Enabling Data Auditing and Mutual Trust in Public Cloud

[1] P Jayarao , [2] Ch. Harika

[1]M.Tech Research Scholar, Department of CSE,
[2] Assistant Professor, Department of CSE
Priyadarshini Institute of Technology & Science, Chintalapudi, India

Abstract— *Cloud computing is a promising computing model that enables convenient and on-demand network access to a shared pool of computing resources. The local management of such huge amount of data is problematic and costly due to the requirements of high storage capacity and qualified personnel. Therefore, Storage-as-a-Service offered by cloud service providers (CSPs) emerged as a solution to mitigate the burden of large local data storage and reduce the maintenance cost by means of outsourcing data storage. We discuss the security issues of the proposed scheme. The proposed secure model provides the security of cloud services by the following ways: 1) secure cloud service 2) secure web platform 3) secure cloud infrastructure 4) Secure cloud data pool. As a result, data possession checking on cloud storage becomes one of the biggest concerns.*

Index Terms—Outsourcing data storage, dynamic environment, data possession, infrastructure

INTRODUCTION

Presently, the amount of sensitive data produced by many organizations is outpacing their storage ability. The management of such huge amount of data is quite expensive due to the requirements of high storage capacity and qualified personnel. Storage-as-a-Service (SaaS) offered by cloud service providers (CSPs) is a paid facility that enables organizations to outsource their data to be stored on remote servers. Thus, SaaS reduces the maintenance cost and mitigates the burden of large local data storage at the organization's end. A data owner pays for a desired level of security and must get some compensation in case of any misbehavior committed by the CSP. On the other hand, the CSP requires a protection from any false allegation that may be claimed by the owner to get illegal compensations.In cloud data storage, a user stores his data through a CSP into a set of cloud servers, which are running in a simultaneous, cooperated and distributed manner. For the application purposes, the user interacts with the cloud servers via CSP to access or retrieve his data. In some cases, the user may need to perform block level operations on his data. The most general forms of these operations we are considering are block update, delete, insert and append.

The data owners lose the control over their sensitive data once the latter is outsourced to a remote CSP which may not be trustworthy. This lack of control raises new formidable and challenging tasks related to data confidentiality and integrity protection in cloud computing. Customers require that their data remain secure over the CSP. Also, they need to have strong evidence that the cloud servers still possess the data and it is not being tampered with or partially deleted over time, especially because the internal operation details of the CSP may not be known to cloud customers.

Various schemes are available which supports the data owner to outsource their sensitive data to the untrusted cloud storage by giving assurance related to the confidentiality, integrity and access control. These schemes prevent and identify malicious actions from the CSP side. In this we proposed a technique which directs some important concerns associated with outsourcing sensitive data to the untrusted remote CSP, namely dynamic data, newness, mutual trust and access control.

Main contributions:

- The implementation of a cloud-based storage scheme that has the following roles: (i) allowing a data owner to outsource to a CSP, and perform full at the block-level operations more dynamically, i.e., it supports operations such as block modification, insertion, deletion, and append; (ii) ensuring the newness property, i.e., the authorized users receive the most recent version of the out-sourced data; (iii) developing *indirect* mutual trust between the data owner and the CSP since each party resides in a different trust domain; and (iv) enabling the access control for the outsourced data.

- We discuss the security features of the proposed scheme. Besides, we justify its performance through theoretical analysis and a prototype implementation on Amazon cloud platform to evaluate storage, communication, and computation overheads.

2 SYSTEM MODEL AND ASSUMPTIONS

System components and relations: Representative network architecture for cloud data storage is illustrated in Fig.1. Three different network entities can be identified as follows: 1) Client, 2) Cloud Service Provider (CSP), and 3) Third Party Auditor (TPA).

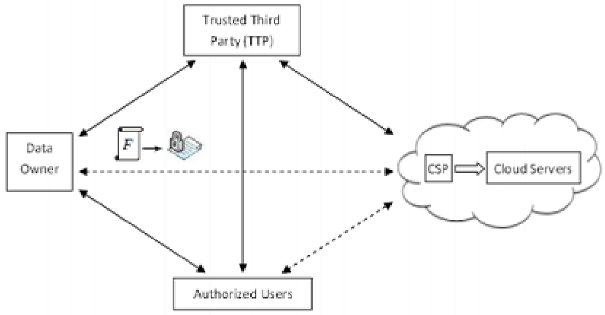

Fig. 1: Cloud computing data storage system model.

1) **Client:** Clients, who have data to be stored in the cloud and rely on the cloud for data computation, consists of both individual consumers and organizations.
2) **Cloud Service Provider (CSP):** A CSP, who has significant resources and expertise in building and managing distributed cloud storage servers, owns and operates live Cloud Computing Systems.
3) **Third Party Auditor (TPA):** An optional TPA, who has expertise and capabilities that clients may not have, is trusted to assess and

expose risk of cloud storage services on behalf of the clients upon request.

In cloud data storage, a client keeps his data through a CSP into a set of cloud servers, which are running in a simultaneous, cooperated and distributed manner. Data reputation can be employed with technique of erasure correcting code to further tolerate mistakes or server crash as clients data grows in size and importance. Thereafter, for application purposes, the client interacts with the cloud servers via CSP to access or retrieve his data. In some cases, the client may need to perform block level operations on his data. The most general forms of these operations we are considering are block update, delete, insert and append.

As clients no longer possess their data locally, it is of critical importance to assure clients that their data are being correctly stored and maintained. That is, clients should be equipped with security means so that they can make continuous correctness assurance of their stored data even without the existence of local copies. In case those clients do not necessarily have the time, feasibility or resources to monitor their data, they can delegate the tasks to an optional trusted TPA of their respective choices. In our model, we assume that the point-to-point communication channels between each cloud server and the client is authenticated and reliable, which can be achieved in practice with little overhead. Note that we don't address the issue of data privacy in this paper, as in Cloud Computing, data privacy is orthogonal to the problem we study here.

Remark: Many owners can use the same Cloud server to provide services to their set of users. In this module CSP has to get the key first. Then only he can store the file in his cloud server.TTP can only check the CSP whether the CSP is authorized or not. If it is not authorized, TTP won't allow the file to store in cloud server. Finally, TTP has monitors the data owners file by verifying the data owner's file and stored the file in a database .Also TTP checks the CSP and find out whether the CSP is authorized one or not.

Threat model: Security threats faced by cloud data storage can come from two different sources. On the one hand, a CSP can be self-interested, untrusted and possibly malicious. Not only does it desire to move data that has not been or is rarely accessed to a lower tier of storage than agreed for monetary reasons, but it may also attempt to hide a data loss incident due to management errors, complex failures and so on. On the other hand, there may also exist an economically-motivated adversary, who has the capability to compromise a number of cloud data storage servers in different time intervals and subsequently is able to modify or delete users' data while remaining undetected by CSPs for a certain period. Specifically, we consider two types of adversary with different levels of capability in this paper:

Weak Adversary: The adversary is interested in corrupting the user's data files stored on individual servers. Once a server is comprised, an adversary can pollute the original data files b y modifying or introducing its own fraudulent data to prevent the original data from being retrieved by the user.

Strong Adversary: This is the worst case scenario, in which we assume that the adversary can compromise all the storage servers so that he can intentionally modify the data files as long as they are internally consistent. In fact, this is equivalent to the case where all servers are colluding together to hide a data loss or corruption incident.

Design Goals: Our design goals can be summarized as the following: (1) Public verification for storage correctness assurance: to allow anyone, not just the clients who originally stored the file on cloud servers, to have the capability to verify the correctness of the stored data on demand; (2) Dynamic data operation support: to allow the clients to perform block-level operations on the data files while maintaining the same level of data correctness assurance. The design should be as efficient as possible so as to ensure the seamless integration of public verifiability and dynamic data operation support; (3) Blockless verification: no challenged file blocks should be retrieved by the verifier (e.g., TPA) during verification process for both efficiency and security concerns. (4) Stateless verification: to eliminate the need for state information maintenance at the verifier side between audits throughout the long term of data storage.

3 SYSTEM PRELIMINARIES

Lazy Revocation: Lazy Revocation the data owner can revoke the rights of some AU for accessing the outsourced data, the user can access the unmodified data block he cannot access the updated or new block.

Key Rotation: Key Rotation is the technique user can generate a sequence of key by using initial key and a master secret key it has to property (i) the owner of the master secret key can generate next key in sequence (ii) AU knowing the sequence of key can generate the previous keys.

Broadcast Encryption (bENC): Broadcast Encryption (bENC) is to enforce the access control over the outsourced data. This allows the broadcaster to encrypt the data for a set of arbitrary user, the set of user only can decrypt the message.

PROPOSED CLOUD-BASED STORAGE SCHEME

Our proposed work addresses some important concerns regarding outsourcing data storage to the remote untrusted storage, such as dynamic data, mutual trust, access control and newness. In our proposed work the owner is allowed to do data modifications on the outsourced data. To validate the newness property of the outsourced data, it requires some metadata which mirror the latest modifications on the outsourced data issued by the data owner. However the block indices must have the awareness that the CSP has modified the blocks at the requested position. At this end, the proposed scheme uses combined hash values and a small data structure called Block Status Table (BST). The TTPA (Trusted Third Party) establishes mutual trust between data owner, CSP and authorized users in an indirect way. To enforce access control the proposed scheme uses three cryptographic functions, namely BrdEnc (Broadcast Encryption), Key Rotation and Lazy Revocation. The BrdEnc allows the data owner to encrypt some confidential information to only authorized users allowing them to access the outsourced data. Lazy revocation enables the revoked users to access the older version of the outsourced data i.e. only the authorized users are allowed to access the most recent version of the outsourced data. Using key rotation authorized users can access both latest version of the data and older version of the data.

Block Status Table

The block status table is a small data structure used to access and restructure the received file blocks. BST will contain three columns SN, BN, and KV. SN is a serial number which indicates physical positioning of the file blocks. BN indicates the block number of the file blocks. KV indicates the Key Version under which the file block is encrypted. Table 1-3 shows the example BST structure for a file with 8 blocks.Initially the ctr is initialized to 1 as in Table I. The KV is set to ctr. Table II indicates the BST entries for the deletion of block at position = 5 while there is no revocation of users. Hence, the ctr remains unchanged. But in Table III the ctr is incremented by 1 i.e., ctr=2 since, there is an revocation. Hence, insertion of new block following revocation is encrypted under KV =2.

A. Our System Model

Cloud storage model considered in our proposed work has four main components as depicted in Fig.1

i. A data owner can be an organization, which generates sensitive data that is to be outsourced to the cloud and made available for only authorized users.

ii. A Trusted Third Party Auditor (TTPA) [17] who is trusted by all other components and has the capability to detect the dishonest party.

iii. A CSP who manages cloud services and provides paid storage service on its infrastructure to the data owner, where he outsources the file and makes them available for authorized users.

iv. Authorized users – a set of owner's clients who have the right to access the outsourced file.

Our cloud storage system model can be adopted by many practical applications. For example, Educational applications can be visualized by our model as in ig.1, where the student's database that contains large and sensitive information can be stored on cloud servers. In this type of application, an institution can be considered as a data owner, the teaching staffs can be considered as the authorized users, who has given the access rights over the outsourced student's information, and an independent organization can be considered as the TTPA. Likewise more practical applications can be envisioned in similar settings. The auditing process of the data received from the CSP is done by authorized users. We used TTPA only to solve disputes that may arise due to data integrity and newness verification.

B. Outsourcing, updating and accessing

The data owner has a file F, which is divided into m blocks and is to be outsourced to CSP, who will provide paid storage space to the data owner. Before outsourcing the file to the cloud server, to achieve confidentiality the owner encrypts the file blocks. After doing so, the owner can interact with the CSP to do full block-level dynamic operations on the file. These block-level operations include insert, delete, append, and modify certain blocks of the outsourced file. For time being, we have considered only insert and delete operations in our work. An authorized user receives the encrypted file, by sending the data access request to the CSP. The encrypted file can be decrypted using a secret key which can be generated by the authorized user.

We imagine that, the verification of the authorized users' identity has already been done with the data owner; hence we haven't considered this in our work. And also all authorized users have the same access privilege over the outsourced data.The TTPA is an autonomous entity, and thus has no motivation to collude with any party in the system. The TTPA and the CSP are always online, while the data owner can be online or offline. Even though the owner is in offline, the authorized users can access the outsourced data from the CSP.

C. Access control mechanism

The three cryptographic techniques Lazy Revocation, Key Rotation and Broadcast Encryption which are discussed below are combined to enforce access control over outsourced data.

i. Lazy Revocation

The data owner in our proposed work is allowed to revoke access right of some users from accessing the outsourced data at any time. The revoked users are allowed to access unmodified blocks in Lazy Revocation. However, modified or new blocks must not be accessed by such revoked users. This is equivalent to accessing the file blocks from caches. The idea behind this scheme is, modified or new blocks following revocation are encrypted under new key. Thus each data block may have more than one key. Lazy Revocation trades re-encryption cost. Lazy Revocation has been used in many cryptographic schemes.

ii. Key Rotation

In this technique [7], a sequence of keys can begenerated from an initial key and a master secret key. The sequence of keys has two main characteristics

i. The next key in the sequence can only be generated by the owner of the master secret key.

ii. The authorized users knowing the key in the sequence can able to generate previous keys in the sequence. i.e. given the ith key key_i in the sequence, the authorized users can compute the previous keys in the sequence { Keyj } where j < i, but it is infeasible to compute { Keyj }, where j > i without having the master secret key.

Property i. allows the data owner to revoke the access right over outsourced data Property ii. Allows the authorized users to maintain access to the file blocks Let N = pq denote a RSA modulus (p & q are prime numbers), a public key = (N, e)and a master secret key d. The key d is known only to the data owner, and ed ≡ 1 mod(p-1)(q-1).Whenever a user's access is revoked the key is rotated forward to generate new key in the sequence as

$$Key_{ctr+1} = key_{ctr}^{d} \bmod N$$

The authorized users can recursively generate older versions of the key (backward rotation) as

iii. Broadcast Encryption

Broadcast Encryption (BrdEnc) scheme allows a broadcaster to encrypt a message for a group of users. The users in the group can only able to decrypt the message. However, the users outside the group collude they could not decrypt the message. In our work, we use BrdEnc to enforce access control over outsourced data. This scheme is a combination of three algorithms

a) Setup b) Encrypt c) Decrypt

5. EXPERIMENTAL EVALUATION

In this section we experimentally evaluate the computation overhead the proposed scheme brings to a cloud storage system that has been dealing with static data with only confidentiality requirement. The experiments are conducted using NETBEANS on a system with an Intel(R) 2-GHz processor and 3GB RAM running Windows XP. Algorithms (hashing, broadcast encryption, digital signatures, etc.) are implemented using MIRACL library version 5.5.4. For a 128-bit security level, bENC uses an elliptic curve with a 256-bit group order. In the experiments, we utilize SHA-256, 256-bit BLS signature, and Bar-reto-Naehrig (BN) [50] curve defined over prime field GF(p) with p = 256 bits and embedding degree = 12 (the BN curve with these parameters is provided by the MIRACL library). To evaluate the computation overhead on the owner side due to dynamic operations, we perform 100 different block operations from which 50% are executed following revocations (this percent is higher than an average value in practical applications).Scalability (i.e., how the system performs when more users are added) is an important feature of cloud storage systems. The access control of the proposed scheme depends on the square root of the total number of system users. To identify the dishonest party in the system in case of disputes, the TTP verifies two signatures (F and T), computes combined hashes for the data (file and table), and compare the computes hashes with the authentic values (THTTP and FHTTP). Thus, the computation overhead on the TTP side is about 10.77 seconds. Through our experiments, we use only one desktop computer to simulate the TTP and accomplish its work. In practice, the TTP may choose to split the work among a few devices or use a single device with a multi-core processor which is becoming prevalent these days, and thus the computation.

In the worst case, the TTP executes only 4 hashes per dynamic request to reflect the change on the outsourced data. Thus, the maximum computation overhead on the TTP side is about 0.08 milliseconds, i.e., the proposed scheme brings light overhead on the TTP during the normal system operations. The computation overhead on the user side due to data access comes from five aspects divided into two groups. The first group involves signatures verification and hash operations to verify the received data (file and table). The second group involves broadcast decryption, backward key rotations, and hash operations to compute the DEK.

6 CONCLUSIONS

In this paper, we have proposed a cloud-based storage scheme which supports outsourcing of dynamic data, where the owner is capable of not only archiving and accessing the data stored by the CSP, but also updating and scaling this data on the remote servers. The proposed scheme enables the authorized users to ensure that they are receiving the most recent version of the outsourced data. Moreover, in case of dispute regarding data integrity/newness, a TTP is able to determine the dishonest party. The data owner enforces access control for the outsourced data by combining three cryptographic techniques: broadcast encryption, lazy revocation, and key rotation. We have studied the security features of the proposed scheme.

We have investigated the overheads added by our scheme when incorporated into a cloud storage model for *static* data with only *confidentiality* requirement. The storage overhead is ≈ 0.4% of the outsourced data size, the communication overhead due to block-level dynamic changes on the data is ≈ 1% of the block size, and the communication overhead due to retrieving the data is ≈ 0.2% of the outsourced data size. For a large organization with 10^5 users, performing dynamic operations and enforcing access control add about 63 milliseconds of overhead. Therefore, important features of outsourcing data storage can be supported without excessive over- heads in storage, communication, and computation.

REFERENCES

[1] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Pe-terson, and D. Song, “Provable data possession at untrusted stores,” in *Proceedings of the 14th ACM Conference on Computer and Communications Security*, ser. CCS ’07, 2007, pp. 598–609.

[2] F. Sebe,´ J. Domingo-Ferrer, A. Martinez-Balleste, Y. Deswarte, and J.-J. Quisquater, “Efficient remote data possession checking in critical information infrastructures,” *IEEE Trans. on Knowl. and Data Eng.*, vol. 20, no. 8, 2008.

[3] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, “Scalable and efficient provable data possession,” in *Proceedings of the 4th International Conference on Security and Privacy in Communication Netowrks*, 2008, pp. 1–10.

[4] C. Erway, A. Kupc¸¨u,¨ C. Papamanthou, and R. Tamassia, “Dynamic provable data possession,” in *Proceedings of the 16th ACM Confer-ence on Computer and Communications Security*, 2009, pp. 213–222.

[5] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, “Enabling public verifiability and data dynamics for storage security in cloud computing,” in *Proceedings of the 14th European Conference on Research in Computer Security*, 2009, pp. 355–370.

[6] A. F. Barsoum and M. A. Hasan, “Provable possession and replication of data over cloud servers,” Centre For Applied Cryp-tographic Research, Report 2010/32, 2010, http://www.cacr.math. uwaterloo.ca/techreports/2010/cacr2010-32.pdf.

[7] R. Curtmola, O. Khan, R. Burns, and G. Ateniese, “MR-PDP: multiple-replica provable data possession,” in *28th IEEE ICDCS*, 2008, pp. 411–420.

[8] A. F. Barsoum and M. A. Hasan, “On verifying dynamic multi-ple data copies over cloud servers,” Cryptology ePrint Archive, Report 2011/447, 2011, 2011, http://eprint.iacr.org/.

[9] K. D. Bowers, A. Juels, and A. Oprea, “HAIL: a high-availability and integrity layer for cloud storage,” in *CCS ’09: Proceedings of the 16th ACM conference on Computer and communications security*. New York, NY, USA: ACM, 2009, pp. 187–198.

[10] Y. Dodis, S. Vadhan, and D. Wichs, “Proofs of retrievability via hardness amplification,” in *Proceedings of the 6th Theory of Cryptography Conference on Theory of Cryptography*, 2009.

[11] A. Juels and B. S. Kaliski, “PORs: Proofs of Retrievability for large files,” in *CCS’07: Proceedings of the 14th ACM conference on Computer and communications security*. ACM, 2007, pp. 584–597.

[12] H. Shacham and B. Waters, “Compact proofs of retrievability,” in *ASIACRYPT ’08*, 2008, pp. 90–107.

[13] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K. Fu, “Plutus: Scalable secure file sharing on untrusted storage,” in *Proceedings of the FAST 03: File and Storage*

Technologies, 2003.

[14] E.-J. Goh, H. Shacham, N. Modadugu, and D. Boneh, “Sirius: Securing remote untrusted storage,” in *Proceedings of the Network and Distributed System Security Symposium, NDSS*, 2003.

[15] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, “Improved proxy re-encryption schemes with applications to secure distributed storage,” in *NDSS*, 2005.

[16] S. D. C. di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati, “Over-encryption: Management of access control evo-lution on outsourced data,” in *Proceedings of the 33rd International Conference on Very Large Data Bases*. ACM, 2007, pp. 123–134.

[17] V. Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute-based encryption for fine-grained access control of encrypted data,” in *CCS ’06*, 2006, pp. 89–98.

[18] S. Yu, C. Wang, K. Ren, and W. Lou, “Achieving secure, scalable, and fine-grained data access control in cloud computing,” in *INFOCOM’10*, 2010, pp. 534–542.