and Efficient Provable Data Possession", *SecureComm 2008*, 2008.

[3] C. C. Erway, A. Kupcu, C. Papamanthou, R. Tamassia, "Dynamic Provable Data Possession", *CCS'09*, pp. 213-222, 2009.

[4] F. Seb´e, J. Domingo-Ferrer, A. Mart´ınez-Ballest´e, Y. Deswarte, J. Quisquater, "Efficient Remote Data Integrity checking in Critical Information Infrastructures", *IEEE Transactions on Knowledge and Data Engineering*, 20(8), pp. 1-6, 2008.

[5] H.Q. Wang, "Proxy Provable Data Possession in Public Clouds," *IEEE Transactions on Services Computing*, 2012. http://doi.ieeecomputersociety.org/10.1109/TSC.2012.35

[6] 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.

[7] Y. Zhu, H. Wang, Z. Hu, G. J. Ahn, H. Hu, S. S. Yau, "Efficient Provable Data Possession for Hybrid Clouds", *CCS'10*, pp. 756-758, 2010.

[8] R. Curtmola, O. Khan, R. Burns, G. Ateniese, "MR-PDP: Multiple- Replica Provable Data Possession", *ICDCS'08*, pp. 411-420, 2008.

[9] A. F. Barsoum, M. A. Hasan, "Provable Possession and Replication of Data over Cloud Servers", CACR, University of Waterloo, Report2010/32,2010. Available at

http://www.cacr.math.uwaterloo.ca/techreports/2010/cacr2010-32.pdf.

[10] Z. Hao, N. Yu, "A Multiple-Replica Remote Data Possession Checking Protocol with Public Verifiability", *2010 Second International Symposium on Data, Privacy, and E-Commerce*, pp. 84-89, 2010.

[11] A. F. Barsoum, M. A. Hasan, "On Verifying Dynamic Multiple Data Copies over Cloud Servers", *IACR eprint report 447, 2011*. Available at http://eprint.iacr.org/2011/447.pdf.

[12] A. Juels, B. S. Kaliski Jr., "PORs: Proofs of Retrievability for Large Files", *CCS'07*, pp. 584-597, 2007.

[13] H. Shacham, B. Waters, "Compact Proofs of Retrievability", *ASIACRYPT 2008*, LNCS 5350, pp. 90-107, 2008.

[14] K. D. Bowers, A. Juels, A. Oprea, "Proofs of Retrievability: Theory and Implementation", *CCSW'09*, pp. 43-54, 2009.

[15] Q. Zheng, S. Xu. Fair and Dynamic Proofs of Retrievability. *CODASPY' 11*, pp. 237-248, 2011.

[16] Y. Dodis, S. Vadhan, D. Wichs, "Proofs of Retrievability via Hardness Amplification", *TCC 2009*, LNCS 5444, pp. 109-127, 2009.

[17] Y. Zhu, H. Wang, Z. Hu, G. J. Ahn, H. Hu, "Zero-Knowledge Proofs of Retrievability", *Sci China Inf Sci*, 54(8), pp. 1608-1617, 2011.

[18] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing", *INFOCOM 2010*, IEEE, March 2010.

[19] Q. Wang, C. Wang, K. Ren, W. Lou, J. Li, "Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing", *IEEE Transactions on Parallel And Distributed Systems* , 22(5), pp. 847-859, 2011.

[20] C. Wang, Q. Wang, K. Ren, N. Cao, W. Lou, "Toward Secure and Dependable Storage Services in Cloud Computing," *IEEE Transactionson Services Computing*, 5(2), pp. 220-232, 2012.

AUTHOR'SPROFILE:

1.V.ROHINI

valigatlarohini94@gmail.com

2.D.RAJA REDDY

raajaareddy@gmail.com

ASSISTANT PROFESSOR