ISSN:0975-9646

A Comparative Analysis of Provable Data Possession Schemes in Cloud

Pooja Natu[#], Prof. Shikha Pachouly[*]

[#]*PG scholar AISSMS COE, Department of Computer Engineering,*
Savitribai Phule Pune University, Pune, India

[*] *Assistant Professor AISSMS COE, Department of Computer Engineering,*
Savitribai Phule Pune University, Pune, India

***Abstract*— In cloud computing environment, data owners usually host huge data on the cloud servers where clients access the data without knowing actual location. Due to this data outsourcing on un-trusted servers, efficient and reliable verification of the outsourced data becomes an open challenge in data security of Cloud Storage. Additionally, the integrity checking protocol must be efficient in order to save the verifier's cost. This triggered huge set of research activities, resulting in amount of proposals. Integrity verification of client data is achieved commonly by using a technique called Provable Data Possession (PDP). This paper provides overview of current variations in PDP technique by specifying models, functionality, strengths and weaknesses.**

***Keywords*— Cloud Computing, Provable Data Possession (PDP), Integrity Verification, Cloud Storage Security**

I. Introduction

Cloud computing is an internet based computing model which provides on-demand service, local independence, scalability, elasticity, ubiquitous network access, resource pooling and pay-as-you-go policies. Cloud Storage is one of the important services of cloud computing, which allows data owners to load data to the cloud. Data outsourcing is beneficial to small and medium sized businesses as it is cost effective solution. While making clients free from data storage burdens, cloud brings new and severe security threats in user's outsourced data.

The critical issue of data integrity comes whenever client uploads data on un-trustworthy servers. In such scenarios, clients need to implement strategies to prove originality of data. The client may need to access whole file to ensure data integrity, which is time and space consuming. Considering the huge size of the outsourced data and the users constrained resource it is not always possible to access complete data. In this paper, we investigate the approaches of Provable Data Possession (PDP) along with their attributes, functionality, pros and cons. In this paper we surveyed latest core integrity techniques in detail considering functionality used, advantages and disadvantages.

A. *Provable Data Possession Model*

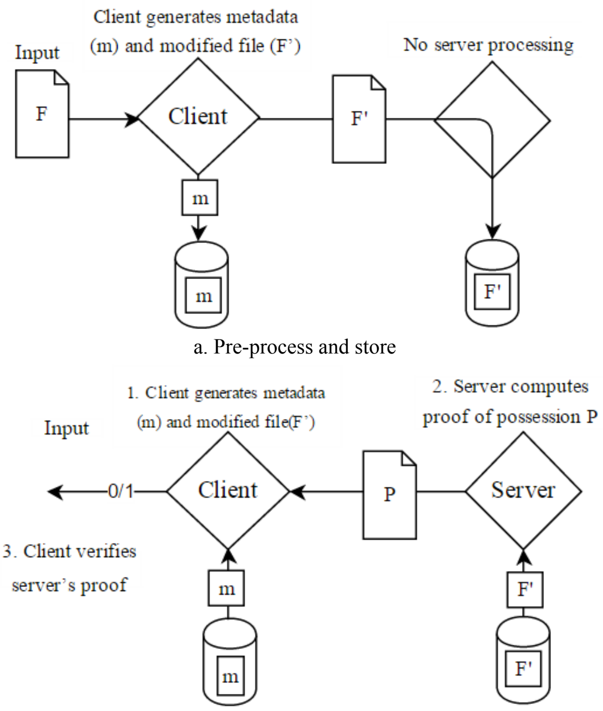
The Provable Data Possession (PDP) is one of the best techniques for ensuring data intactness when the client data is hosted on cloud server. In this technique, the client computes some metadata in order to ensure integrity of hosted data. The metadata is stored at client side and used later on for integrity verification by client. The server stores actual data along with appropriate metadata generated by client. Whenever the client asks for verification, server returns the response which is then verified by client. In order to improve the performance of the PDP technique, many schemes are proposed under various systems and security models in last some years.

The client with data hosted on cloud, requires guarantees about the authenticity of data on cloud, namely that storage servers possess data [1]. It is inadequate to detect that data have been altered when accessing the outsourced data, because it may be too late to recover damaged data. Additionally, the cloud service providers (CSPs) may try to hide data loss and claim that the data is still intact in the Cloud. Hence, data owners need to be convinced always that their data is correctly stored and intact in the Cloud. So, one of the critical concerns with outsourced data storage is that of data integrity verification. In order to overcome the problem of data integrity verification, many schemes are proposed under different systems and security models.

The author Giuseppe Ateniese et al. [1] proposed the model for provable data possession (PDP) which provides probabilistic proof that a third party stores a file intact. It allows the server to access small blocks or portions of the file to generate the proof without accessing the entire file [1].

The following figure (fig. 1) shows the working of PDP model where Fig.1 (A) explains the pre-processing and data hosting activities. The fig.1 (B) shows the verification phase of PDP model. The PDP model basically works with four important stages: Setup, Update, Challenge and Verify [1]. In setup phase, the client sets up with metadata, and then in case of Update phase the client may try to update the hosted data. In challenge phase server tries to generate the proof of originality where in Verify phase the client verifies the response of server.

The PDP approaches can also be executed with Third Party Auditors where there will be an extra operational entity as auditor. There are multiple approaches presented with auditing schemes. However, this paper limits the PDP schemes executed and controlled by client.



a. Pre-process and store

b. Verify server possession

Fig.1 Protocol for provable data possession [1]

B. *Objectives of PDP techniques*

The basic PDP technique can be enhanced by implementing various other concepts. The main objectives to be considered while designing PDP model are described here [9].

- Support for Data Dynamics

The older PDP techniques were designed only for static data. Moreover, clients were not allowed to update the file once it has been hosted on cloud storage. Wherever, in practice, clients need to update hosted files online and so PDP techniques are immerging to overcome this issue. The important data dynamics operations are inserting, updating existing data and deleting selected data from cloud storage.

- Computational Complexity

As data integrity checking is critical, the data integrity checking models must consider reduced and constant complexities at client and server ends. This is achieved in some proposed models with the help of special terminologies. The computational complexity can be reduced to constant so that clients can perform data verification periodically with less computational requirements.

- Block-less verification:

This ensures that challenged file blocks should be retrieved by the verifier during verification process for both efficiency and security purpose.

- Communication or network Overhead

The data verification includes sending and receiving data or group of data among client and storage server. This inturns increases network traffic on server. In this case, the models must try to reduce the network traffic as much as possible by reducing network communication.

- Type of variability

The cloud is public and so it is accessible to many clients. The Public verifiability allows any one (not just a client) to conduct the integrity checking test on any data.

- Privacy-Preserving Approach

When the verification is done by a third party verifier (not by owner or client), the protocol must ensure that no private information shared with third party is leaked. In Batch auditing, multiple auditing tasks from different users can be performed simultaneously by the third party auditor to improve the model performance. It is necessary to maintain privacy among client and third party auditor in case of distributed workload. In this approach, we can have dumb or less intelligent clients with very less processing and hardware resources.

- Unboundedness

The PDP technique deals with request response cycles and hence the number of verifications allowed must not be limited .The client must verify his/her data for any number of times.

C. *Definitions in PDP technique*

The PDP scheme gives following definitions of algorithms [1]. It is a collection of four polynomial time algorithms. The algorithms can be given as:

KeyGen(1^K) → (pk,sk)

Is a probabilistic key generating algorithm and run by client in Setup. It generates the public key and secret key for client and server.

TagBlock(pk, sk, m) → T_m

Is algorithm run by the client to create verification metadata. In this, the file blocks are assigned with the tags which are used for verification.

GenProof(pk, F, chal, ∑) → V

Is called by the server to generate a proof of challenged possession.

CheckProof (pk, sk,Chal, V) → {success, failure}

Is run by the client to verify whether server generated proof.

The above techniques are called by client and server in different steps to perform data integrity verification. The following contents are arranged as: section 2 specifies the review of PDP models. The section 3 provides comparative analysis of PDP schemes. Finally, we have concluded the work by specifying best approach of PDP.

II. Related Work

The data intactness is an important property to be retained in case of outsourced data. The very first approach proposed the PDP model [1] for data possession on un-trusted storages without downloading the actual data and provided an RSA-based scheme for a static data. It also includes a public verifiability, where anyone can challenge the server for data possession. This extends the application areas of PDP protocol by separating the data owners and the users. But this is insecure against attacks in dynamic scenarios. Additionally, they do not support multi-cloud storage because there is no homomorphism property in the verification process. To overcome this static file storage limits in PDP and to provide dynamic data operations in PDP the Scalable PDP [8] model have been proposed. It is a lightweight scheme based on cryptographic hash function and symmetric key encryption. But fails in randomness in the challenges and by using previous metadata, the servers can deceive the owners. The other flaw in this model is block size and its length are fixed and hence the modification of blocks cannot be done anywhere.

With this background, the Yongjun Ren, et al. [3] proposed designated verifier approach for PDP where the verification overhead is shifted at third separate module which makes client free from computations. Based on this work two Dynamic PDP [4] has been proposed. In this model data dynamics was achieved with the help of HVTs and hash values. Then the Robustness was added to PDP scheme by providing RS code based on Cauchy Matrices by author Bo Chen, et al. [5] who proposed two approaches towards Robust DPDP. The overhead of security key generation and maintenance was observed in case of PKI settings and Identity Based approach was proposed by author Huaqun Wang in Identity-based PDP[6] model. As previous models fails in multi cloud storages, the Cooperative PDP [2] was developed where the multi cloud storage was allowed. Then to eliminate overhead of PKI settings, ID-based multi cloud PDP [7] was proposed by Wang H. This scheme allows faster authentication process and multi cloud storage.

III. Various PDP Models

A. *Cooperative Provable Data Possession*

The parallel computing can be implemented in several ways of computing like instruction level, task and data parallelism. The data verification techniques like PDP perform slower in case of large volume of data. In such situations, the data integrity verification can be done in parallel and data storages can be on multiple clouds. The YanZhu, et al. [2] proposed Cooperative PDP model, which is based on zero knowledge proof mechanism and interactive proof system to prove the integrity of data stored in a multi cloud. A CPDP is a collection of two main algorithms (Key Gen, Tag Gen) and interactive proof system Proof [2].

Key Gen: It takes a security parameter as input and returns a secret key.

Tag Gen: It takes a secret key, file and set of cloud storage providers as input and returns triplet.

GenProof: A protocol to generate a proof of data possession among the CSP's and data verifier.

The CPDP approach allows parallel computing which enhances performance and also provides support for large file storage on cloud.

B. *Designated-Verifier Provable Data Possession in Public Cloud Storage*

In public clouds, it data integrity is a matter of crucial importance when the client cannot perform the remote data possession checking. The normal PDP approach increases overhead for clients where client needs to calculate tags and hash values for the data. The Yongjun Ren, et al. [3] proposed the designated data verification model for the clients with less recourses and computational power. The authors have proposed to use ECC (Elliptic Curve Cryptography)-based homomorphism authenticator to design PDP scheme, which does not compute expensive and time consuming bilinear and consume small amount of calculation and communications. This scheme is best suited for mobile clouds.

In terms of complexities, compared to RSA, elliptic curves cryptography (ECC)[10] provides shorter key length based on the same level of security. It has been shown by authors that 160-bit ECC provides comparable security to 1024-bit RSA. The communication overhead caused mostly comes from the DV-PDP response.

C. *Dynamic Provable Data Possession (DPDP)*

The Data dynamics plays important role in data integrity checking techniques. The PDP provides best suited scheme for only static outsourced data files, where as in general, client may wish to alter the outsourced data occasionally. To address and solve this issue C. Chris Erway, et al. provided Dynamic Provable Data Possession (DPDP) to allow data dynamics in outsourced data. They present a framework and efficient structures for DPDP approach, which extends the PDP model to support provable updates to stored data by introducing new version of authenticated dictionaries based on rank information.

They provided two approaches of DPDP [4], where a rank-based authenticated dictionary was built over a skip list. This construction provides a DPDP scheme with log computation and communication and the same detection probability as the original PDP scheme; and other is an alternative construction of a rank-based authenticated dictionary using an RSA tree [4]. This construction results in a DPDP scheme with improved detection probability but increases server computation.

D. *Robust DPDP*

A robust DPDP scheme implements mechanisms to mitigate arbitrary amounts of data corruption. The protection against small corruptions (i.e., bytes or even bits) ensures that attacks that modify a few bits do not destroy an encrypted file or invalidate authentication information. As updating a small portion of the file may require retrieving the entire file, the PDP scheme must be robust enough to perform dynamic updates.

The author Bo Chen, et al. [5] proposed two approaches towards Robust DPDP, the first construction provides efficient encoding, but causes high communication cost for updates. The second construction overcomes this drawback through a combination of techniques that consists RS codes based on Cauchy matrices, separating the encoding for robustness from the symbol position in the file, and reducing add/remove operations to append/modify operations when updating the RS-encoded parity data. Robustness is a vital property for all PDP schemes that rely on spot checking, which includes the majority of static and dynamic PDP protocols.

E. *Identity based Remote Data Possession Checking*

The existing PDP protocols have been designed in the PKI (public key Infrastructure) setting. In PDP approach, the cloud server has to authenticate the users' certificates before storing the data uploaded by the users in order to prevent spam. This incurs considerable costs as many users may frequently upload data to the cloud server. The author Huaqun Wang addressed this problem with a new model of identity-based RDPC (ID-RDPC) protocols [6]. They provided first ID based PDP protocol to be secure assuming the hardness of the standard computational Diffie-Hellman (CDH) problem. In addition to the structural advantage of elimination of certificate management [11] and verification, the ID-RDPC protocol also outperforms existing PDP protocols in the PKI setting in terms of computation and communication.

Firstly, the PKG (Private Key Generator) generates the system public key and the master secret key along with the private keys for the clients of an organization [6]. The main challenge to design the ID-RDPC protocol was that it requires the client to generate aggregatable ID-based signatures like tags for blocks without applying the hash-and-sign paradigm to the original data. The authors addressed this with a variation of the well-known Schnorr signature [11].

F. *Identity Based Distributed PDP*

In some scenarios, the clients have to store their data on multi-cloud servers to allow parallelism and huge data storage. So, the integrity checking protocol must be efficient to save the verifier's cost. The author Wang, H. proposed a novel PDP model as ID-DPDP (identity-based distributed provable data possession) in multi-cloud storage. Based on the bilinear pairing concept, the complete ID-DPDP protocol is designed [7]. The proposed ID-DPDP protocol is provably secure under the hardness assumption of the standard CDH (computational Diffie- Hellman) problem as tested by author. In addition to the structural advantage of elimination of managing certificate, the ID-DPDP approach is efficient and flexible. Depending on the client's authorization, the proposed ID-DPDP protocol can identify private verification and public verification.

IV. COMPARATIVE ANALYSIS OF PDP TECHNIQUES

The analysis of above PDP schemes will help to identify the best suited approach for given business context. The following table (Table1-I and II) provides comparative analysis of PDP schemes. The first table specifies variations of PDP algorithm along with the techniques used and whether it supports single or multi cloud storage. The second table we have compared the variations of PDP schemes by specifying advantages and disadvantages of them.

TABLE I COMPARATIVE ANALYSIS OF PDP SCHEMES- I

Sr. No.	Integrity checking Scheme	Algorithm/ Technique	Single/ Multi Cloud
1	PDP [1]	HVT, E-PDP	Single Cloud
2	CPDP (Cooperative PDP) [2]	Cooperative model for PDP which allows multi-cloud storage.	Multi Cloud
3	SPDP (Scalable PDP) [8]	PDP,MHT (Markle Hash Values), Scalable	Single Cloud
4	DV-PDP (Designated-Verifier PDP) [3]	Designated Verifier, elliptic curves cryptography (ECC)	Single Cloud
5	DPDP-I (Dynamic PDP) [4]	Authenticated Skip List	Single Cloud
6	RDPDP(Robust DPDP) [5]	RS (Reed-Solomon) codes based on Cauchy matrices	
7	ID-RDPC (Identity Based Dynamic PDP) [6]	Identity based cryptography, DPDP	Single Cloud
8	ID-Distributed PDP [7]	Bilinear-pairings, DPDP, Identity based technique.	Multi-Cloud

TABLE III COMPARATIVE ANALYSIS OF PDP SCHEMES-II

Sr. No.	PDP Scheme	Advantages	Disadvantages
1	PDP [1]	1. Protection against small corruptions. 2. Reduced update block communication 3. RSA scheme for security. 4. Allows public verifiability	1. Searching the block is poor (with brute force) 2. It is more efficient scheme but can applicable only for static files. 3. It is insecure against dynamic block of data.
2	CPDP [2]	1. Allows multi cloud storage. 2. Hash index hierarchy reduces search complexity.	1. Due to multi cloud storage, Combiner model needs to be added which may increase complexity.
3	SPDP [8]	1. It provides secure PDP by encryption 2. It is light weight PDP scheme as it supports homomorphic hash function.	1. The model fails in randomness. Hence by using the previous challenges, client can easily deceive the server.
4	DV-DPDP [3]	1. No client expertise is required. 2. Elliptic curves cryptography (ECC) has shorter key length based on the same level of security. 3. The total communication overhead is more efficient.	1. Extra setup is needed for designated verifier 2. Pairing based approach increases complexity.
5	DPDP [4]	1.Block modification and updating of block is allowed. 2.Efficient integrity verification is made by querying and updating DPDP scenario.	1. Client needs to perform extra computations. 2. It provides efficient verification but construction of rank based scheme is complex.
6	R-PDPD [5]	1. Lightweight as it reduces overheads and communication 2. Spot checking allows clients to randomly check data integrity.	1. Provides high communication overhead in first model of RDPDP
7	ID-DPDP [6]	1.Reduced Communication Overhead 2. It allows Data dynamics.	1. The approach increases 2. Can't adopt for multi-cloud storage.
8	ID-Distributed PDP [7]	1. Allows multi-cloud storage. 2. Provides more flexibility as a block is divided in multiple parts.	1. Incurred overhead due to combiner and PKG modules.

V. PROPOSED WORK

The above PDP schemes need client initiations for data integrity checking. Also, in case of company oriented environment, it will be useful if the system is tracking the data integrity verification transactions for future enhancements.

The model can be designed to achieve self initiated approach of PDP and log-based approach can be used for administration purpose. According to above PDP schemes, the proposed system will help client to maintain the data integrity verification records for further proceedings and also client will be freed from initiating the data integrity checking process.

In detail, we will design a system where there will be a timer which will keep generating interrupts and due to these interrupts the data integrity verification request will get generated on behalf of the client. The request then will be served by cloud server as normal PDP approach and will return the proof back to client. The figure (Fig.2) depicts an architecture of proposed system where the client is equipped with timer and has access to the log file generated by verifier module.

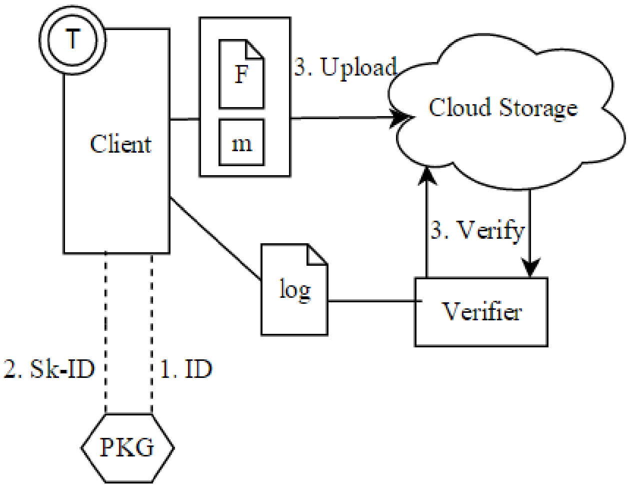


Fig.2 Proposed System Architecture

Our scheme will compare this proof to verify and the result will be saved on permanent storage as like a log file. The client can then periodically check the log file to analyse the request responses made and can assure data integrity.

VI. CONCLUSION

In cloud computing, the data integrity verification is crucial part. There are many PDP techniques which are available and further improved to achieve efficient integrity verification. We have identified latest PDP variations and compared those PDP schemes based on their approaches, techniques, advantages and disadvantages. As a result, we have proposed the enhanced Identity based PDP scheme for data integrity verification which will make client free from the data intactness checking and also will provide a scheme to perform administrative tasks.

REFERENCES

[1] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song., Provable data possession at untrusted stores. In CCS '07, pp.598-609, 2007. J. Breckling, Ed. *The Analysis of Directional Time Series: Applications to Wind Speed and Direction*, ser. Lecture Notes in Statistics. Berlin, Germany: Springer, 1989, vol. 61.

[2] Yan Zhu, Hongxin Hu,Gail-Joon Ahn and Mengyang Yu., Cooperative Provable Data Possession for Integrity Verification in Multicloud Storage. IEEE Transactions on Parallel and Distributed Systems, 23, 12(2012) M. Wegmuller, J. P. von der Weid, P. Oberson, and N. Gisin, "High resolution fiber distributed measurements with coherent OFDR," in *Proc. ECOC'00*.

[3] Yongjun Ren, Jiang Xu, Jin Wang, and Jeong-Uk Kim Designated-Verifier Provable Data Possession in Public Cloud Storage. In International Journal of Security and Its Applications Vol.7, No.6 (2013), pp.11-20.

[4] Erway.C.C, Kupcu.A, Papamanthou.C, and Tamassia.R(2009), Dynamic provable data possession, in ACM Conference on Computer and Communications Security, E. Al-Shaer, S. Jha, and A. D. Keromytis, Eds. ACM, pp. 213– 222.

[5] Bo Chen Reza Curtmola, Robust Dynamic Provable Data Possession. In IEEE Transactions on Distributed Computing Systems Workshop, 2012 32nd International Conference. Pp. 515-525.

[6] Huaqun Wang, Qianhong Wu, Bo Qin, and Josep Domingo-Ferrer, Identity-Based Remote Data Possession Checking in Public Clouds. In Information Security, IET (Volume:8 , Issue: 2), p.p.: 114-121.

[7] Wang, H., Identity-Based Distributed Provable Data Possession in Multi-Cloud Storage. At Services Computing, IEEE Transactions on (Volume:PP , Issue: 99).

[8] 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.

[9] T S Khatri, Prof G B Jethava, Survey on data Integrity Approaches used in the Cloud Computing. In International Journal of Engineering Research & Technology (IJERT) Vol. 1 Issue 9, November - 2012 ISSN: 2278-0181.

[10] A. Miyaji, M. Nakabayashi, S. Takano. New Explicit Conditions of Elliptic Curve Traces for FR-reduction. IEICE Transactions Fundamentals, 5:1234-1243, 2001.

[11] D. Boneh, M. Franklin. Identity-based Encryption from the Weil Pairing. CRYPTO 2001, LNCS 2139, 213-229, 2001.