International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 7, July - 2013

A Comparative Survey On Various Provable Data Possession Schemes Of Integrity Verification In Single And Multi Cloud Environments

Kochumol Abraham

Assistant Professor, P G Dept. of Computer Applications, Marian College, Kuttikkanam

Win Mathew John

Associate Professor, P G Dept. of Computer Applications, Marian College, Kuttikkanam

Abstract

This paper surveys the various provable data possession schemes that verify remote data integrity. These protocols have been proposed as a primitive for ensuring the long-term integrity and availability of data stored at remote untrusted hosts. Externalizing data storage to multiple network hosts is becoming widely used in several distributed storage and P2P systems, which urges the need for new solutions that provide security properties for the remote data. Replication techniques cannot ensure on their own data integrity and availability, since they only offer probabilistic guarantees. Moreover, peer dynamics and their potential misbehavior exacerbate the difficult challenge of securing remote data. To this end, remote data integrity verification protocols have been proposed with the aim to detect faulty and misbehaving storage hosts, in a dynamic and open setting. In this survey, we analyze several of these protocols, compare them with respect to expected security guarantees and discuss their advantages and limitations. In this work we surveyed data integrity proving schemes by reviewing different schemes in the area along with their efficiency and security considerations.

Keywords **:**

Provable Data Possession, Integrity Verification

1. Introduction

As cloud computing is a powerful network architecture for on-demand network access to computing resources such as servers, storage, applications and services that can be quickly deployed, the use of it is certainly on the rise. The CSPs provides data access and storage services in a pay-as-you-go manner. Users do not need to know about the physical location and configuration of the system that delivers the services. This elasticity of resources, without any pre investment, attracts more and more people to join the cloud storage. Cloud providers are offering efficient on-demand storage solutions that can virtually scale indefinitely. Recently, data generation is outpacing users' storage availability, thus there is an increasing need to outsource such huge amount of data. Outsourcing data to a remote Cloud Service Provider (CSP) is a growing trend for numerous customers and organizations alleviating the burden of local data storage and maintenance. Moreover, customers rely on the data replication provided by the CSP to guarantee the availability and durability of their data. Thus it's of crucial importance to customers to have a strong evidence that they actually get the service they pay for. Moreover, they need to verify that all their data copies are not being tampered with or partially deleted over time. While process transformation via the cloud is a key to achieving real and lasting benefits, getting there will not be simple. It requires great effort to develop innovative strategies and plans to redefine and overhaul operating models and processes in order to take advantage of cloud capabilities. Otherwise, these transformational benefits will remain vague and aspirational.

2. Multi Cloud Architecture

In a multi cloud environment the cloud architecture is made up of multi clouds in which in a big cloud there are different sub-clouds or inter-clouds. Each sub cloud will store the data and the user can access the stored information from any cloud. The data from all the sub clouds can be maintained and can be accessed by the user from a database management system. This is known as the multi cloud database system MCDB . In multi-clouds the solutions for the security issues like data integrity, intrusion and service providence have to be considered. But if the hacker knows the cloud from which the users access his data he can easily hack it. So the security for the data storage has not been achieved to a complete level. There exist various tools and technologies for multi cloud, such as Platform VM Orchestrator, VMware, VSphere, and Ovirt. These tools help cloud providers construct a distributed cloud storage platform (DCSP) for managing clients' data.

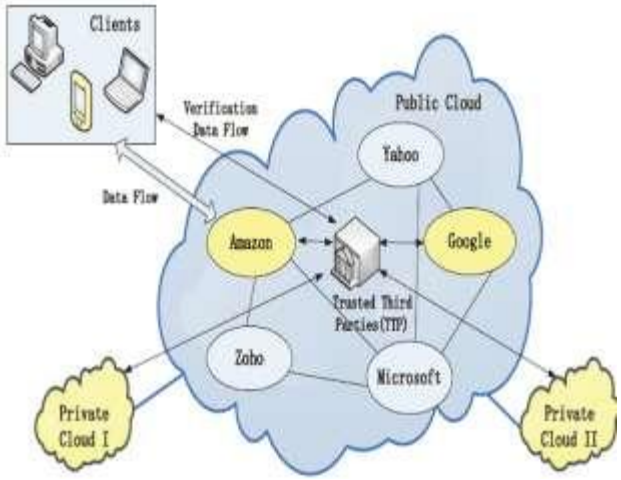

Figure 1: Multi cloud architecture

However, this new environment could bring irretrievable losses to the clients due to lack of integrity verification mechanism for distributed data outsourcing.

3. Security and privacy challenges

Cloud computing exists at the forefront of technology modernization, widely accepted as the obvious path toward IT efficiency, yet security concerns continue to be a significant hurdle for mainstream cloud adoption. Implementing a cloud computing strategy means placing critical data in the hands of a third party, Ensuring that your data is securely protected both at rest and in transit, restricting and monitoring access to that data via user authentication and access logging, and adequately planning for the very real possibilities of compromised or inaccessible data due to data breaches or natural disasters are all key security challenges that a company must address when considering cloud computing providers. While there are some key security advantages, there are just as many if not more security challenges that prevent customers from committing to a cloud computing strategy. Choosing a cloud provider that considers the security of your data as a major concern is an important matter. The security measures provided vary from provider to provider and among the various types of clouds. Certain providers have standard terms and conditions that may answer all the questions such as what methods of protection do they have? Will they have backups of our data? What barriers are in place to keep your information separate from other companies?

The ultimate challenge in cloud computing is data-level security, and sensitive data is the domain of the enterprise, not the cloud computing provider. Cloud does not differentiate between a sensitive data from a common data thus enabling anyone to access those sensitive data's. Thus there is a lack of data integrity in cloud computing. Thus there is a high possibility that the data can be stolen from the external server by a malicious user. Since customers' data has been outsourced to remote servers, efficient verification of the completeness and correctness of the outsourced data becomes a formidable challenge for data security in CC. The traditional cryptographic primitives for data integrity and availability based on hashing and signature schemes are not applicable on the outsourced data without having a local copy of the data. Of course, it is impractical for the clients to download all stored data in order to validate its integrity; this would require an expensive I/O cost and immense communication overheads across the network.

4. Integrity Verification

It is a crucial demand of customers to have strong evidence that the cloud servers still possess their 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 by cloud customers. Integrity requires that authorized changes must be detected and tracked, and changes must be limited to a specific scope. Managing an entity's admittance and rights to specify enterprise cloud resources ensures that valuable data and services are not abused. Due to the increased number of entities and access points in a cloud environment, authorization is crucial in assuring that only authorized entities can interact with data. A cloud computing provider is trusted to maintain data integrity and accuracy. To verify integrity, one must examine the net effects on the control cloud data related to data integrity. There will be a set of standards for monitoring the integrity of data. Integrity monitoring is essential in cloud storage as data integrity is critical for any data centre. Here, we consider the existence of multiple CSPs to collaboratively store and maintain the clients' data. Moreover, there are lots of PDP techniques to verify the integrity and availability of stored data in CSPs. Integrity verification should be made by clients to assure that their data has been properly stored and maintained in third party server. The availability and durability of data is provided by multi-cloud concept.

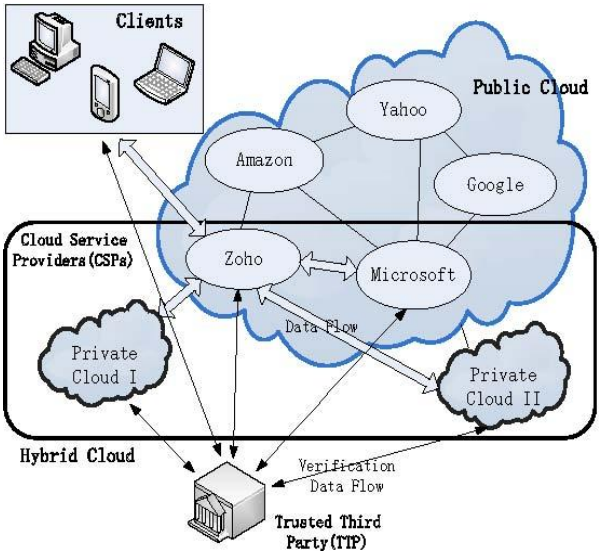

Figure 2: Integrity verification in multi cloud

5. Provable data possession

Provable data possession is a technique for ensuring the integrity of data in outsourcing storage service. It's a technique for a storage provider to prove the integrity and ownership of clients' data without downloading data. The proof-checking without downloading makes it especially important for large-size files and folders (typically including many clients' files) to check whether these data have been tampered with or deleted without downloading the latest version of data. Thus, it is able to replace traditional hash and signature functions in storage outsourcing. In fact, PDP is essentially an interactive proof between a CSP and a client because the client makes a false/true decision for data possession without downloading data. Existing PDP schemes could provide an efficient integrity checking for outsourced data; however, most of these schemes ignore the problem of information leakage among the interactive processes. Thus, as a public verification service without a strong security mechanism for data protection, a malicious attacker could easily exploit such a service to obtain private data. This attack is extremely dangerous to the confidential data of an enterprise.

6. Related Work

6.1 Static Provable Data Possession (PDP) Schemes

The fundamental goal of the PDP scheme is to allow a verifier to efficiently, periodically, and securely validate that a remote server which is supposed to store the owner's potentially very large amount of data is not cheating the verifier. The problem of data integrity over remote servers has been addressed for many years and there is a simple solution to tackle this problem as follows.

6.1.1 Basic PDP Scheme based on MAC

The data owner

a.) Computes a Message Authentication Code (MAC) of the whole file before outsourcing to a remote server.

b.) Keeps only the computed MAC on his local storage, sends the file to the remote server, and deletes the local copy of the file.

c.) Later, whenever a verifier needs to check the data integrity, he sends a request to retrieve the file from the archive service provider, re-computes the MAC of the whole file, and compares the re-computed MAC with the previously stored value**.**

Limitation

MAC of the whole file should be computed

Alternatively, instead of computing and storing the MAC of the whole file, the data owner

1. Divide the file F into blocks {b1, b2, . . . , bm},
2. Computes a MAC σi for each block bi : σi = MAC sk (i||bi)1≤i≤m,
3. Sends both the data file F and the MACs {σi}1≤i≤m to the remote/cloud server, deletes the local copy of the file, and stores only the secret key sk. **[1]**

During the **verification** process, the verifier requests for a set of randomly selected blocks and their corresponding MACs, re-computes the MAC of each retrieved block using sk, and compares the re-computed MACs with the received values from the remote server.

Advantage

The checking part of the file is much easier than the whole of it.

Disadvantage

The communication complexity is linear with the queried data size which is impractical especially when the available bandwidth is limited.

6.1.2 PDP Schemes based on functions f and $H^{|}$

$H^{|}$ is a one-way function, f is another function such that f(C, $H^{|}$ (File)) = H (C||File), where

H - Secure hash function and

C - Random challenge number sent from the verifier to the remote server. [2]

The protocol is as follows:

1. Data owner computes $H^{|}$ (File) and store it on the local storage.[2]
2. To audit the file,

   a.) Verifier generates a random challenge C,

b.) Computes V = f(C, H| (File)), and sends C to the remote server.

3. The server computes R = H(C||File) and sends the response R to the verifier.

4. To validate the file integrity, the verifier checks V ?= R.

Limitation

At least one of the two functions f and H| must be kept secret because if both were public, it would be easy for a malicious server to compute and store only H| (File) that is not the entire file, and then dynamically responds with a valid value f(C, H| (File)) that is not the expected one H(C||File).

Unfortunately there are no such functions f, H, and H| satisfying the desired verification rule.

2.) Alternatively,

1. Finite number eN of random challenges is generated offline for the file to be checked,

2. The corresponding responses H (Ci||File)1¡Âi¡Â eN are pre-computed offline as well, and then

3. Pre-computed responses are stored on the verifier local storage.

4. To audit the file, one of the eN challenges is sent to the remote server and the response received from the server is compared with the pre-computed response which is previously stored on the verifier side.

Limitations

- Limited number of audits per file.
- In each verification, the remote server has to do the exponentiation over the entire file.
- Storage overhead on the verifier side.

6.1.3 RSA Based PDP Schemes.

1. RSA-based Homomorphic hash function[3]

A function H is Homomorphic if, given two operation + and x, we have H (d+d|) = H(d) x H(d|).

The response R = H(d) is a homomorphic function in the data file d; $H(d + d|) = r^{d+ d|}\ r^{d}\ r^{d|} = H(d)H(d|)$ mod N.[3]

To find a collision for this hash function, one has to find two messages d, d′ such that $r^{d} \equiv r^{d'}$, i.e., $r^{d-d'} \equiv 1$ *mod N*. Thus, d − d′ must be multiple of ϕ(N). Finding such two messages d, d′ is believed to be difficult since the factorization of N is unknown.

Limitations

The archive service provider has to exponentiate the entire data file plus the storage overhead on the verifier side.

Solution is to use an RSA-based hash function on the blocks.

2. RSA-based hash function on the blocks

1. Fragment the file into blocks
2. Fingerprint each block, and then use an RSA-based hash function on the blocks.

Thus, the file F is divided into a set of m blocks:

F = {b1, b2, . . . , bm}, where m fingerprints {Mi}1¡Âi¡Âm are generated for the file and stored on the verifier local storage. Their proposal does not require the exponentiation of the entire file.

Limitation

Although the protocol does not require exponentiation of the entire file, a local copy of the fingerprints whose size is linear in the number of file blocks must be stored on the verifier side.

6.1.4 Data Storage Commitment Schemes.

A storage-enforcing commitment scheme (SEC) is a three-party protocol executed between a message source S, a prover P, and a verifier V. [4]

1. The message source communicates the message M to the prover and the commitment C to the verifier.
2. The verifier V may verify whether the prover is storing the secret by invoking a probabilistic interactive algorithm. This algorithm may be executed an unlimited number of times.
3. Once the message is revealed, the verifier may check the commitment by running the algorithm Verify.

This scheme has three properties called binding, concealing, and storage-enforcing.

Limitations

- It only ensures that the server is storing something at least as large as the original data file but not necessarily the file itself.
- In addition, the verifier's public key is about twice as large as the data file.

6.1.5 Privacy-Preserving PDP Schemes

a. The data owner first encrypts the file,

b. Sends both the encrypted file along with the encryption key to the remote server.

c. Moreover, the data owner sends the encrypted file along with a key-commitment that fixes a value for the key without revealing the key to the TPA.

The primary purpose of this scheme is to ensure that the remote server correctly possesses the client's data along with the encryption key, and to prevent any information leakage to the TPA which is responsible for the auditing task[5]. Thus, clients

especially with constrained computing resources and capabilities can resort to external audit party to check the integrity of outsourced data, and this third party auditing process should bring in no new vulnerabilities towards the privacy of client's data. In addition to the **auditing task** of the TPA, it has another primary task which is **extraction of digital contents**. [6]

Advantage

An external Third Party Auditor (TPA) can verify the integrity of files stored by a remote server without knowing any of the file contents.

Limitations

a. The number of times a particular data item can be verified is limited and must be fixed beforehand.

b. Storage overhead.

c. Lack of support for stateless verification.

d. Very high communication complexity.

6.1.6 PDP in Database Context based on signature aggregation

Each database record is signed before outsourcing the database to a remote service provider.[7]

2 aggregation mechanisms:

1. Scheme based on RSA [8] and
2. Scheme based on BLS signature [9]

Scheme based on the RSA signature

Each record in the database is signed as: $\sigma i = h(bi)d \bmod N$ where h is a one-way hash function, bi is the data record, d is the RSA private key, and N is the RSA modulus.

a.) A user issues a query to be executed over the outsourced database, the server processes the query and computes an aggregated signature $\sigma = \Sigma ti=1\ \sigma i \bmod N$, where t is the number of records in the query result. The server sends the query result along with the aggregated signature to the user.

c.) To verify the integrity of the received records, the user checks $\sigma e = \pi ti=1\ \sigma i \bmod N$, where e is the RSA public key.

Scheme based on the BLS signature is similar to the first scheme but the record signature $\sigma i = h(mi)x$, where $x \in R\ Zp$ is a secret key.

Limitations

- Although data integrity (correctness) is an imperative concern in the database outsourcing paradigm, completeness is another crucial demand for database users. Completeness means that the service provider should send all records that satisfy the query criteria not just subset of them. This scheme did not fulfill the completeness requirement
- We emphasize that the techniques based on aggregated signature would fail to provide block less verification

6.1.7 PDP Schemes Based on Homomorphic Variable Tags(HVTs)/Homomorphic Linear Authenticators(HLAs).

HVTs/HLAs are unforgeable verification metadata constructed from the file blocks in such a way that the verifier can be convinced that a linear combination of the file blocks is accurately computed by verifying only the aggregated tag/authenticator.

Public verifiability and private verifiability.

In public verifiability anyone not necessarily the data owner who knows the owner's public key can challenge the remote server and verify that the server is still possessing the owner's files. On the other side, private verifiability allows only the original owner (or a verifier with whom the original owner shares a secret key) to perform the auditing task.

Two main PDP schemes:

Sampling PDP (S-PDP) and Efficient PDP (E-PDP) schemes. [10]

Based on KEA1 assumption (Knowledge of Exponent Assumption). It focuses on the problem of auditing if an untrusted server stores a client's data.

Advantages

They incur a low (or even constant) overhead at the server and require a small, constant amount of communication per challenge. Key components of these schemes are the support for spot checking, which ensures that the schemes remain lightweight, and the homomorphic verifiable tags, which allow to verify data possession without having access to the actual data file.

Limitations

In fact, there is a slight difference between the S-PDP scheme and the E-PDP model, but the E-PDP model provides weaker guarantee of data possession.

The E-PDP scheme only guarantees possession of the sum of file blocks and not necessarily possession of each one of the blocks being challenged.

6.2 Dynamic Provable Data Possession (DPDP)

The PDP schemes discussed above focus on static or warehoused data which is essential in numerous different applications such as libraries, archives,

and astronomical /medical /scientific /legal repositories. On the other side, Dynamic Provable Data Possession (DPDP) schemes investigate the dynamic file operations such as update, delete, append, and insert operations. There are some DPDP constructions in the literature satisfying different system requirements.

6.2.1 Scalable DPDP

This scheme is based entirely on symmetric-key cryptography. [11]

1. Before outsourcing, data owner pre-computes a certain number of short possession verification tokens, each token covering some set of data blocks. The actual data is then handed over to server.

2. Subsequently, when data owner wants to obtain a proof of data possession, it challenges server with a set of random-looking block indices.

3. In turn, server must compute a short integrity check over the specified blocks (corresponding to the indices) and return it to data owner.

For the proof to hold, the returned integrity check must match the corresponding value pre-computed by data owner. However, in this scheme data owner has the choice of either keeping the pre-computed tokens locally or outsourcing them – in encrypted form – to server. Notably, in the latter case, data owner's storage overhead is constant regardless of the size of the outsourced data. This scheme is also very efficient in terms of computation and bandwidth.

Advantages

1. Requires no bulk encryption of outsourced data and no data expansion due to additional sentinel blocks.

2. Supports secure and efficient dynamic operations on outsourced data blocks.

Limitations

1. Number of updates and challenges is limited and fixed in advance.
2. It does not support block insertion operation

6.2.2 DPDP-I

Given a file F consisting of n blocks, we 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 the update operation describes the most general form of modifications a client may wish to perform on a file. DPDP solution is based on a new variant of authenticated dictionaries, where rank information is used to organize dictionary entries[12]. Thus its able to support efficient authenticated operations on files at the block level, such as authenticated insert and delete. The security of the constructions using standard assumptions has been proved.

Advantages

1. Supports efficient authenticated operations on files at the block level, such as authenticated insert and delete.
2. Supports data possession guarantees of a hierarchical file system as well as file data.

Limitations

It does not support efficient verification of the indices of the blocks, which are used as query and update parameters .

6.2.3 DPDP II

The only difference between the two schemes is the authenticated structure used for protecting the integrity of the tags. It has a higher probability of detection and maintains logarithmic communication complexity but has increased update time[13]. A dynamic authenticated data structure called RSA tree is presented here that achieves constant expected query time (i.e., time to construct the proof), constant proof size, and $O(n^{\epsilon} \log n)$ expected amortized update time, for a given $0 < \epsilon < 1$. We can add rank information to the RSA tree by explicitly storing ranks at the internal nodes. Using this data structure allows the server to answer O (log n) challenges with O(log n) communication cost since the proof for a block tag has O (1) size.

Limitations

It has increased update time.

6.3 DPDP Schemes in Hybrid clouds

6.3.1 Collaborative PDP

Homomorphic verifiable response is the key technique of collaborative PDP because it not only reduces the communication bandwidth, but also conceals the location of outsourced data in hybrid clouds. The collaborate integrity verification for distrusted outsourced data, in essence, is a multi-prover interactive proof system (IPS), so that the corresponding construction should satisfy the security requirements of IPS. Moreover, in order to ensure the security of verified data, this kind of construction is also a Multi-Prover Zero-knowledge Proof (MPZKP) system which can be considered as an extension of the notion of an interactive proof system.

Given an assertion L, such a system satisfies three following properties:

(1) **Completeness**: whenever $x \in L$, there exists a strategy for provers that convinces the verifier

(2) **Soundness**: whenever x 6∈ L, whatever strategy the provers employ, they will not convince the verifier that x ∈ L;

(3) **Zero-knowledge**: no cheating verifier can learn anything other than the veracity of the statement.

Limitations

Latency overhead and scalability of prototype has not been described.

6.3.2 Scalia

A cloud storage brokerage solution that continuously adapts the placement of data based on its access pattern and subject to optimization objectives, such as storage costs. Scalia efficiently considers repositioning of only selected objects that may significantly lower the storage cost.Scalia can run directly at the customer premises as an integrated hardware and software solution (i.e., an appliance) or can be deployed as a hosted service across several data centers, putting the emphasis on providing a scalable and highly available architecture with no single point of failure, able to guarantee higher availability than the storage providers.

Limitations

Latency overhead and scalability of prototype has not been described

6.4 Multi-Copy PDP Schemes (MC-PDP Schemes)

Suppose that a CSP offers to store n copies of an owner's file on n different servers to prevent simultaneous failure of all copies. Thus, the data owner needs a strong evidence to ensure that the CSP is actually storing no less than n copies, all these copies are complete and correct, and the owner is not paying for a service that he does not get. A solution to this problem is to use any of the previous PDP schemes to separately challenge and verify the integrity of each copy on each server. This is certainly not a workable solution; cloud servers can conspire to convince the data owner that they store n copies of the file while indeed they only store one copy.

Whenever a request for a PDP scheme execution is made to one of the n severs, it is forwarded to the server which is actually storing the single copy. The CSP can use another trick to prove data availability by generating the file copies upon a verifier's challenge; however, there is no evidence that the actual copies are stored all the time. The main core of this cheating is that the n copies are identical making it trivial for the servers to deceive the owner. Therefore, one step towards the solution is to leave the control of the file copying operation in the owner's hand to create unique distinguishable/differentiable copies.

6.4.1 Basic Multi-Copy Provable Data Possession (BMC-PDP) scheme

The data owner creates n distinct copies by encrypting the file under n different keys keeping these keys secret from the CSP. Hence, the cloud servers could not conspire by using one copy to answer the challenges for another. This natural solution enables the verifier to separately challenge each copy on each server using any of the PDP schemes, and to ensure that the CSP is possessing not less than n copies.

Limitations

- The computation and communication complexities of the verification task grow linearly with the number of copies.
- Key management is a severe problem with the BMC-PDP scheme.

6.4.2 Multiple-Replica Provable Data Possession (MR-PDP) scheme

Creating distinct replicas/copies of the data file by first encrypting the file then masking the encrypted version with some randomness generated from a Pseudo-Random Function (PRF) is being performed here[14].

Limitations

- Long tags
- Computation overhead on both the verifier and server side,
- Ability of CSP to cheat by using blocks from different files if the data owner uses the same secret key (d, v) for all his files.
- It does not address how the authorized users of the data owner can access the file copies from the cloud servers.
- The MR-PDP supports only private verifiability, where just the data owner (or a verifier with whom the original owner shares a secret key) can do the auditing task.

6.4.3 Efficient Multi-Copy Provable Data Possession (EMC-PDP) schemes

It's based on HLA's. In EMC-PDP models we resort to the diffusion property of any secure encryption scheme. Diffusion means that the output bits of the cipher text should depend on the input bits of the plain text in a very complex way. In an encryption scheme with strong diffusion property, if there is a change in one single bit of the plaintext, then the cipher text should completely change in an unpredictable way. This methodology of generating

distinct copies is not only efficient, but also successful in solving the authorized users problem of the MRPDP scheme to access the file copy received from the CSP. The two versions of the EMC-PDP schemes are[15]:

i.) Deterministic EMC-PDP (DEMC-PDP) scheme

ii.) Probabilistic EMC-PDP (PEMC-PDP) scheme

In the DEMC-PDP version, the CSP has to access all the blocks of the data file, while in the PEMC-PDP, spot checking is performed by validating a random subset of the file blocks. It is a trade-off between the performance of the system and the strength of the guarantee provided by the CSP. In the PEMC-PDP scheme, we use the same indices for the challenged blocks across all copies. The rationale behind the PEMC-PDP scheme is that checking part of the file is much easier than the whole of it, and thus reducing the computation and storage overhead on the servers side.

Limitations

Storage and computation cost is larger.

6.4.4 Pairing based provable multi-copy data possession (PB-PMDP) scheme

This scheme provides an adequate guarantee that the CSP stores all copies that are agreed upon in the service contract, and these copies are intact. The authorized users can seamlessly access the copies received from the CSP. The PB-PMDP scheme supports public verifiability.

Generating unique differentiable copies of the data file is the core to design a multi-copy provable data possession scheme[16]. Identical data 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. 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 and access a file copy received from the CSP without recognizing the copy index.

Homomorphic linear authenticators (HLAs) are basic building blocks in the proposed scheme.

Limitations

While identifying corrupted copies, the cost of extra storage, communication, and computation overheads occurs.

6.4.5 Distributed and Replicated (DR-DPDP) scheme

DR-DPDP is a scheme that provides transparent distribution and replication of user data over multiple servers. There are three entities in the model. The client, who stores data on the CSP, challenges the CSP to check the integrity of data, and updates the stored data[17]. The organizer, who is one of the servers in CSP and is responsible for communication with the client and other servers (acts as a gateway or load-balancer). The servers, who store the user data, perform provable updates on behalf of the client, and respond to the client challenges coming via the organizer. They only communicate with the organizer and there is no inter-server communication.

It is very important to observe that even though it seems like a central entity, the organizer is not expected to perform any disk operations or expensive group operations (e.g., exponentiation). He will only perform simple hashing, and work with a very small skip list. Hence, his load will be very light, making it very easy to replicate the organizer to prevent it from becoming a bottleneck or single-point-of-failure.

Limitations

The computation time in the organizer becomes greater than that of the servers.

7. Comparative Study

This comparative study provides a consolidated report of all the techniques of the PDP schemes Static, Dynamic and multi-copy PDP Schemes in single and hybrid clouds.

Comparative study of Static, Dynamic and multi-copy PDP Schemes

Static PDP Schemes			
Scheme	**Technique used**	**Advantage**	**Disadvantage**
Basic	MAC	Secure and efficient	The communication complexity is linear with the queried data size which is impractical especially when the available bandwidth is limited.
PDI	Secure hash function	Checking part of the file is much easier than the whole of it.	1. Limited number of audits per file. 2. In each verification, the remote server has to do the exponentiation over the entire file. 3. Storage overhead on the verifier side.
Challenge-response	RSA Homomorphic hash function	The freshness of the response computation by the server is guaranteed by the fact that a challenge is never reused before reboot of the server.	Although the protocol does not require exponentiation of the entire file, a local copy of the fingerprints whose size is linear in the number of file blocks must be stored on the verifier side.
Data Storage Commitment	n-Power Computational Diffie-Hellman (n-PCDH) assumption	Makes use of storage space as large as the client's data	1. It only ensures that the server is storing something at least as large as the original data file but not necessarily the file itself. 2. In addition, the verifier's public key is about twice as large as the data file.
Privacy Preserving	Encryption	An external Third Party Auditor (TPA) can verify the integrity of files stored by a remote server without knowing any of the file contents.	1. The number of times a particular data item can be verified is limited and must be fixed beforehand. 2. Storage overhead on the TPA 3. Lack of support for stateless verification 4. Very high communication complexity
Database	Signature Aggregation	Signature aggregation enables bandwidth and computation efficient integrity verification of query replies.	1. Does not fulfill the completeness requirement. 2. Fails to provide block less verification
S-PDP & E-PDP	HVT'S, HLA'S KEA1 assumption	Client is convinced of data possession, without actually having to retrieve file blocks. Provide data format independence. Offers public verifiability.	1. HVTs are based on RSA and thus are relatively long. 2. The time taken to generate the tags is too long. 3. Since there is no indicator for the file identifier in the block tag, a malicious server can cheat by using blocks from different files if the data owner uses the same secret keys.
Dynamic PDP Schemes			
Scalable PDP	Based on cryptographic Hash function & symmetric-key encryption	1. Requires no bulk encryption of outsourced data and no data expansion due to additional sentinel blocks. 2. Supports secure and efficient dynamic operations on outsourced data blocks.	1. Number of updates and challenges is limited and fixed in advance. 2. It does not support block insertion operation
DPDP I	Rank-based authenticated skip list	1. Supports efficient authenticated operations on files at the block level, such as authenticated insert and	It does not support efficient verification of the indices of the blocks, which are used as query and update parameters .

		delete. 2. Supports data possession guarantees of a hierarchical file system as well as file data.	
DPDP II	RSA trees	Blockless verification of data.	It has increased update time.
Dynamic PDP Schemes in Hybrid clouds			
Cooperative PDP	HVR, HIH, IPS, MPZKPS interactive proof system and multi-prover zero-knowledge proof system	1. Multi-prover zero-knowledge proof system (MP-ZKPS), which has completeness, knowledge soundness, and zero-knowledge properties. 2. It has security against data leakage attack and tag forgery attack.	Latency overhead and scalability of prototype has not been described.
Scalia	Multi-datacenter	High data duarability and minimizes the storage cost for clients	Latency overhead and scalability of prototype has not been described.
Multi-Copy PDP Schemes(MC-PDP Schemes)			
BMC-PDP	Encryption	It generate unique distinguishable/differentiable copies of the data file.	1. The computation and communication complexities of the verification task grow linearly with the number of copies. 2. Key management is a severe problem.
MR-PDP	Signature aggregation	Each unique replica can be produced at the time of the challenge it can generate further replicas on demand	1. It does not address how the authorized users of the data owner can access the file copies from the cloud servers. 2. Computation overhead on both the verifier and server side. 3. Storage overhead
DEMC-PDP	Bilinear Map/Pairing.	1. Strongest guarantee at the expense of the storage overhead. 2. Shortest verification time	Storage and computation cost is larger.
PEMC-PDP	Bilinear Map/Pairing	Lowest storage overhead on the server side by using spot checking.	Storage and computation cost is larger.
PB-PMDP	BLS HLAs	1. It provides an evidence to the customers that all outsourced copies are actually stored and remain intact. 2. It allows authorized to seamlessly access the file copies stored by the CSP Supports public verifiability. 3. Secure against colluding servers.	While identifying corrupted copies the cost of extra storage, communication, and computation overheads occurs.
DR-DPDP	Rank-based	It provides transparent	The computation time in the

	authenticated skip list	distribution and replication of user data over multiple servers.	organizer becomes greater than that of the servers.

8. References

1. K. Zeng, "Publicly verifiable remote data integrity," in ICICS, 2008, pp. 419–434.
2. Y. Deswarte, J-J. Quisquater, and A. Sa¨ıdane ,"Remote integrity checking," in 6th Working Conference on Integrity and Internal Control in Information Systems (IICIS), S. J. L. Strous, Ed., 2003, pp. 1–11.
3. D. L. G. Filho and P. S. L. M. Barreto, "Demonstrating data possession and uncheatable data transfer," Cryptology ePrint Archive, Report 2006/150, 2006.
4. P. Golle, S. Jarecki, and I. Mironov, "Cryptographic primitives enforcing communication and storage complexity," in FC'02: Proceedings of the 6th International Conference on Financial Cryptography, Berlin, Heidelberg, 2003, pp. 120–135.
5. M. A. Shah, M. Baker, J. C. Mogul, and R. Swaminathan, "Auditing to keep online storage services honest," in HOTOS'07: Proceedings of the 11th USENIX workshop on Hot topics in operating systems, Berkeley, CA, USA, 2007, pp. 1–6.
6. M. A. Shah, R. Swaminathan, and M. Baker, "Privacy-preserving audit and extraction of digital contents," Cryptology ePrint Archive, Report 2008/186, 2008.
7. E. Mykletun, M. Narasimha, and G. Tsudik, "Authentication and integrity in outsourced databases," Trans. Storage, vol. 2, no. 2, 2006.
8. R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," Commun. ACM, vol. 26, no. 1, 1983.
9. D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," in ASIACRYPT '01: Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security, London, UK, 2001, pp. 514–532.
10. G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in CCS '07: Proceedings of the 14th ACM Conference on Computerand Communications Security, New York, NY, USA, 2007, pp. 598–609.
11. G. Ateniese, R. Di Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in Secure Comm '08: Proceedings of the 4th International Conference on Security and Privacy in Communication Netowrks, New York, NY, USA, 2008, pp. 1–10.
12. Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," in ESORICS'09: Proceedings of the 14th European Conference on Research in Computer Security, Berlin, Heidelberg, 2009, pp. 355–370.
13. Z. H. G.-J. A. H. H. Yan Zhu, Huaixi Wang and S. S. Yau, "Cooperative provable data possession," Cryptology ePrint Archive, Report 2010/234, 2010.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, 2007, pp. 584–597.
14. R. Curtmola, O. Khan, R. Burns, and G. Ateniese, "MR-PDP: Multiple-Replica Provable DataPossession," in 28th IEEE ICDCS, 2008, pp. 411–420.
15. Ayad F.Barsoum and M.Anwar Hasan Provable Possession and Replication of Data over Cloud Servers.
16. Ayad F. Barsoum, M. Anwar Hasan. "Integrity Verification of Multiple Data Copies over Untrusted Cloud Servers"
17. Mohammad Etemad and Alptekin Koc University, Istanbul, Turkey." Transparent, Distributed, and Replicated Dynamic Provable Data Possession".