ISSN: 2321-7782 (Online)
Impact Factor: 6.047

Volume 4, Issue 6, June 2016

International Journal of Advance Research in Computer Science and Management Studies

Research Article / Survey Paper / Case Study

Available online at: www.ijarcsms.com

# *Unique Identity Based Provable Data Possession at Unstructured Stores in Multi-Cloud Storage Using IBE*

**E. Kiruthika**[1]
Research Scholar
H.H.The Rajah's College (Autonomous),
Pudukkottai – India

**Dr. S. Ravichandran**[2]
Research Guide, Head of The Department of CA
H.H.The Rajah's College (Autonomous),
Pudukkottai – India

***Abstract:*** ***Cloud computing has become an important thing in computer field. Cloud computing takes information processing as a service, such as storage and computing. Data integrity is important thing in cloud storage. In certain situations, clients should store their data such as image or text in multi cloud. When the client stores his/her data on multi cloud servers, the distributed storage and integrity checking is very important. Checking remote data possession is of crucial importance in public cloud storage. It enables the users to check that their outsourced data have been kept intact without downloading the original data. The existing remote data possession checking (RDPC) protocols have been designed in the PKI (public key infrastructure) setting. The cloud server has to validate the users' certificates before storing the data uploaded by the users in order to prevent spam. This incurs considerable costs since numerous users may frequently upload data to the cloud server. This research work proposes 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.***

***Keywords: Cloud Storage, IBE, Multi-cloud servers, Provable Data Possession, Storage Security***

## I. INTRODUCTION

In this research work, propose an effective and flexible distributed scheme with explicit dynamic data support to ensure the correctness of users' data in the cloud. This paper proposed work relies on erasure correcting code in the file distribution preparation to provide redundancies and guarantee the data dependability [1]. This construction drastically reduces the communication and storage overhead as compared to the traditional replication-based file distribution techniques. By utilizing the homomorphic token with distributed verification of erasure-coded data, our scheme achieves the storage correctness insurance as Well as data error localization: whenever data corruption has been detected during the storage correctness verification, our scheme can almost guarantee the simultaneous localization of data errors, i.e., the identification of the misbehaving server(s).

This work is among the first few ones in this field to consider distributed data storage in Cloud Computing. Our contribution can be summarized as the following three aspects:

1) Compared to many of its predecessors, which only provide binary results about the storage state across the distributed servers, the challenge-response protocol in our work further provides the localization of data error.

 

2) Unlike most prior works for ensuring remote data integrity, the new scheme supports secure and efficient dynamic operations on data blocks, including: update, delete and append.

3) Extensive security and performance analysis shows that the proposed scheme is highly efficient and resilient against Byzantine failure, malicious data modification attack, and even server colluding attacks.

In order to solve the problem of data integrity checking, many schemes are proposed under different systems and security models. In all these works, great efforts are made to design solutions that meet various requirements: high scheme efficiency, stateless verification, unbounded use of queries and retrievability of data, etc. Considering the role of the verifier in the model, all the schemes presented before fall into two categories: private auditability and public auditability. Although schemes with private auditability can achieve higher scheme efficiency, public auditability allows anyone, not just the client (data owner), to challenge the cloud server for correctness of data storage while keeping no private information. Then, clients are able to delegate the evaluation of the service performance to an independent third party auditor (TPA), without devotion of their computation resources. In the cloud, the clients themselves are unreliable or may not be able to afford the overhead of performing frequent integrity checks. Thus, for practical use, it seems more rational to equip the verification protocol with public auditability, which is expected to play a more important role in achieving economies of scale for Cloud Computing. Moreover, for efficiency consideration, the outsourced data themselves should not be required by the verifier for the verification purpose.

Another major concern among previous designs is that of supporting dynamic data operation for cloud data storage applications. In Cloud Computing, the remotely stored electronic data might not only be accessed but also updated by the clients, e.g., through block modification, deletion, insertion, etc. Unfortunately, the state of the art in the context of remote data storage mainly focus on static data files and the importance of this dynamic data updates has received limited attention so far . Moreover, as will be shown later, the direct extension of the current provable data possession (PDP) [2] or proof of retrievability (PoR) [3], [4] schemes to support data dynamics may lead to security loopholes. Although there are many difficulties faced by researchers, it is Well believed that supporting dynamic data operation can be of vital importance to the practical application of storage outsourcing services. In view of the key role of public auditability and data dynamics for cloud data storage, this system proposes an efficient construction for the seamless integration of these two components in the protocol design.

## II. LITERATURE SURVEY

**Ateniese et al.** [5] are the first to consider public auditability in their defined “provable data possession” model for ensuring possession of files on untrusted storages. In their scheme, they utilize RSA-based homomorphic tags for auditing outsourced data,thus public auditability is achieved.

**B. Krebs,** [7] describe a “proof of retrievability” model, where spot-checking and error-correcting codes are used to ensure both “possession” and “retrievability” of data files on archive service systems.

**Erway et al.** [8] were the first to explore constructions for dynamic provable data possession. They extend the PDP model to support provable updates to stored data files using rank-based authenticated skip lists.

**M. A. Shah, R. Swaminathan, and M. Baker,** [9] The propose allowing a TPA to keep online storage honest by first encrypting the data then sending a number of pre-computed symmetric-keyed hashes over the encrypted data to the auditor.

**T.Dillon, C.Wa, and E.Chang,** [10] propose to combine BLS-based HLA with MHT to support both public auditability and full data dynamics. Almost simultaneously developed a skip lists based scheme to enable provable data possession with full dynamics support. However, the verification in these two protocols requires the linear combination of sampled blocks just as and thus does not support privacy preserving auditing.

 

**H.-C. Hsiao et al,** [11] the traditional cryptographic technologies for data integrity and availability, based on Hash functions and signature schemes, cannot work on the out sourced data without a local copy of data. In addition, it is not a practical solution for data validation by downloading them due to the expensive communications, especially for large size files.

## III. METHODOLOGY

### *A. Cloud Computing*

Cloud computing is a term used to refer to a model of network computing where a program or application runs on a connected server or servers rather than on a local computing device such as a PC, tablet or Smartphone. Like the traditional client-server model or older mainframe computing, a user connects with a server to perform a task.

The computing resources have become "granular", which provides end user and operator benefits including on-demand self-service, broad access across multiple devices, resource pooling, rapid elasticity and service metering capability. In more detail, cloud computing refers to a computing hardware machine or group of computing hardware machines commonly referred as a server or servers connected through a communication network et, an intranet, a local area network (LAN) or wide area network (WAN).

### *B. Enhancing The Accountability*

Cloud computing may be a massive infrastructure which give several services to user while not installation of resources on their own machine. This is often the pay as you utilize model. Samples of the cloud services are Yahoo email, Google, Gmail and Hotmail. There are several users, businesses, government uses cloud, thus knowledge usage in cloud is massive. Thus knowledge maintenance in cloud is advanced. Several Artists desires to try to business of their art victimization cloud.

### *C. Cloud Ingredients*

There is need to be compelled to offer technique that is ready to audit information in cloud. On the idea of accountability, they have an inclination to projected one mechanism that keeps use information clear suggests that data owner got to get information regarding use of his information. This process support accountability in distributed area, data owner should not problem regarding his information, he may acknowledge his information is handled per service level agreement and his information is riskless on cloud. Data owner will determine the authorization principles and policies and user will handle information victimization this rule and logs of each information access are created. Throughout this mechanism there are unit two main parts i.e. logger and log harmonizer. The feller is with the data owner's information, it provides work access to information and encrypts log record by pattern public key that's given by data owner and send it to log harmonizer. The log harmonizer is taking part in the observance and rectifying, it generates the key it holds cryptography key decrypting the logs, and at the consumer side cryptography it sends key to shopper. Throughout this mechanism data owner will creates personal key and public key, pattern generated key owner will produce feller that will be a JAR file, it encloses his authorization principles and work policies with information send to cloud service provider.

### *D. Flow Of Data*

The overall CIA framework, combining information, users, logger and harmonizer is sketched. At the start, every user creates a combine of public and personal keys supported Identity-Based encoding. This IBE scheme could be a This paper-pairing-based IBE scheme that protects us against one among the most current attacks to exploitation the generated key, the user can produce a logger part that may be a JAR file, to store its data items.

 

Depending on the configuration settings outlined at the time of creation, the JAR can give usage management related to logging, or can give solely work practicality. As for the work, when there's associate access to the information, the JAR can mechanically generate a log record, encipher it victimization the general public key distributed by the data owner. The encoding of the log file prevents unauthorized changes to the file by attackers. The data owner could opt to reuse the same key pair for all JARs or create different key pairs for different JARs. Using separate keys are able to improve the authorization without introducing any overhead except in the starting phase. In inclusion, some error correction data will be sent to the log harmonizer to handle possible log file corruption. To ensure reliability of the logs, each record is signed by the entity accessing the content. In earlier, own records are hashed together to create a chain formation, can easily identify possible errors or lost files.

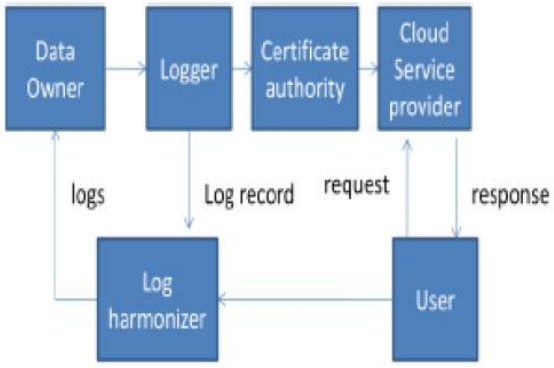


Fig.1. IBE Process over Cloud

*E. State Transition Diagram*

Where, 0: Unsuccessful 1:

Successful Transitions are:

S0: Data Owner will send data to logger.

S1: Data Owner will create logger which is a jar file to store data and principles. .

S2: Authentication of CSP to JAR file.

S3: Authentication of user.

S4: owner can see merge log

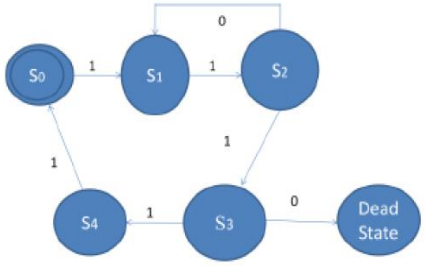


Fig. 2. State Diagram for Data Owner Logger Process

*F. FOG Computing Methodology*

In this paper which proposes a different approach for securing data in the cloud using offensive decoy method. To supervise information access in the cloud and detect abnormal data access patterns. When unsecured access is identified and after checked by raising queries, utilize a mislead attack by forwarding large amounts of decoy information to the attacker. This prevents the

 

utilization of the user's own information. Hypothesis supervises in a local file context provides evidence that this approach may provide unprecedented levels of user data security in a Cloud environment.

## IV. PROBLEM STATEMENT

### A. *FOG Computing Methodology*

Representative network architecture for cloud data storage is illustrated in Fig. 1. Three different network entities can be identified as follows:

- **Client:** an entity, which has large data files to be stored in the cloud and relies on the cloud for data maintenance and computation, can be either individual consumers or organizations;
- **Cloud Storage Server (CSS):** an entity, which is managed by Cloud Service Provider (CSP), has significant storage space and computation resource to maintain the clients' data;
- **Third Party Auditor:** an entity, which has expertise and capabilities that clients do not have, is trusted to assess and expose risk of cloud storage services on behalf of the clients upon request.

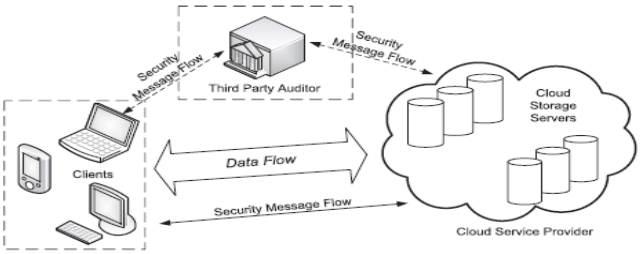


Fig 3.1. Cloud Storage Architecture

In the cloud paradigm, by putting the large data files on the remote servers, the clients can be relieved of the burden of storage and computation. As clients no longer possess their data locally, it is of critical importance for the clients to ensure that their data are being correctly stored and maintained.

### B. *Security Model*

Following the security model defined that the checking scheme is secure if,

1) There exists no polynomial time algorithm that can cheat the verifier with non negligible probability; and

2) There exists a polynomial time extractor that can recover the original data files by carrying out multiple challenges-responses.

The client or TPA can periodically challenge the storage server to ensure the correctness of the cloud data, and the original files can be recovered by interacting with the server. The proposed system can also define the correctness and soundness of their scheme: the scheme is correct if the verification algorithm accepts when interacting with the valid prover (e.g., the server returns a valid response) and it is sound if any cheating server that convinces the client it is storing the data file is actually storing that file. Note that in the "game" between the adversary and the client, the adversary has full access to the information stored in the server, i.e., the adversary can play the part of the prover (server). The goal of the adversary is to cheat the verifier successfully, i.e., trying to generate valid responses and pass the data verification without being detected.

### C. *Design Goals*

Our design goals can be summarized as the following:

1. Public auditability 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 auditability 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 efficiency concern

*D. Problem Definition*

The cloud has the following requirements: The photographs are downloaded only by users who have paid for the services. Potential buyers are allowed to view the pictures first before they make the payment to obtain the download right. Due to the nature of some of the works, only users from certain countries can view or download some sets of photographs. For some of the works, users are allowed to only view them for a limited time, so that the users cannot reproduce the work easily. In case any dispute arises with a client, they want to have all the access information of that client. The user want to ensure that the cloud service providers of Sky High do not share the data with other service providers, so that the accountability provided for individual users can also be expected from the cloud service providers. With the above scenario in mind, to identify the common requirements and develop several guidelines to achieve data accountability in the cloud.

## V. IMPLEMENTATION

*A. User Behavior Profiling*

It is expected that access to a user's information in the Cloud will exhibit a normal access. User profiling is a popular method that can be applied here to design how, when, and how much a client utilizes their data in the Cloud. Such 'normal user' behavior can be continuously checked to determine whether abnormal access to a user's information is happening or not.

*B. Decoys*

Decoy information, such as decoy documents, honey files, honey pots, and various other bogus information can be generated on demand and serve as a means of detecting unauthorized access to information and to 'poison' the thief's ex-filtrated information. Serving decoys will confound and confuse an attacker into believing they have ex-filtrated useful information or not. This method may be integrated with user behavior profiling technology to secure a user's information in the Cloud. Whenever exceptional access to a cloud events are observed, decoy data can be get back by the Cloud and delivered in such a way as to appear completely lawful.

In Push Pull key matching algorithm the key is generated by UID, Access type, Time and Address Key is generated by UID, Access type, Time and address if Tserver-Tclient is minimum than expected time the no network issues So the key is encrypted after that Registration Manager Broadcast Same encrypted key to the Log Manager and application User. If key of Log manager and user key matched then decrypt the key. Another case is that if Tserver-Tclient is larger than expected time then permission for that log is denied.

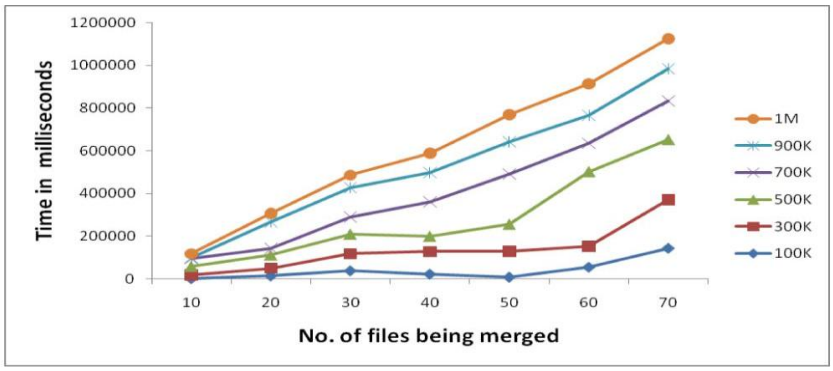


Fig. 3. Performaence Evaluation of Push and Pull Algorithm

Finally, to investigate whether a single logger, used to handle more than one file, results in storage overhead. Measure the size of the loggers (JARs) by varying the number and size of data items held by them. Tested the increase in size of the logger containing 10 content files (i.e., images) of the same size as the file size increases. Intuitively, in case of larger size of data items held by a logger, the overall logger also increases in size. The size of logger grows from 3,500 to 4,035 KB when the size of Content items changes from 200 KB to 1 MB.

In this algorithm when any other request meet to the cloud server and that will be already register by the log manager then Cloud Subscriber only performing pull operation performed. If CSP load is minimum than that of moderated load then server is light Weighted so whatever the request available in request queue that will be executed. If in Case load on the server increases i.e CSP load is more than that of expected load then reduced server load if due to higher load the CSP is timed out then Switch Push mode to pull mode.

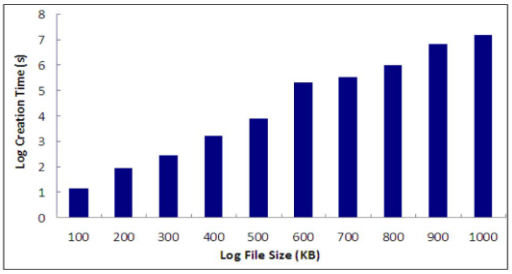


Fig Analysis the Process of Log Creation

*Temporal Identity Based Encryption*

This Policy generates temporary policy based on the Identity of user. This requires the following parameter such as UserId, Access type as Timing information about both the User and Registration Server and Location determine by the IP and MAC address.

Step 1: Start

Step2: GenKey (MSK,uk,A,T,L):Registration server Takes the user's ID, Access Type A, Time information T and Location of User U.

Step 3: Encrypt (EK): Registration Manager Delegate Key to the user and also to the Log manager.

Step 4: Decrypt (LSK,EK): Log Manager Issues the Decryption key from Registration manager LSK over Encrypted key EK and both the keys in Log manager and User are decrypted.

Step 5: Stop

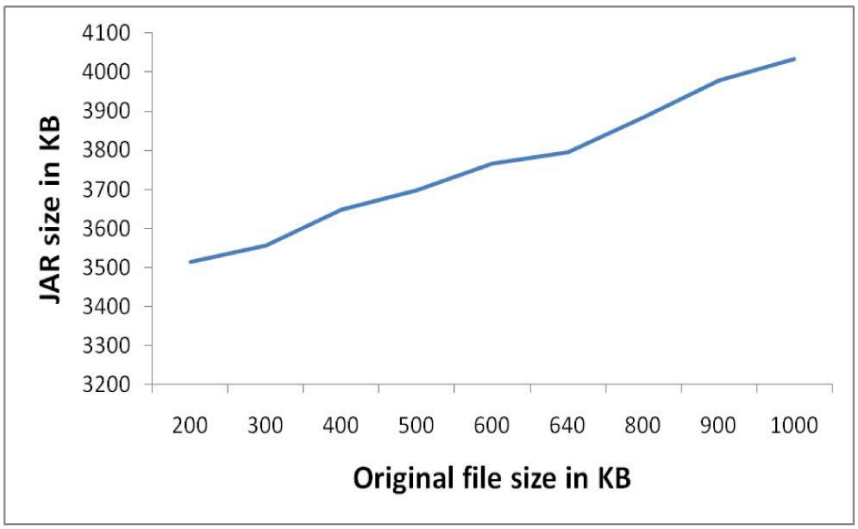


Fig 6.3 Encrypted File Size measure of JAR in Cloud

 

## VI. Conclusion and Further Work

Cloud computing security is still considered the major issue in the cloud computing environment. Customers do not want to lose their private information as a result of malicious insiders in the cloud. In addition, the loss of service availability has caused many problems for a large number of customers recently. Furthermore, data intrusion leads to many problems for the users of cloud computing. Cloud computing is currently the latest trend when it comes to online computing, it may help the enterprise and the end user by providing their needs, but the provider has to make sure that they are valuable and customer data is safe. The purpose of this work is to provide a simple yet effective architecture that will give end to end solution for cloud security as Wells it will maintain transparency among owner, CSP and End user.

In future would like to develop a cloud, on which will install JRE and JVM, to do the authentication of JAR. Try to improve security of store data and to reduce log record generation time. It is more and more important to defend and preserve people privacy on the Internet, against unwanted and unauthorized disclosure of their confidential data.

## References

1. Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing," Proc. 14th European Symp. Research in Computer Security (ESORICS '09), pp. 355-370, 2009.
2. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable Data Possession at Untrusted Stores," Proc. 14th ACM Conf. Computer and Comm. Security (CCS'07), pp. 598-609, 2007.
3. Juels and B.S. Kaliski Jr., "Pors: Proofs of Retrievability for Large Files," Proc. 14th ACM Conf. Computer and Comm. Security (CCS '07), pp. 584-597, 2007.
4. Shacham and B. Waters, "Compact Proofs of Retrievability," Proc. 14th Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT '08), pp. 90-107, 2008.
5. G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable Data Possession at Untrusted Stores," Proc. 14th ACM Conf. Computer and Comm. Security (CCS '07), pp. 598-609, 2007.
6. G. Ateniese, R.D. Pietro, L.V. Mancini, and G. Tsudik, "Scalable and Efficient Provable Data Possession," Proc. Fourth Int'l Conf. Security and Privacy in Comm. Networks (SecureComm '08), pp. 1-10, 2008.
7. B. Krebs, "Payment Processor Breach May Be Largest Ever," Online at http://voices.washingtonpost.com/securityfix/2009/01/payment processor breach may b.html, Jan. 2009.
8. C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia, "Dynamic Provable Data Possession," Proc. 16th ACM Conf. Computer and Comm. Security (CCS '09), 2009.
9. M. A. Shah, R. Swaminathan, and M. Baker, "Privacypreserving audit and extraction of digital contents," Cryptology ePrint Archive, Report 2008/186, 2008.
10. T.Dillon, C.Wa, and E.Chang, "Cloud Computing" IEEE Int' I. Conf. Advanced Info. Networking and Apps. 2010, pp.
11. H.-C. Hsiao, Y.-H. Lin, A. Studer, C. Studer, K.-H. Wang, H. Kikuchi, A. Perrig, H.-M. Sun, and B.-Y. Yang. A study of user-friendly hash comparison schemes. In ACSAC, pages 105–114, 2009.
12. Y. Zhu, H. Wang, Z. Hu, G.-J. Ahn, H. Hu, and S. S. Yau. Efficient provable data possession for hybrid clouds. In Proceedings of the 17th ACM conference on Computer and communications security, pages 756–758, 2010.
13. M. Xie, H. Wang, J. Yin, and X. Meng. Integrity auditing of outsourced data. In C. Koch, J. Gehrke, M. N. Garofalakis, D. Srivastava, K. Aberer, A. Deshpande, D. Florescu, C. Y. Chan, V. Ganti, C.-C. Kanne, W. Klas, and E. J. Neuhold, editors, VLDB, pages 782–793. ACM, 2007.
14. Peter Mell, and Tim Grance, "The NIST Definition of Cloud Computing," Version 15, 10-7-09, http://www.wheresmyserver.co.nz/ storage/media/faq-files/cloud-def-v15.pdf.
15. DoD, "National Industrial Security Program Operating Manual", 5220.22-M, February 28, 2006.

 