International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 24 Issue 9 – JUNE 2017.

Efficient Technique for Outsourcing and Dynamic Data Operations in Cloud

Shaik Jubedabi[#1], P.Raghuveer[*2] and Prof.V.Suryanarayana[*3]

[#] *STUDENT, DEPT OF C.S.E, NRI INSTITUTE OF TECHNOLOGY,AGIRIPAALI, A.P, INDIA*

[*2] *Asst. Prof., DEPT OF C.S.E, NRI INSTITUTE OF TECHNOLOGY,AGIRIPAALI, A.P, INDIA*

[*3]*HOD, DEPT OF C.S.E, NRI INSTITUTE OF TECHNOLOGY,AGIRIPAALI, A.P, INDIA*

***Abstract*— Increasingly more and more organizations are opting for outsourcing data to remote cloud service providers (CSPs). Customers can rent the CSPs storage infrastructure to store and retrieve almost unlimited amount of data by paying fees metered in gigabyte/month. For an increased level of scalability, availability, and durability, some customers may want their data to be replicated on multiple servers across multiple data centers. The more copies the CSP is asked to store, the more fees the customers are charged. Therefore, customers need to have a strong guarantee that the CSP is storing all data copies that are agreed upon in the service contract, and all these copies are consistent with the most recent modifications issued by the customers. In this paper, we propose a map-based provable multicopy dynamic data possession (MB-PMDDP) scheme that has the following features: 1) it provides an evidence to the customers that the CSP is not cheating by storing fewer copies; 2) itsupports outsourcing of dynamic data, i.e., it supports block-level operations, such as block modification, insertion, deletion, and append; and 3) it allows authorized users to seamlessly access the file copies stored by the CSP. We give a comparative analysis of the proposed MB-PMDDP scheme with a reference model obtained by extending existing provable possession of dynamic single-copy schemes. The theoretical analysis is validated through experimental results on a commercial cloud platform. In addition, we show the security against colluding servers, and discuss how to identify corrupted copies by slightly modifying the proposed scheme.**

the data file in its original form, it needs to correctly compute a response to a challenge vector sent from a verifier — who can be the original data owner or a trusted entity that shares some information with the owner.

One of the core design principles of outsourcing data is to provide dynamic behavior of data for various applications. This means that the remotely stored data can be not only accessed by the authorized users, but also updated and scaled (through block level operations) by the data owner. PDP schemes focus on only static or warehoused data, where the outsourced data is kept unchanged over remote servers.

***Index Terms*—Cloud Computing, mulicopy dynamic data possession, CSP**

I. Introduction

OUTSOURCING data to a remote cloud service provider (CSP) allows organizations to store more data on the CSP than on private computer systems. Such outsourcing of data storage enables organizations to concentrate on innovations and relieves the burden of constant server updates and other computing issues. Moreover, many authorized users can access the remotely stored data from different geographic locations making it more convenient for them.

Once the data has been outsourced to a remote CSP which may not be trustworthy, the data owners lose the direct control over their sensitive data. This lack of control raises new formidable and challenging tasks related to data confidentiality and integrity protection in cloud computing. The confidentiality issue can be handled by encrypting sensitive data before outsourcing to remote servers. As such, it is a crucial demand of customers to have a strong evidence that the cloud servers still possess their data and it is not being tampered with or partially deleted over time. Consequently, many researchers have focused on the problem of provable data possession (PDP) and proposed different schemes to audit the data stored on remote servers.

PDP is a technique for validating data integrity over remote servers. In a typical PDP model, the data owner generates some metadata/information for a data file to be used later for verification purposes through a challenge-response protocol with the remote/cloud server. The owner sends the file to be stored on a remote server which may be untrusted, and deletes the local copy of the file. As a proof that the server is still possessing

II. Main Contributions:

Our contributions can be summarized as follows:

• We propose a map-based provable multi-copy dynamic data possession (MB-PMDDP) scheme. This scheme provides an adequate guarantee that the CSP stores all copies that are agreed upon in the service contract. Moreover, the scheme supports outsourcing of dynamic data, i.e., it supports block-level operations such as block modification, insertion, deletion, and append. The authorized users, who have the right to access the owner's file, can seamlessly access the copies received from the CSP.

• We give a thorough comparison of MB-PMDDP with a reference scheme, which one can obtain by extending existing PDP models for dynamic single-copy data. We also report our implementation and experiments using Amazon cloud platform.

• We show the security of our scheme against colluding servers, and discuss a slight modification of the proposed scheme to identify corrupted copies.

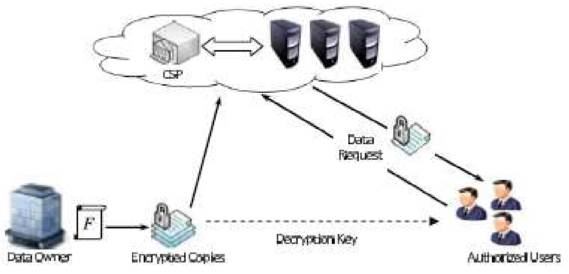

Fig. 1:. Cloud computing data storage system model.

III. OUR SYSTEM AND ASSUMPTIONS

A. *System Components:*

The cloud computing storage model considered in this work consists of three main components as:

(i) a **data owner** that can be an organization originally possessing sensitive data to be stored in the cloud;

(ii) a **CSP** who manages cloud servers (CSs) and provides paid storage space on its infrastructure to store the owner's files; and

(iii) **authorized users** — a set of owner's clients who have the right to access the remote data.

B. *Outsourcing, Updating, and Accessing*

The data owner has a fileF consisting of m blocks and the CSP offers to store n copies { F1, F2,..., Fn} of the owner's file on different servers — to prevent simultaneous failure of all copies — in exchange of pre-specified fees metered in GB/month. The number of copies depends on the nature of data; more copies are

needed for critical data that cannot easily be reproduced, and to achieve a higher level of scalability. This critical data should be replicated on multiple servers across multiple data centers. On the other hand, non-critical, reproducible data are stored at reduced levels of redundancy. The CSP pricing model is related to the number of data copies.

For data confidentiality, the owner encrypts his data before outsourcing to CSP. After outsourcing all n copies of the file, the owner may interact with the CSP to perform block-level operations on all copies. These operations includes modify, insert, append, and delete specific blocks of the outsourced data copies. An authorized user of the outsourced data sends a dataaccess request to the CSP and receives a file copy in an encryptedform that can be decrypted using a secret key shared with the owner. According to the load balancing mechanism used by the CSP to organize the work of the servers, the data-access request is directed to the server with the lowest congestion, and thus the user is not aware of which copy has been received.

C. *Underlying Algorithms*

The proposed scheme consists of seven polynomial time algorithms:

KeyGen, CopyGen, TagGen, PrepareUpdate, ExecUpdate, Prove, and Verify.

The data owner runs the algorithms KeyGen, CopyGen, TagGen, and PrepareUpdate.

The CSP runs the algorithms ExecUpdate and Prove, while a verifier runs the Verify algorithm. –

(pk,sk)← KeyGen(). This algorithm is run by the data owner to generate a public key pk and a private key sk. The private key sk is kept secret by the owner, while pk is publicly known.

F ← CopyGen(CN i, F)1≤i≤n. This algorithm is run by the data owner. It takes as input a copy number CN i and a file F, and generates n copies F ={ Fi}1≤i≤n. Theowner sends the copies F to the CSP to be stored on cloud servers.

← TagGen(sk, F). This algorithm is run by the data owner. It takes as input the private key sk and the file copies F, and outputs tags/authenticators set , which is an ordered collection of tags for the data blocks. The owner sends to the CSP to be stored along with the copies F. − (D, UpdateReq)

← PrepareUpdate(D, UpdateInfo). This algorithm is run by the data owner to update the outsourced file copies stored by the remote CSP. The input parameters are a previous metadata D stored on the owner side, and some information UpdateInfo about the dynamicoperationto be performedon a specific block. The outputs of this algorithm are a modified metadata D and an update request UpdateReq. This request may contain a modified version of a previously stored block, a new block to be inserted, or a delete command to delete a specific block from the file copies. UpdateReq also contains updated (or new) tags for modified (or inserted/appended) blocks, and it is sent from the data owner to the CSP in order to perform the requested update.

← ExecUpdate(F, , UpdateReq). This algorithm is run by the CSP, where the input parameters are the file copies F, the tags set , and the request UpdateReq. It outputs an updated version of the file copies F along with an updated tags set. The latter does not require the private key to be generated; just replacement/insertion/deletion of one item of by a new item sent from the owner.

− P ← Prove(F,,chal). This algorithm is run by the CSP. It takes as input the file copies F, the tags set , and a challenge chal (sent from a verifier). It returns a proof P which guarantees that the CSP is actually storing n copies and all these copies are intact, updated, and consistent.

{ 1,0}←Verify(pk,P,D). This algorithm is run by a verifier (original owner or any other trusted auditor). It takes as input the public key pk, the proof P

returned from the CSP, and the mostrecent metadata D. The output is 1 if the integrity of all file copies is correctly verified or 0 otherwise.

IV. PROPOSED MB-PMDDP SCHEME

A. *Overview and Rationale:*

Generating unique differentiable copies of the data file is the core to design a provable multi-copy data possession scheme. Identical copies enable the CSP to simply deceive the owner by storing only one copy and pretending that it stores multiple copies. Using a simple yet efficient way, the proposed scheme generates distinct copies utilizing the diffusion property of any secure encryption scheme. The diffusion property ensures that the output bits of the ciphertext depend on the input bits of the plaintext in a very complex way, i.e., there will be an unpredictable complete change in the ciphertext, if there is a single bit change in the plaintext.

The interaction between the authorized users and the CSP is considered through this methodology of generating distinct copies, where the former can decrypt/access a file copy received from the CSP. In the proposed scheme, the authorized users need only to keep a single secret key (shared with the data owner) to decrypt the file copy, and it is not necessarily to recognize the index of the received copy.

we propose a MB-PMDDP scheme allowing the data owner to update and scale the blocks of file copies outsourced to cloud servers which may be untrusted. Validating such copies of dynamic data requires the knowledge of the block versions to ensure that the data blocks in all copies are consistent with the most recent modifications issued by the owner. Moreover, the verifier should be aware of the block indices to guarantee that the CSP has inserted or added the new blocks at the requested positions in all copies. To this end, the proposed scheme is based on using a small data structure (metadata), which we call a map-version table.

B. *Map-Version Table:*

The map-version table (MVT) is a small dynamic data structure stored on the verifier side to validate the integrity and consistency of all file copies outsourced to the CSP. The MVT consists of three columns: serial number (SN), block number (BN), and block version (BV).

The SN is an indexing to the file blocks. It indicates the physical position of a block in a data file. The BN is a counter used to make a logical numbering/indexing to the file blocks. Thus, the relation between BN and SN can be viewed as a mapping between the logicalnumber BN and the physical position SN. The BV indicates the current version of file blocks. When a data file is initially created the BV of each block is 1. If a specific block is being updated, its BV is incremented by 1.

SN	*BN*	*BV*
1	1	1
2	2	1
3	3	1
4	4	1
5	5	1
6	6	1
7	7	1
8	8	1
Initially		

SN	BN	BV
1	1	1
2	2	1
3	3	1
4	9	1
5	4	1
6	5	2
7	6	1
8	7	1
9	8	1
Insert block after position 3		

SN	BN	BV
1	1	1
2	2	1
3	3	1
4	4	1
5	5	2
6	6	1
7	7	1
8	8	1
Modifying block at position 5		

C. *Notations:*

− F is a data file to be outsourced, and is composed of a sequence of m blocks, i.e., F ={ b1,b2,...,bm}.

− πkey(·) is a pseudo-random permutation (PRP) : key × {0,1}log2(m) →{ 0,1}log2(m).1

− ψkey(·) is a pseudo-random function (PRF): key × {0,1}∗ → Zp (p is a large prime).

− Bilinear Map/Pairing:

Let G1, G2, andGT be cyclic groups of prime order p. Letg1 and g2 be generators of G1 and G2, respectively.

A bilinear pairing is a map ˆ e : G1×G2 → GT with the properties :

1) Bilinear: ˆ e(ua,vb) =ˆ e(u,v)ab ∀ u ∈ G1,v ∈ G2,and a,b∈ Zp

2) Non-Degenerate: ˆ e(g1,g2)=1

3) Computable: there exists an efficient algorithm for computing ˆ e

− H(·) is a map-to-point hash function : {0,1}∗ → G1.

− EK is an encryption algorithm with strong diffusion property,

V. REFERENCE MODEL AND PERFORMANCE ANALYSIS

A. *Reference Model*

It is possible to obtain a provable multi-copy dynamic data possession scheme by extending existing PDP models for single-copy dynamic data. Such PDP schemes selected for extension must meet the following conditions:

(i) support of full dynamic operations (modify, insert, append, and delete),

(ii) (support of public verifiability,

(iii) based on pairing cryptography in creating block tags (homomorphic authenticators); and

(iv) block tags are outsourced along with data blocks to the CSP

Meeting these conditions allows us to construct a PDP reference model that has similar features to the proposed MB-PMDDP scheme. Therefore, we can establish a fair comparison between the two schemes and evaluate the performance of our proposed approach.

B. *Implementation:*

We have implemented the proposed MB-PMDDP scheme and the TB-PMDDP reference model on top of Amazon Elastic Compute Cloud (Amazon EC2) [30] and Amazon Simple Storage Service (Amazon S3) [31] cloud platforms. Through Amazon EC2 customers can lunch and manage Linux/Unix/Windows server instances (virtual servers) in Amazon's infrastructure. The number of EC2 instances can be automatically scaled up and down according to customers' needs. Amazon S3 is a web storage service to store and retrieve almost unlimited amount of data. Moreover, it enables customers to specify geographic locations for storing their data.

Our implementation of the presented schemes consists of three modules:

OModule (owner module), CModule (CSP module), and VModule (verifier module).

OModule, which runs on the owner side, is a library that includes KeyGen, CopyGen, TagGen, andPrepareUpdate algorithms.

CModule is a library that runs on Amazon EC2 and includes ExecuteUpdate and Prove algorithms.

VModule is a library to be run at the verifier side and includes the Verify algorithm.

In the experiments, we do not consider the system pre-processing time to prepare the different file copies and generate the tags set. This pre-processing is done only once during the life time of the system which may be for tens of years. Moreover, in the implementation we do not consider the time to access the file blocks, as the state-of-the-art hard drive technology allows as much as 1MB to be read in just few nanoseconds . Hence, the total access time is unlikely to have substantial impact on the overall system performance.

C. *Applications:*

- E-Health: Where the patient's database that contains large and sensitive information can be stored on the cloud servers.In this type of applications , the e-health organization can be considered as the data owner, and the physicians as the authorized users who have the right to access the patient's medical history.
- Financial
- Scientific Educational etc…

VI. CONCLUSION:

Outsourcing data to remote servers has become a growing trend for many organizations to alleviate the burden of local data storage and maintenance.

We have proposed a new PDP scheme (referred to as MB-PMDDP), which supports outsourcing of multi-copy dynamic data, where the data owner is capable of not only archiving and accessing the data copies stored by the CSP, but also updating and scaling these copies on the remote servers

The interaction between the authorized users and the CSP is considered in our scheme, where the authorized users can seamlessly access a data copy received from the CSP using a single secret key shared with the data owner.

A slight modification can be done on the proposed scheme to support the feature of identifying the indices of corrupted copies. The corrupted data copy can be reconstructed even from a complete damage using duplicated copies on other servers. Through security analysis, we have shown that the proposed scheme is provably secure.

REFERENCES

[1] G. Ateniese et al., "Provable data possession at untrusted stores," in Proc. 14th ACM Conf. Comput. Commun. Secur. (CCS), New York, NY, USA, 2007, pp. 598–609.
[2] K. Zeng, "Publicly verifiable remote data integrity," in Proc. 10th Int. Conf. Inf. Commun. Secur. (ICICS), 2008, pp. 419–434.
[3] Y. Deswarte, J.-J. Quisquater, and A. Saïdane, "Remote integrity checking," in Proc. 6th Working Conf. Integr. Internal Control Inf. Syst. (IICIS), 2003, pp. 1–11.
[4] D. L. G. Filho and P. S. L. M. Barreto, "Demonstrating data possession and uncheatable data transfer," IACR (International Association for Cryptologic Research) ePrint Archive, Tech. Rep. 2006/150, 2006.
[5] F. Sebé, J. Domingo-Ferrer, A. Martinez-Balleste, Y. Deswarte, and J.-J. Quisquater, "Efficient remote data possession checking in critical information infrastructures," IEEE Trans. Knowl. Data Eng., vol. 20, no. 8, pp. 1034–1038, Aug. 2008.
[6] P. Golle, S. Jarecki, and I. Mironov, "Cryptographic primitives enforcing communication and storage complexity," in Proc. 6th Int. Conf. Financial Cryptograph. (FC), Berlin, Germany, 2003, pp. 120–135.
[7] M. A. Shah, M. Baker, J. C. Mogul, and R. Swaminathan, "Auditing to keep online storage services honest," in Proc. 11th USENIX Workshop Hot Topics Oper. Syst. (HOTOS), Berkeley, CA, USA, 2007, pp. 1–6.
[8] M. A. Shah, R. Swaminathan, and M. Baker, "Privacy-preserving audit and extraction of digital contents," IACR Cryptology ePrint Archive, Tech. Rep. 2008/186, 2008.
[9] E. Mykletun, M. Narasimha, and G. Tsudik, "Authentication and integrity in outsourced databases," ACM Trans. Storage, vol. 2, no. 2, pp. 107–138, 2006.
[10] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in Proc. 4th Int. Conf. Secur. Privacy Commun. Netw. (SecureComm), New York, NY, USA, 2008, Art. ID 9.
[11] C. Wang, Q. Wang, K. Ren, and W. Lou. (2009). "Ensuring data storage security in cloud computing," IACR Cryptology ePrint Archive, Tech. Rep. 2009/081. [Online]. Available: http://eprint.iacr.org/
[12] C. Erway, A. Küpçü, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in Proc. 16th ACM Conf. Comput. Commun. Secur. (CCS), New York, NY, USA, 2009, pp. 213–222.
[13] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," in Proc. 14th Eur. Symp. Res. Comput. Secur. (ESORICS), Berlin, Germany, 2009, pp. 355–370.
[14] Z. Hao, S. Zhong, and N. Yu, "A privacy-preserving remote data integrity checking protocol with data dynamics and public verifiability," IEEE Trans. Knowl. Data Eng., vol. 23, no. 9, pp. 1432–1437, Sep. 2011.
[15] A. F. Barsoum and M. A. Hasan. (2010). "Provable possession and replication of data over cloud servers," Centre Appl. Cryptograph. Res., Univ. Waterloo, Waterloo, ON, USA, Tech. Rep. 2010/32. [Online]. Available: http://www.cacr.math.uwaterloo.ca/techreports/2010/ cacr2010-32.pdf
[16] R. Curtmola, O. Khan, R. Burns, and G. Ateniese, "MR-PDP: Multiplereplica provable data possession," in Proc. 28th IEEE ICDCS, Jun. 2008, pp. 411–420.
[17] Z. Hao and N. Yu, "A multiple-replica remote data possession checking protocol with public verifiability," in Proc. 2nd Int. Symp. Data, Privacy, E-Commerce, Sep. 2010, pp. 84–89.

[18] H. Shacham and B. Waters, "Compact proofs of retrievability," in Proc. 14th Int. Conf. Theory Appl. Cryptol. Inf. Secur., 2008, pp. 90–107.
[19] A. Juels and B. S. Kaliski, Jr., "Pors: Proofs of retrievability for large files," in Proc. 14th ACM Conf. Comput. Commun. Secur. (CCS), 2007, pp. 584–597.
[20] R. Curtmola, O. Khan, and R. Burns, "Robust remote data checking," in Proc. 4th ACM Int. Workshop Storage Secur. Survivability, 2008, pp. 63–68.
[21] K. D. Bowers, A. Juels, and A. Oprea, "Proofs of retrievability: Theory and implementation," in Proc. ACM Workshop Cloud Comput. Secur. (CCSW), 2009, pp. 43–54.
[22] Y. Dodis, S. Vadhan, and D. Wichs, "Proofs of retrievability via hardness amplification," in Proc. 6th Theory Cryptograph. Conf. (TCC), 2009, pp. 109–127.
[23] K. D. Bowers, A. Juels, and A. Oprea, "HAIL: A high-availability and integrity layer for cloud storage," in Proc. 16th ACM Conf. Comput. Commun. Secur. (CCS), New York, NY, USA, 2009, pp. 187–198.
[24] C. E. Shannon, "Communication theory of secrecy systems," Bell Syst. Tech. J., vol. 28, no. 4, pp. 656–715, 1949.
[25] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing," in Proc. 7th Int. Conf. Theory Appl. Cryptol. Inf. Secur. (ASIACRYPT), London, U.K., 2001, pp. 514–532.
[26] G. Ateniese, S. Kamara, and J. Katz, "Proofs of storage from homomorphic identification protocols," in Proc. 15th Int. Conf. Theory Appl. Cryptol. Inf. Secur. (ASIACRYPT), Berlin, Germany, 2009, pp. 319–333.
[27] R. C. Merkle, "Protocols for public key cryptosystems," in Proc. IEEE Symp. Secur. Privacy, Apr. 1980, p. 122.
[28] C. Martel, G. Nuckolls, P. Devanbu, M. Gertz, A. Kwong, and S. G. Stubblebine, "A general model for authenticated data structures," Algorithmica, vol. 39, no. 1, pp. 21–41, Jan. 2004.
[29] P. S. L. M. Barreto and M. Naehrig, Pairing-Friendly Elliptic Curves of Prime Order With Embedding Degree 12, IEEE Standard P1363.3, 2006.
[30] Amazon Elastic Compute Cloud (Amazon EC2). [Online]. Available: http://aws.amazon.com/ec2/, accessed Aug. 2013.
[31] Amazon Simple Storage Service (Amazon S3). [Online]. Available: http://aws.amazon.com/s3/, accessed Aug. 2013.
[32] Amazon EC2 Instance Types. [Online]. Available: http://aws.amazon.com/ec2/, accessed Aug. 2013.
[33] P. S. L. M. Barreto and M. Naehrig, "Pairing-friendly elliptic curves of prime order," in Proc. 12th Int. Workshop SAC, 2005, pp. 319–331.
[34] A. L. Ferrara, M. Green, S. Hohenberger, and M. Ø. Pedersen, "Practica l short signature batch verification," in Proc. Cryptograph. Track RSA Conf., 2009, pp. 309–324.
[35] A. F. Barsoum and M. A. Hasan. (2011). "On verifying dynamic multiple data copies over cloud servers," IACR Cryptology ePrint Archive, Tech. Rep. 2011/447. [Online]. Available: http://eprint.iacr.org/
[36] Y. Zhu, H. Wang, Z. Hu, G.-J. Ahn, H. Hu, and S. S. Yau, "Efficient provable data possession for hybrid clouds," in Proc. 17th ACM Conf. Comput. Commun. Secur. (CCS), 2010, pp. 756–758.