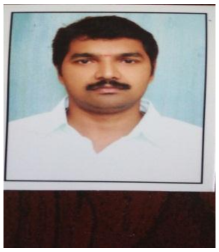
Cloud Outsourced Key with Identity Based Encryption



1.KOMARAM PRUDVI RAJ **2.C MADAN KUMAR**

1.PG Scholar, Department of CSE, Vaagdevi College of Engineering, Bollikunta, Warangal, Telangana,

2. Assistant Professor, Department of CSE, Vaagdevi College of Engineering, Bollikunta, Warangal, Telangana,

ABSTRACT

Remote data integrity checking is of crucial importancein cloud storage. It can make the clients verify whethertheir outsourced data is kept intact without downloading thewhole data. In some application scenarios, the clients have tostore their data on multi-cloud servers. At the same time, theintegrity checking protocol must be efficient in order to savethe verifier’s cost. From the two points, we propose a novelremote data integrity checking model: ID-DPDP (identity-baseddistributed provable data possession) in multi-cloud storage. Theformal system model and security model are given. Based onthe bilinear pairings, a concrete ID-DPDP protocol is designed.The proposed ID-DPDP protocol is provably secure under thehardness assumption of the standard CDH (computational Diffie-Hellman) problem. In addition to the structural advantage ofelimination of certificate management, our ID-DPDP protocol isalso 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 importanttheme in the computer field. Essentially, it takes the informationprocessing as a service, such as storage, computing. Itrelieves of the burden for storage management, universal dataaccess with independent geographical locations. At the sametime, it avoids of capital expenditure on hardware, software,and personnel maintenances, etc. Thus, cloud computing attractsmore intention from the enterprise.The foundations of cloud computing lie in the outsourcingof computing tasks to the third party. It entails the securityrisks in terms of confidentiality, integrity and availability ofdata and service. The issue to convince the cloud clients thattheir data are kept intact is especially vital since the clients donot 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, thedistributed storage and integrity checking are indispensable.On the other hand, the integrity checking protocol must beefficient in order to make it suitable for capacity-limited enddevices. Thus, based on distributed computation, we will studydistributed remote data integrity checking model and presentthe corresponding concrete protocol in multi-cloud storage.

II. Motivation

We consider an ocean information service corporation Corin the cloud computing environment. Cor can provide the followingservices: ocean measurement data, ocean environmentmonitoringdata, hydrological data, marine biological data,GIS information, etc. Besides of the above services, Cor hasalso some private information and some public information,such as the corporation's advertisement. Cor will store thesedifferent ocean data on multiple cloud servers. Different cloudservice 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 selectdifferent cloud service providers to store its different data.For some sensitive ocean data, it will copy these data manytimes and store these copies on different cloud servers. Forthe private data, it will store them on the private cloud server.For the public advertisement data, it will store them on thecheap public cloud server. At last, Cor stores its whole dataon the different cloud servers according to their importanceand sensitivity. Of course, the storage selection will takeaccount into the Cor's profits and losses. Thus, the distributedcloud storage is indispensable. In multi-cloud environment,distributed provable data possession is an important elementto secure the remote data.In PKI (public key infrastructure), provable data possessionprotocol needs public key certificate distribution andmanagement. It will incur considerable overheads since theverifier will check the certificate when it checks the remotedata integrity. In addition to the heavy certificate verification,the system also suffers from the other complicated certificatesmanagement such as certificates generation, delivery,revocation, renewals, etc. In cloud computing, most verifiersonly have low computation capacity. Identity-based publickey cryptography can eliminate the complicated certificatemanagement. In order to increase the efficiency, identity-basedprovable data possession is more attractive. Thus, it will bevery meaningful to study the ID-DPDP.

III Related work

In cloud computing, remote data integrity checking is an importantsecurity problem. The clients' massive data is outsidehis control. The malicious cloud server may corrupt the clients'data in order to gain more benefits. Many researchers proposedthe corresponding system model and security model. In 2007,provable data possession (PDP) paradigm was proposed byAteniese*et al.* [1]. In the PDP model, the verifier can checkremote data integrity with a high probability. Based on theRSA, they designed two provably secure PDP schemes. Afterthat, Ateniese*et al.* proposed dynamic PDP model and concretescheme [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 authenticatedlip table [3]. The similar work has also been done by F.

Seb´e*et al.* [4]. PDP allows a verifier to verify the remote dataintegrity without retrieving or downloading the whole data. Itis 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 theintegrity checking. PDP is an interesting remote data integritychecking model. In 2012, Wang proposed the security modeland concrete scheme of proxy PDP in public clouds [5]. Atthe same time, Zhu *et al.* proposed the cooperative PDP in themulti-cloud storage [6].Following Ateniese*et al.*'s pioneering work, many remotedata integrity checking models and protocols have been proposed[7], [8], [9], [10], [11], [12]. In 2008, Shachampresented the first proof of retrievability (POR) scheme withprovable security [13]. In POR, the verifier can check theremote data integrity and retrieve the remote data at any time.The state of the art can be found in [14]. Onsome cases, the client may delegate the remote data integritychecking task to the third party. It results in the third partyauditing in cloud computing. One ofbenefits of cloud storage is to enable universal data access withindependent geographical locations. This implies that the enddevices may be mobile and limited in computation and storage.Efficient integrity checking protocols are more suitable forcloud clients equipped with mobile end devices.

IVSECURITY MODEL OF ID-DPDP

The ID-DPDP system model and security definition arepresented in this section. An ID-DPDP protocol comprisesfour different entities. Wedescribe them below:

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

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

3) *Combiner*: an entity, which receives the storage requestand distributes the block-tag pairs to the correspondingcloud servers. When receiving the challenge, it splitsthe challenge and distributes them to the different cloudservers. When receiving the responses from the cloudservers, it combines them and sends the combinedresponse to the verifier.

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

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

Definition 1 (Interactive Proof System):

Let $c, s : N \rightarrow R$ be functions satisfying $c(n) > s(n) + 1p(n)$ forsome polynomial $p(\cdot)$. An interactive pair (P, V) is called ainteractive proof system for the language L, with completenessbound $c(\cdot)$ 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 interactivemachine 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 collectionof three algorithms (Setup, Extract, TagGen) and aninteractive proof system (Proof). They are described in detail

below.

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

2) Extract(1k, params,mpk,msk, ID): Input the publicparameters params, the master public key mpk, themaster secret key msk, and the identity ID of a client,it outputs the private key skIDthat corresponds to theclient 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 thetuple {φi, (Fi, Ti)}, where φidenotes the i-th record ofmetadata, (Fi, Ti) denotes the i-th block-tag pair. Denoteall the metadata {φi} as φ.

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

1) The verifier can perform the ID-DPDP protocol withoutthe 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 ifP and C collude.To capture the above security requirements, we define thesecurity of an ID-DPDP protocol as follows.

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

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

2) First-Phase Queries: The adversary A adaptively makesExtract, Hash, TagGen queries to the challenger C asfollows:

- Extract queries. The adversary A queries theprivate key of the identity ID. By runningExtract(params,mpk,msk, ID), the challenger Cgets the private key skIDand forwards it to A.

Let S1 denote the extracted identity set in the firstphase.

- Hash queries. The adversary A queries *hash* functionadaptively. C responds the *hash* values to A.
- TagGen queries. The adversary A makes block-tagpair queries adaptively. For a block tag query Fi, thechallenger calculates the tag Ti and sends it back tothe adversary. Let (Fi, Ti) be the queried block-tagpair for index i∈

I1, where I1 is a set of indicesthat the corresponding block tags have been queriedin the first-phase.

3) Challenge: C generates a challenge chal which definesa 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 possessionproof for the blocks Fi1 , · · · , Fic .

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

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

V CONCLUSION

In multi-cloud storage, this paper formalizes the ID-DPDPsystem model and security model. At the same time, wepropose the first ID-DPDP protocol which is provably secureunder the assumption that the CDH problem is hard. Besidesof the elimination of certificate management, our ID-DPDPprotocol has also flexibility and high efficiency. At the sametime, the proposed ID-DPDP protocol can realize private verification,delegated verification and public verification basedon 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 andEfficient Provable Data Possession", *SecureComm 2008*, 2008.

[3] C. C. Erway, A. Kupcu, C. Papamanthou, R. Tamassia, "DynamicProvable 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 InformationInfrastructures", *IEEE Transactions on Knowledge and DataEngineering*, 20(8), pp. 1-6, 2008.

[5] H.Q. Wang, "Proxy Provable Data Possession in PublicClouds," *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 Possessionfor Integrity Verification in Multicloud Storage", *IEEE Transactions onParallel 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 ProvableData 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 Possessionand Replication of Data over Cloud Servers", CACR,University of Waterloo, Report2010/32,2010. Available athttp://www.cacr.math.uwaterloo.ca/techre ports /2010/cacr2010-32.pdf.

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

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

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

[13] H. Shacham, B. Waters, "Compact Proofs of Retrievability", *ASIACRYPT2008*, LNCS 5350, pp. 90-107, 2008.

[14] K. D. Bowers, A. Juels, A. Oprea, "Proofs of Retrievability: Theory andImplementation", *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 HardnessAmplification", *TCC 2009*, LNCS 5444, pp. 109-127, 2009.

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

[18] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving PublicAuditing for Data Storage Security in Cloud Computing", *INFOCOM2010*, IEEE, March 2010.

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

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