Volume 6, Issue 1, January 2019 ISSN: 2348-8565 (Online)

International Journal of Modern Engineering and Research Technology

Website: http://www.ijmert.org Email: editor.ijmert@gmail.com

Integrity Verification in Multi-Cloud Storage by Efficient Cooperative Provable Data Possession

Madhulata Silawat
M.Tech. Research Scholar
Takshshila Institute of Engineering & Technology
Jabalpur (M.P.), [INDIA]
Email: madhusilawat026@gmail.com

Abhishek Pandey
Assistant Professor
Department of Computer Science & Engineering
Takshshila Institute of Engineering & Technology
Jabalpur (M.P.), [INDIA]
Email: abhishekpandey@takshshila.org

ABSTRACT

The word 'Cloud' is a network system of remote servers hosted on the internet and used to supervise, Store, and Process data in put of local server or personal computer. Multi cloud storage is an great technique using a collection of cloud to supply data storage and data sharing for patrons cooperatively. Construction of an efficient PDP scheme for distributed cloud storage to support the scalability of service and data migration. Provable data possession (PDP) is a technique for ensuring the integrity of data in storage outsourcing. We present an efficient method for selecting optimal parameter standards to minimize the computation costs of clients and storage service providers. We use Cooperative provable data possession system for based on interactive proof system (IPS). This Cooperative PDP scheme adopting Layered Index Hash Hierarchy (IHH).

Keywords:— *Cloud Storage, RSA, Data Integrity, CPDP, HMAC.*

I. INTRODUCTION

The term *Cloud* is used as a metaphor for the Internet, based on the cloud diagram used to depict the Internet in computer network diagrams as an concept of underlying transportation it represents. Typical cloud computing providers deliver common business application online which are accessed from web browser, while the software and data are stored on server.

II. CLOUD COMPUTING

Cloud computing has become one of the most discussed IT paradigms of recent years. With the rapid progress of dispensation and storage technologies and the achievement of the Internet, computing resources have become cheaper, more powerful and more ubiquitously accessible than ever before. This technological trend has enabled the understanding of a new computing model called cloud computing, in which resources (e.g. CPU and storage) are provided as general utilities that can be leased and released by users through the Internet in an on-demand style.

Cloud computing gives virtually limitless pay-per-use computing resources to users, while leaving the burden of administration the underlying communications to the cloud providers [1]. Most cloud service providers use machine virtualization to supply flexible and cost efficient resource distribution. With the usage of cloud computing services, organizations can considerably lower the need for upfront savings in information technology and ongoing support or preservation by simply

paying only for the services, capability or programs as they are used.

Cloud computing delivers communications, stage, and software as services, which are made accessible as subscription-based services to customers.

Cloud computing entrusts remote services with a user's data, software and calculation. The cloud providers manage the communications and platforms on which the applications run. End users access cloud based application through a web browser or a insubstantial desktop or mobile application while the business software and user's data are stored on servers at a remote position. Cloud computing allows enterprises to get their applications up and running earlier, with better manageability and less protection, and enables IT to more quickly adjust resources to meet variable and changeable business demand.

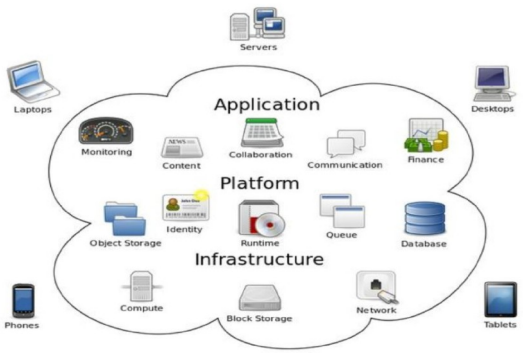

Figure 1 : Cloud Computing Infrastructure

III. Related Work

A. Collaborative Integrity Verification In Hybrid Clouds

A hybrid cloud is a cloud computing environment in which an organization provides and manages some internal resources and the others provided externally. However, this new environment could bring irretrievable losses to the clients due to lack of integrity verification mechanism for distributed data outsourcing. In this paper address the construction of a collaborative integrity verification mechanism in hybrid clouds to support the scalable service and data migration, in which consider the existence of multiple cloud service providers to collaboratively store and maintain the clients' data. A hybrid cloud is a cloud computing environment in which an organization provides and manages some internal resources as well as external resources.

Architecture considers a data storage service involving three different entities: Granted clients, a large amount of data to be stored in hybrid clouds and have the permissions to access and manipulate the stored data;

Cloud Service Providers (CSPs), work together to provide data storage services and have enough storage spaces and calculation resources; and Trusted Third Parties (TTPs), Trusted to store the confirmation parameters and offer the query services for these parameters.

B. Scalable and Efficient Provable Data Possession

Storage outsourcing is a rising trend which prompts a number of attractive security issues, many of which have been extensively investigated in the past. However, Provable Data Possession (PDP) is a topic that has only recently appeared in the research literature.

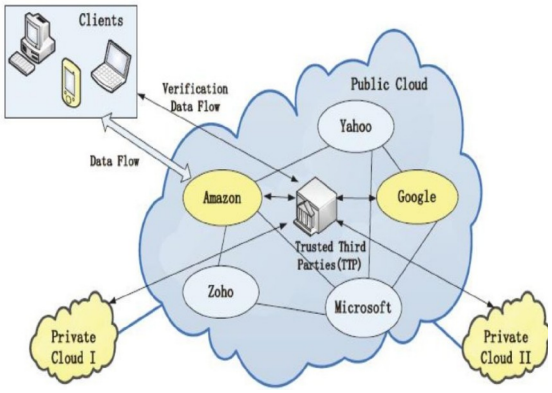

Figure 2: Hybrid Cloud

The main issue is how to regularly, competently and securely verify that a storage server is faithfully storing its client's (potentially very large) outsourced data. The storage server is assumed to be untrusted in terms of both security and dependability. (In other words, it might maliciously or unintentionally erase hosted data; it might also downgrade it to slow or off-line storage.) The problem is exacerbated by the client being a small computing device with limited resources. Prior work has addressed this difficulty using either public key cryptography or requiring the client to outsource its data in encrypted form. Construct a highly efficient and provably secure PDP technique based completely on symmetric key cryptography, while not requiring any bulk encryption. Also, in contrast with its predecessors, our PDP practice allows outsourcing of dynamic data, i. e, it professionally supports operations, such as block alteration, deletion and append.

C. Ensuring Data Storage Security In Cloud Computing

Cloud Computing has been envisioned as the next generation structural design of IT Enterprise. In contrast to traditional solutions, where the IT services are under proper physical, logical and personnel controls, Cloud Computing moves the application software and databases to the large data centers, where the management of the data and services may not be fully trustworthy. This unique attribute, however, poses many new security challenges which have not been well understood. In this article, focus on cloud data storage security, which has always been an important aspect of quality of service. To ensure the correctness of users' data in the cloud, propose an efficient and flexible dispersed scheme with two salient features, opposing to its predecessors. By utilizing the homomorphic token with distributed verification of erasure-coded data, our scheme achieves the integration of storage correctness insurance and data error localization, i.e., the identification of misbehaving server(s). Unlike most prior works, the new scheme further supports secure and efficient dynamic operations on data blocks, including: data update, delete and append. 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. Propose an effective and flexible distributed scheme with explicit dynamic data support to ensure the correctness of users' data in the cloud. We rely on erasure correcting code in the file distribution preparation to provide redundancies and guarantee the data dependability. 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).

D. Privacy-Preserving Audit and Extraction of Digital Contents

A growing number of online services, such as Google, Yahoo!, and Amazon, are starting to charge users for their storage. Customers often use these services to store valuable data such as email, family photos and videos, and disk backups. A customer must entirely trust such external services to maintain the integrity of hosted data and return it intact. Unfortunately, no service is infallible. To make storage services

accountable for data loss, we present protocols that allow a third party auditor to periodically verify the data stored by a service and assist in returning the data intact to the customer. Most importantly, our protocols are privacy-preserving, in that they never reveal the data contents to the auditor. Our solution removes the burden of verification from the customer, alleviates both the customer's and storage service's fear of data leakage, and provides a method for independent arbitration of data retention contracts.

E. Dynamic Provable Data Possession

As storage-outsourcing services and resource-sharing networks have become popular, the problem of efficiently proving the integrity of data stored at untrusted servers has received increased attention. The Provable Data Possession (PDP) model, the client preprocesses the data and then sends it to an untrusted server for storage, while keeping a small amount of meta-data. The client later asks the server to prove that the stored data has not been tampered with or deleted (without downloading the actual data). However, the original PDP scheme applies only to static (or append-only) files. They present a definitional framework and efficient constructions for Dynamic Provable Data Possession (DPDP), which extends the PDP model to support provable updates to stored data use a new version of authenticated dictionaries based on rank information. They provide a definitional framework and efficient constructions for Dynamic Provable Data Possession (DPDP), which extends the PDP model to support provable updates on the stored data. Given a file F consisting of n blocks, define an update as either insertion of a new block (anywhere in the file, not only append), or modification of an existing block, or deletion of any block. Therefore our update operation describes the most general form of modifications a client may wish to perform on a file.

F. Space-Efficient Block Storage Integrity

The new methods to provide block-level integrity in encrypted storage systems, i.e., so that a client will detect the modification of data blocks by an untrusted storage server. It shows cryptographic definitions for this setting, and develop solutions that change neither the block size nor the number of sectors accessed, an important consideration for modern storage systems.

In order to achieve this, a trusted client component maintains state with which it can authenticate blocks returned by the storage server, and we explore techniques for minimizing the size of this state. Demonstrate a scheme that provably implements basic block integrity (informally, that any block accepted was previously written), that exhibits a tradeoff between the level of security and the additional client's storage overhead, and that in empirical evaluations requires an average of only 0.01 bytes per 1024-byte block. Extend this to a scheme that implements integrity resistant to replay attacks (informally, that any block accepted was the last block written to that address) using only 1.82 bytes per block, on average, in our one-month long empirical tests. Address the storage integrity problem in this context. Due to the length preserving requirements for cryptographic operations on blocks, it is not possible to add information to each block (e.g., a MAC) in order to detect its modification, a fact explicitly noted in the SISW requirements.

Moreover, due to the performance demands of I/O intensive applications, it would be undesirable to put these MACs in separate blocks also stored at the service, which would require the retrieval of two blocks (one of data, one of MACs) on the critical

path of client read operations. Therefore, here consider a strategy in which a trusted client component— presumably the same one that holds keys for encrypting blocks before their transmission to the storage service, and for decrypting blocks upon their retrieval—holds this integrity information.

G. Compact Proofs of Retrievability

Proof-of-retrievability system, a data storage center must prove to a verifier that he is actually storing all of a client's data. The central challenge is to build systems that are both and provably secure that is, it should be possible to extract the client's data from any prover that passes a verification check.

They give the first proof-of-retrievability schemes with full proofs of security against arbitrary adversaries in the strongest model. Our first scheme, built from BLS signatures and secure in the random oracle model, has the shortest query and response of any proof -of-retrievability with public verifiability.

Our second scheme, which builds elegantly on Pseudorandom Functions (PRFs) and is secure in the standard model, has the shortest response of any Proof-of retrievability scheme with private verifiability (but a longer query). Both schemes rely on homomorphic properties to aggregate a proof into one small authenticator value.

Recent visions of cloud computing" and \software as a service" call for data, both personal and business, to be stored by third parties, but deployment has lagged. Users of outsourced storage are at the mercy of their storage providers for the continued availability of their data.

H. Evaluation: Formal Security Models

Such proof-of-storage systems should be evaluated by both systems" and crypto" criteria. Systems criteria include:

1. The system should be as possible in terms of both computational complexity and communication complexity of the proof-of- storage protocol, and the storage overhead on the server should be as small as possible;
2. The system should allow unbounded use rather than imposing a priori bound on the number of audit protocol interactions;
3. Verifiers should be stateless, and not need to maintain and update state between audits, since such state is difficult to maintain if the verifier's machine crashes or if the verifier's role is delegated to third parties or distributed among multiple machine.

I. Proofs of Retrievability via Hardness Amplification

Proofs of Retrievability (PoR), introduced by Juels and Kaliski allow the client to store a file F on an untrusted server, and later run an efficient audit protocol in which the server proves that it (still) possesses the client's data. Constructions of PoR schemes attempt to minimize the client and server storage, the communication complexity of an audit, and even the number of file blocks accessed by the server during the audit. Identify several different variants of the problem (such as bounded-use vs. unbounded-use, knowledge-soundness vs. information-soundness), and giving nearly optimal PoR schemes for each of these variants.

IV. Problem Description and Proposed Work

A. Problem description:

Provable Data Possession (PDP) schemes evolved around public and single clouds offer a publicly accessible remote interface to check and manage the tremendous amount of data. The majority of existing PDP schemes is incapable of satisfying such an inherent requirement of hybrid clouds in terms of bandwidth and time. Although existing schemes can make a false or true decision for data possession without downloading data at untrusted stores, they are not suitable for a distributed cloud storage environment. In existing PDP scheme didn't have auto blocking mechanism to user data which stored in to cloud server and did not useful for the large amount of data. In existing PDP scheme server can generate tag for multiple file blocks in term of single response value on client side, but the response from multiple cloud can be combined into single response value. For lack of homomorpic response the PDP protocol to check the integrity of file block stored in multi cloud server. Also client need to known exact position of file block stored into multi cloud so verification process in case will lead to high communication overhead and computation cost. Existing scheme RSA algorithm are used for key generation.

B. RSA Scheme

The RSA relies on the fact that it is easy to multiply two large prime number together but extremely hard to factor them back from the result. RSA is a block cipher in which the plain text and cipher text are integer between 0 ang n-1. The RSA is public key cryptosystem that is based on the intricacy of integer factoring. The RSA public key encryption method is the first instance of a provably secure public key encryption method against preferred massage attacks. Assuming that the factoring trouble is computationally obstinate and it is rigid to uncover the prime factor of n = p * q. The RSA method is describe as:

Key generation algorithm :

To generate the key entity A have to do the following :

1. by chance and secretly choose two large prime number p and q.
2. Compute the modulus p*q.
3. Compute ø(n) = (p-1) * (q-1).
4. Select chance integer e, 1 <e <n where gcd(e, ø) = 1.
5. Baghdad method[16] used to calculate the single decryption key d, 1<d< ø(n) where e*d = 1mod ø (n)
6. Determine public key and private key for entity A, the pair (e, n) as a public key (d, n) as private key.

Public key encryption algorithm Message m encrypt by entity B for entity A which entity A decrypts to it.

Encryption : entity B should do following :

1. Obtain entity A public key (e, n).
2. Message m as an integer in the interval {0….n-1}.
3. Calculate c = m^e mod n.
4. Send the encrypted message c to A.

Decryption : To recover the message m fron the cipher text c. Entity A must do the following :

1. Get the cipher text c from entity B
2. recover the message m = c^d mod n

Disadvantages:-

1. This algorithm has some limitation alongside certain attacks (i.e. Brute

force, Mathematical attack, Timing attacks and Chiper-text attacks).

2. In existing system doesn't have feature of automatic blocking the cloud server.

3. Existing system are less secure because of no modern cryptographic technique are used.
4. There are no feature to prove integrity based on public key or any other key while based on file name.
5. The details of attackers are not dynamic store but use the log file to store details and used data mining concepts to viewing it, that is time consuming job and less security.
6. Cloud user data store in untrusted cloud servers.
7. The data integrity is proving only based on the file name and not on the public key or any Other key
8. For lack of homomorphic responses, the verification process in such a case will lead to high Communication overheads and computation costs at client sides as well.

V. Conclusion

We projected the structure of an brilliant PDP technique for circulated cloud storage. On the basis on confusion index pecking order and homomorphic confirmable response we estimated a capable helpful PDP (ECPDP) technique, that's maintain to active inquiry such as placing, removal append and alteration etc on manifold storage servers. We also explained that our technique offer all safety resources follow to zero information interactive evidence system, therefore that it can resist different attacks which deployed in cloud as a public audit service. Moreover, we optimized the probabilistic indecision and intermittent confirmation to recover the audit presentation. These experiments obviously established that our approaches only introduce a less amount of communication and calculation outlay. These method are more appropriate for storing the big amount of information in multi cloud server. An Efficient accommodating verifiable Data Possession scheme to support dynamically scalability on manifold storage server.

VI. Future Work

In our resourceful CPDP plan Efficient RSA algorithm are used to key production and MD5 algorithm for tag generation. Some one used to more successful algorithm to get better it. Trusted Third Party are used to monitoring all this procedure it should be able to make normal verify on the integrity and accessibility of these delegated data at suitable interval and should be able to organize, manage and preserve the outsourcing data.

REFERENCES:

[1] Rajkumar Buyya, Christian Vecchiola and S. Thamarai Selvi "Mastering Cloud Computing".

[2] G. Joon Ahn, Y.Zun, H. Hu, "Cooperative provable data possession for Interiority verification in multi cloud storage," IEEE Transaction on parallel and distributed system, vol: PP, issue 99, 14-02-2012.

[3] I. M. Llorente, I. T. Foster, R. S. Montero, B. Sotomayor, "Virtual infrastructure management in private and hybrid clouds," IEEE Internet Computing, vol.13, no. 5, pp. 14-22-2009.

[4] J. Herring, L. Kissner, Z. N. J. Peterson, G. Ateniese, R. C. Burns, R. Curtmola, and D. X. Song, "Provable data possession at untrusted stores,"

ACM Conference on Computer and Communications Security, P. Ning, S.D.C. di Vimercati, and P. F. Syverson, Eds. ACM, 2007, pp. 598-609.

[5] P. Ning, S. D. C. di Vimercati, A. Juels and B. S. K. Jr., "Proofs of retrievability for large files," ACMConference on Computer and Communications Security P. F. Syverson, Eds. ACM, 2007, pp. 584-597.

[6] Pankaj Sareen "Cloud Computing: Types, Architecture, Applications, Concerns and Virtualization and Role of IT Governance in cloud", IJACCSSE, Volume-03, issue-03 pp. 533-538, March 2013.

[7] R. D. Pietro and G. Tsudik, G. Ateniese, L. V. Mancini, "Scalable and efficient provable data possession," Proceedings of the 4th international conference on Security and privacy in communication netowrks, Secure Comm, 2008, pp. 1-10.

[8] C. Papamanthou, A. K¨upc¸ ¨u, C. C. Erway and R. Tamassia, "Dynamic provable data possession," in ACM Conference on Computer and Communications Security, E. Al-Shaer, S. Jha, and A. D. Keromytis, Eds. ACM, 2009, pp. 213-222.

[9] B. Waters, H. Shacham "Compact proofs of retrievability," ASIACRYPT, ser. Lecture Notes in Computer Science, J. Pieprzyk, Ed., vol. 5350. Springer, 2008, pp. 90-107.

[10] C.Wang and W. Lou, Q. Wang, J. Li, K. Ren, "Enabling public verifiability and data dynamics for storage security in cloud computing," ESORICS, ser. Lecture Notes in Computer Science, M. Backes and P. Ning, Eds., vol. 5789. Springer, 2009, pp. 355-370.

[11] H. Hu, and S. S. Yau, Y. Zhu, H. Wang, Z. Hu, G.-J. Ahn, "Dynamic audit services for integrity verification of outsourced storages in clouds," SAC, W. C. Chu, W. E. Wong, M. J. Palakal, and C.-C. Hung, Eds. ACM, 2011, pp. 1550-1557.

[12] A. Oprea and A. Juels, K. D. Bowers, "Hail: a high-availability and integrity layer for cloud storage," ACM Conference on Computer and Communications Security. ACM, 2009, pp. 187-198.

[13] S. P. Vadhan, and D. Wichs, Y. Dodis, "Proofs of retrievability via hardness amplification," TCC, ser. Lecture Notes in Computer Science, O. Reingold, Ed., vol. 5444. Springer, 2009, pp. 109-127.

[14] L. Fortnow, M. Sipser and J. Rompel "On the power of multiprover interactive protocols," Theoretical Computer Science, 1988, pp. 156-161.

[15] G.-J. Ahn, Y. Zhu, H. Hu, Y. Han, and S. Chen, "Collaborative integrity verification in hybrid clouds," in IEEE Conference on the 7th International Conference on Collaborative Computing*: October 15-18*, 2011, pp. 197-206.

[16] Sattar Aboud, "Baghdad Method for Calculating Multiplicative Inverse", International Conference on Information Technology, Las Vegas, Nevada, USA. Pp: 816-819, 2004

[17] A. Fox, R. Griffith, M. Armbrust, A. D. Joseph, R. H. Katz, G. Lee, A. Konwinski, D. A. Patterson, A.

Rabkin, I. Stoica, and M. Zaharia, "Above the clouds: A berkeley view of cloud computing," EECS Department, University of California, Berkeley, Tech. Rep., Feb 2009.

[18] M. Franklin D. Boneh, "Identity-based encryption from the weil pairing," in *Advances in* Cryptology *(CRYPTO'2001)*, vol. 2139 of LNCS, 2001, pp. 213-229.

[19] O. Goldreich, *Foundations of Cryptography: Basic Tools*. Cambridge University Press, 2001.

[20] http://www.*javacodegeeks*.com/2013/08/writing-a-hadoop-mapreduce-task-in-java.html

* * * * *