ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Demonstrable Replication Vibrant Statistics Detention in Cloud Servers

Rudrashi Satish
Assistant Professor,
Priyadarshini Institute Of Technology,
Tirupati, AP, India.

Roopa R
Assistant Professor,
Priyadarshini Institute Of Technology,
Tirupati, AP, India.

Abstract:
Cloud Computing (CC) is a rising computing epitome that can potentially offer a number of essential advantages. One of the fundamental advantages of CC is pay-as-you-go pricing model. Where consumers pay only according to their usage of the services at present data creation is outpacing users' storage accessibility, thus there is an increasing need to outsource such huge amount of data are opting for outsourcing data to remote cloud service providers (CSPs). The data reproduction provided by the CSP to assurance the availability and permanence of their data. Web services crossing point that can be used to hoard and recover an unlimited amount of data with fees metered in GB/month. This important data should be simulated on several servers across multiple data centers reproducible data are stored at reduced levels of redundancy. A map-based demonstrable replication dynamic data possession (MB-DRDDP) scheme that has the following features i) it provides indication to the consumers that the CSP is not devious by storing fewer copies; ii) it supports outsourcing of vibrant data and its supports block-level operations, such as block modification, insertion, deletion, and append and iii) it allows endorsed users to impeccably access the file copies stored by the CSP give a proportional analysis of the proposed MB-DRDDP scheme with a reference model obtained by extending existing demonstrable replication of active single-copy schemes. Efficient Multi-Copy Demonstrable Data Possession (EMC-DDP) through extensive performance analysis and experimental results, we demonstrate the efficiency of our protocols.

Keywords: ***Cloud Computing, data replication, outsourcing data storage, data integrity, cryptographic protocols.***

1.INTRODUCTION

Cloud Computing (CC) is an emerging computing paradigm that can be viewed as a virtualized collection of computing resources, where customers are provisioned and de-provisioned recourses as they need. CC represents the vision of providing computing services as public utilities like water and electricity. CC services can be categorized into software as a service (SaaS). Platform as a service (PaaS) and Infrastructure as a service (IaaS). The extensively used model of CC services is the SaaS model in which the customers have access to the applications running on the cloud provider's infrastructure. Google Docs, Google Calendar are publicly known examples of this model.

The substantial interest of Cloud Computing archetype is due to a number of key advantages which make it a challenging research area in both academia and industry. This paradigm of Information Technology (IT) architecture supplies cost effective means of computing over a shared pool of resources, where users can avoid capital expenditure on hardware, software, and services as they pay only for CC model provides low management overhead and immediate access to abroad range of applications, maintenance

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

cost is reduced as a third part is responsible for everything from running the cloud to storing data. It is not only the economic benefits that customers can gain from CC model, but also flexibility to scale up and down IT capacity over time to business needs. The fact that data owners are no longer physically possess their sensitive data raises new redoubtable and demanding tasks related to data security and integrity protection in Cloud Computing.

Data security can be achieved by encrypting sensitive data before outsourcing to remote servers. It is a critical demand of customers to have strong indication that the cloud servers still possess their data and it is not being tempered with or partially deleted over time, especially because the internal operation details of the CSP may not be known by cloud customers.

The habitual 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. 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. Therefore, clients need efficient techniques to verify the integrity of their outsourced data with minimum computation, communication, and storage overhead. Consequently, many researchers have focused on the problem of Demonstrable Data Possession (DDP) and proposed different schemes to audit the data stored on remote servers. When verify multiple data copies, the overall system integrity check fails if there is one or more corrupted copies. To address this issue and recognize which copies have been corrupted, we discuss a slight modification to be applied to the proposed scheme.

2.PROBLEM DESCRIPTION, INSPIRATION& CENTRAL ASSISTANCE

Users resort to data reproduction to ensure the accessibility and durability of their sensitive data, especially if it cannot easily be reproduced. When we are writing a research paper, we are very careful to keep multiple copies of our paper to be able to recover it in case of any failure or physical damage. In the Cloud Computing paradigm, customers rely on the CSP to undertake the data replication task relieving the burden of local data storage and maintenance, but they have to pay for their usage of the CSP's storage infrastructure. On the other side, cloud customers should be securely and efficiently convinced that the CSP actually possesses all data copies our contributions can be summarized as follows a map-based demonstrable multi-copy dynamic data possession (MB-DMDDP) scheme. This scheme provides an adequate guarantee that the CSP stores all copies that are agreed upon in the service contract. The scheme supports outsourcing of dynamic data, i.e., it supports block-level operations such as block modification, insertion, deletion, and append. The authorized users, who have the right to access the owner's file, can seamlessly access the copies received from the CSP. Give a thorough comparison of MB-DMDDP with a reference scheme, which one can obtain by extending existing PDP models for implementation and experiments using Amazon cloud platform.

2.1 Inspiration and Challenges

The mechanism used for data reproduction vary according to the nature of the data copy's are needed for critical data that cannot easily be reproduced. The pricing model of the CSPs is related to the replication strategy. For example, Amazon S3 standard storage strategy maintains copies of customers' data on multiple servers across multiple data centers, while with Amazon Reduced Redundancy Storage (RRS) strategy — which enables customers to reduce their costs — noncritical, reproducible data is stored at reduced level of redundancy. As a consequence, the pricing for the Amazon S3 standard storage is approximately 50% higher than that of the RRS. Cloud servers can collude to cheat the customers by showing that they are storing all copies, while in reality they are storing a single copy. Therefore, cloud customers need secure and efficient techniques to ensure that the CSP is actually keeping all data copies that are agreed upon,

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

these copies are not corrupted, and thus they pay for real services.

2.2 System Model

In multi-owner and multi-user cloud computing model, four entities are involved, as illustrated in they are data owners, the cloud administration server, and data users. Data owners have a collection of files F. To enable efficient search operations on these files this will be encrypted, data owners first build a secure searchable index I on the keyword set W extracted from F, and then they submit I to the administration server. Finally, data owners encrypt their files F and outsource the corresponding encrypted files C to the cloud server. Upon receiving I, the administration server re-encrypts I for the authenticated data owners and outsources the re-encrypted index to the cloud server. Once a data user wants to search t keywords over these encrypted files stored on the cloud server, he first computes the corresponding trapdoors and submits them to the administration server. Once the data user is authenticated by the administration server, the administration server will further re-encrypt the trapdoors and submit them to the cloud server. Upon receiving the trapdoor T, the cloud server searches the encrypted index I of each data owner and returns the corresponding set of encrypted files. To improve the file retrieval accuracy and save communication cost, a data user would tell the cloud server a parameter k and cloud server would return the top-k relevant files to the data user. Once the data user receives the top-k encrypted files from the cloud server, he will decrypt these returned files. Instead, we treat the cloud server as 'curious but honest' which is the same as in previous works.

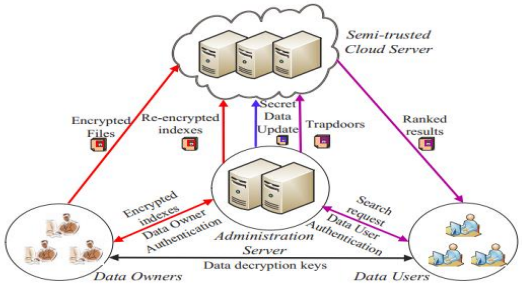


Fig1: Architecture of privacy preserving keyword search in a multi-owner and multi-user cloud model

1.3 Design Goals and Security Definitions

To enable privacy preserving ranked multi-keyword search in the multi-owner and multi-user cloud environment, our system design should simultaneously satisfy security and performance goals.

• **Ranked Multi-keyword Search over Multiowner**: The proposed scheme should allow multi-keyword search over encrypted files which would be encrypted with different keys for different data owners. It also needs to allow the cloud server to rank the search results among different data owners and return the top-k results.

• **Data owner scalability**: The proposed scheme should allow new data owners to enter this system without affecting other data owners or data users; the scheme should support data owner scalability in a plug-and-play model.

• **Data user revocation**: The proposed scheme should ensure that only authenticated data users can perform correct searches. Moreover, once a data user is revoked, he can no longer perform correct searches over the encrypted cloud data.

• **Security Goals:** Since the adversary A can know whether an encrypted keyword matches a trapdoor, we use the weaker security goal, that is, we should ensure that the probability for the adversary A to infer the actual value of a keyword is negligibly more than randomly guessing.

2.4 System Components:

The cloud computing storage model considered in this work consists of three main components as illustrated in a data owner that can be an organization originally possessing sensitive data to be stored in the cloud CSP who manages cloud servers (CSs) and provides paid storage space on its infrastructure to store the owner's files authorized users — a set of owner's clients who

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

have the right to access the remote data. The storage model used in this work can be adopted by many practical applications. For example, e-Health applications can be envisioned by this model where the patients' database that contains large and sensitive information can be stored on the cloud servers. In these types of applications, the e-Health organization can be considered as the data owner, and the physicians as the authorized users who have the right to access the patients' medical history. Many other practical applications like financial, scientific, and educational applications can be viewed in similar settings.

2.5 Threat Model

The integrity of customers' data in the cloud may be at risk due to the following reasons. First, the CSP — whose goal is likely to make a profit and maintain a reputation — has an incentive to hide data loss (due to hardware failure, management errors, various attacks) or reclaim storage by discarding data that has not been or is rarely accessed. Second, a dishonest CSP may store fewer copies than what has been agreed upon in the service contact with the data owner, and try to convince the owner that all copies are correctly stored intact. Third, to save the computational resources, the CSP may totally ignore the data-update requests issued by the owner, or not execute them on all copies leading to inconsistency between the file copies. The goal of the proposed scheme is to detect (with high probability) the CSP misbehavior by validating the number and integrity of file copies.

The proposed scheme consists of seven polynomial time algorithms: KeyGen, Copy Gen the CSP runs the algorithms Exec Update and Prove, while a verifier runs the Verify algorithm. − (pk, sk) Key Gen (). This algorithm is run by the data owner to generate a public key pk and a private key sk. The private key sk is kept secret by the owner, while pk is publicly known.

2.Multi-Copy Demonstrable Data Possession (MC-DDP) schemes

CSP offers to store n copies of an owner's file on n different servers — to prevent simultaneous failure of all copies and to achieve the availability aspect in exchange for prespecified fees metered in GB/month. 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 negative 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.

Through extensive analysis, we will elaborate the various features and limitations of the MR-DDP model especially from the authorized users' side did not consider how the authorized users of the data owner can access the file copies from the cloud servers noting that the internal operations of the CSP are opaque. Moreover, we will demonstrate the efficiency of our protocols from the storage, computation, and communication aspects. We believe that the investigation of both BMC-PDP and MR-PDP models will lead us to our main schemes.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

3.1Basic Multi-Copy Demonstrable Data Possession (BMC-DDP) scheme

The cloud servers can conspire to convince the data owner that they store n copies of the file while indeed they store fewer than n copies and this is due to the fact that the n copies are identical. Thus, the BMC-PDP scheme tries to solve the problem by leaving the control of the file copying operation in the owner's hand to generate unique distinguishable/differentiable copies of the data file.

To this end, 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. Although the BMC-DDP scheme is a workable solution, it is impractical and has the following critical drawbacks:

• The computation and communication complexities of the verification task grow linearly with the number of copies. Essentially, the BMC-DDP scheme is equivalent to applying any PDP schemes to n different files.

• Key management is a severe problem with the BMC-DDP scheme. Since the data file is encrypted under n different keys, the data owner has to keep these keys secret from the CSP and — at the same time — to share these n keys with each authorized user.

Moreover, when the authorized user interacts with the CSP to retrieve the data file, it is not necessarily to receive the same copy each time. According to the load balancing mechanism used by the CSP to organize the work of the servers, the authorized user's request is directed to this ever with the lowest congestion. Consequently, each copy should contain some indicator about the key used in the encryption to enable the authorized users to properly decrypt the received copy.

3.2 Multiple-Replica Demonstrable Data Possession (MR-DDP) scheme

Proposed Multiple-Replica DDP (MR-DDP) scheme where a data owner can verify that several copies of a file are stored by a storage service provider. The MR-DDP scheme is an extension to the PDP models proposed by 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).

• Since the MR-PDP scheme is an extension to the PDP models proposed by inherits all the limitations of these models identified earlier in the literature survey section: long tags (1024 bits to achieve 80-bit security level), 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.

• A slightly modified version of the critical key management problem of the BMC-DDP scheme is another concern in the MR-DDP scheme. The authorized users have to know which copy has been specifically retrieved from the CSP to properly unmask it before decryption.

Due to the opaqueness nature of the internal operations of the CSP, — the server on which a specific copy is exactly stored is unknown to cloud customers — the MR-DDP scheme 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.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Setup

- $N = pq$ is the RSA modulus (p & q are prime numbers)
- g is a generator of QR_N (QR_N is the set of quadratic residues modulo N)
- Public key $pk = (N, g, e)$, secret key $sk = (d, v, x)$, $v, x \in_R \mathbb{Z}_N$, and $ed \equiv 1 \bmod (p-1)(q-1)$
- π is a pseudo-random permutation, f_x is a pseudo-random function keyed with the secret key x, and H is a hash function
- File $F = \{b_1, b_2, \ldots, b_m\}$
- E_K is an encryption algorithm under a key K

Data Owner

- Obtains an encrypted file $\tilde{F}$ by encrypting the original file F using the encryption key K $\tilde{F} = \{\tilde{b}_1, \tilde{b}_2, \ldots \tilde{b}_m\}$, where $\tilde{b}_j = E_K(b_j)$, $1 \le j \le m$.
- Uses the encrypted version of the file $\tilde{F}$ to create a set of tags $\{T_j\}_{1 \le j \le m}$ for all copies to be used in the verification process: $T_j = (H(v||j) \cdot g^{\tilde{b}_j})^d \bmod N$
- Generates n distinct replicas $\{\hat{F}_i\}_{1 \le i \le n}$, $\hat{F}_i = \{\hat{b}_{i,1}, \hat{b}_{i,2}, \ldots, \hat{b}_{i,m}\}$, using random masking as follows.

  for $i = 1$ to n do
    for $j = 1$ to m do
      1. Computes a random value $r_{i,j} = f_x(i||j)$
      2. Computes the replica's block $\hat{b}_{i,j} = \tilde{b}_j + r_{i,j}$ (added as large integers in $\mathbb{Z}$)
- Data owner sends the ***specific*** replica $\hat{F}_i$ to a ***specific*** server S_i, $1 \le i \le n$

Checking possession of replica $\hat{F}_z$

To check the possession of the replica $\hat{F}_z = \{\hat{b}_{z,1}, \hat{b}_{z,2}, \ldots \hat{b}_{z,m}\}$, the owner challenges the *specific* server S_z as follows:

Owner		Remote Server S_z
1. Picks a key k for the π function, c(# of blocks to be challenged), and $g_s = g^s \bmod N$ ($s \in_R \mathbb{Z}_N$)	$\xrightarrow{c, k, g_s}$	
		2. Computes the challenged block indices: $j_u = \pi_k(u)_{1 \le u \le c}$
		3. Computes $T = \prod_{u=1}^{c} T_{j_u} \bmod N$
		4. Computes $\rho = g_s^{\sum_{u=1}^{c} \hat{b}_{z,j_u}} \bmod N$
	$\xleftarrow{T, \rho}$	
5. Computes $j_u = \pi_k(u)_{1 \le u \le c}$		
6. Checks $(\frac{T^e}{\prod_{u=1}^{c} H(v\|j_u)} \cdot g^{r_{chal}})^s \stackrel{?}{=} \rho$, where $r_{chal} = \sum_{u=1}^{c} r_{z,j_u}$ is the sum of the random values used to obtain the blocks in the replica $\hat{F}_z$		

Fig4: The MR-PDP protocol by Carmela

The data owner has a file F and the CSP offers to store n copies, {F1, F2.. , Fn}, of the owner's file in exchange for pre-specified fees metered in GB/month. We provide two versions of our EMC-PDP scheme: Deterministic EMC-DDP (DEMC-DDP) and Probabilistic EMC-DDP (PEMC-DDP). In the DEMC-DDP version, the CSP has to access all the blocks of the data file, while in the PEMC-PDP we depend on spot checking 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.

3.3 Probabilistic EMC-DDP (PEMC-DDP) scheme

As we have previously demonstrated, the PEMC-PDP depends on spot checking by validating a random subset of the file blocks instead of validating all the blocks to achieve better performance in the computation overhead. 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 server's side. The PEMC-PDP scheme also consists of five algorithms: KeyGen, Copy Gen, Tag Gen, Proof, and Verify.

3.4 PEMC-PDP Construction

The procedures of our PEMC-PDP protocol execution are as follows:

• Key Generation. The same as in the DEMC-DDP.

• Distinct Copies Generation. The same as in the DEMC-PDP.

• Tag Generation. Since the PEMC-PDP checks only a random subset of the file blocks, the block index is needed to be embedded into the block tag to prevent the CSP from cheating by using blocks at different indices.

Setup

- $\hat{e} : G_1 \times G_2 \to G_T$ is a bilinear map, u and g are two generators for G_1 and G_2 respectively, $x \in \mathbb{Z}_p$ is a private key, and $y = g^x \in G_2$ is a public key.
- File $F = \{b_1, b_2, \ldots, b_m\}$, where each block $b_i \in \mathbb{Z}_p$

Data Owner

- Creates distinct file copies $\mathbb{F} = \{F_i\}_{1 \le i \le n}$, where $F_i = E_K(i||F)_{1 \le i \le n}$. Each copy F_i is an ordered collection of blocks $\{b_{ij}\}_{1 \le j \le m}$
- Calculates the block tags $\Phi = \{\sigma_{ij}\}_{\substack{1 \le i \le n \\ 1 \le j \le m}}$, where $\sigma_{ij} = (H(F_{id}).u^{b_{ij}})^x \in G_1$
- Sends $\{\mathbb{F}, \Phi, F_{id}\}$ to the CSP and deletes the copies and the tags from its local storage.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

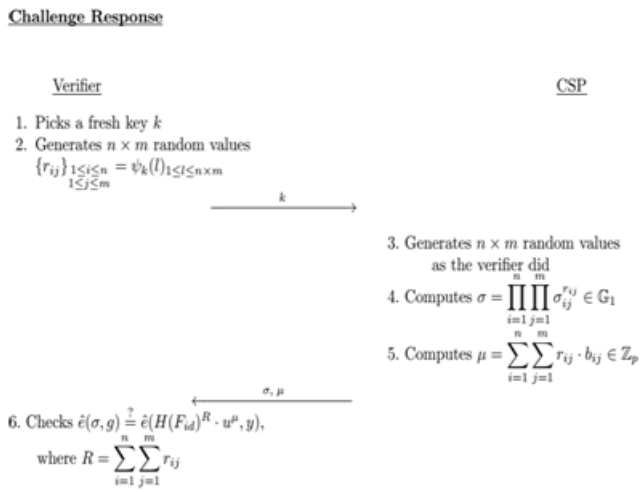


Fig5: The proposed DEMC-PDP scheme

4.PERFORMANCE ANALYSIS AND EXPERIM ENTAL RESULTS

4.1 Performance Analysis

In this section we evaluate the performance of our proposed EMC-DDP schemes and the MRDDP scheme proposed. The computation cost of our schemes and the scheme in is estimated in terms of the cryptographic operations notated loss of generality assumes the desired security level is 80-bit then the elliptic curve group we work on has a 160-bit group order and the size of the RSA modulus N is 1024 bits. Let the key used with the PRP and the PRF be of size 128 bits. Table 3 presents a comparison between our proposed schemes (DEMC-PDP and PEMC-PDP) and the MR-PDP scheme. The comparison is held from these perspectives: the storage and generation cost of the block tags, the communication cost for the challenge and response phase, and the computation cost at both the verifier and the CSP side to establish a fair comparison between our schemes and the MR-DDP we assume tiny modifications to the original protocol proposed. First, we assume that the indices of the challenged blocks are the same across all replicas for the CSP to prove the possession of the blocks, each one of the challenged blocks should be multiplied by a random value modifying the MR-PDP scheme to be an extension to the S-PDP version not the E-DDP version due to the second modification guarantees that the CSP is possessing each of the challenged blocks not just only their sum. Let n, m denote the number of copies and the number of blocks per copy, respectively, and c denotes the number of challenged blocks in both the PEMC-DDP and the MR-DDP.

		DEMC-PDP	PEMC-PDP	MR-PDP[15]
Tag	Size	160 nm bits	160 m bits	1024 m bits
	Generation	$2nm\mathcal{E}+nm\mathcal{M}+nm\mathcal{H}$	$2nm\mathcal{E}+2nm\mathcal{M}+nm\mathcal{H}$	$2m\mathcal{E}+m\mathcal{M}+m\mathcal{H}$
Communication Overhead	Challenge	128 bits	$256+\log_2(c)$ bits	$1280+\log_2(c)$ bits
	Response	320 bits	$160(n+1)$ bits	$1024(n+1)$ bits[5]
Computation Overhead	Proof	$nm\mathcal{E}+2nm\mathcal{M}+nm\mathcal{A}$	$c\mathcal{E}+c(n+1)\mathcal{M}+cn\mathcal{A}$	$(c+n)\mathcal{E}+c(n+1)\mathcal{M}+cn\mathcal{A}$
	Verify	$2\mathcal{P}+2\mathcal{E}+1\mathcal{M}$ $+nm\mathcal{A}+1\mathcal{H}$	$2\mathcal{P}+(c+2)\mathcal{E}+(c+1)\mathcal{M}$ $+n\mathcal{A}+c\mathcal{H}$	$2n+c+1)\mathcal{E}+(cn+c+n)\mathcal{M}$ $+cn\mathcal{A}+c\mathcal{H}+1\mathcal{D}$

Fig6: Storage, communication, and computation costs for the three schemes. The symbols used in the comparison are defined

By Completeness, we mean that our schemes handle all entities — data owners, CSPs, and authorized users — that comprise almost all practical applications, while the protocol in missed an essential entity in the outsourced data storage system which is the authorized users.

• The storage overhead (total tags size) of our PEMC-PDP is 6 times less than that of the MR-DDP. In general our tag size is 1/6 the tag size of the scheme due to and this mitigates the extra storage cost of our DEMC-PDP. For a 64-MB file with 4-KB blocks and 10 copies, the total tags sizes of the MR-PDP, DEMC-PDP, and PEMC-DDP are 2 MB, 3.125 MB, and 0.3125 MB, respectively.

• The communication overhead of our schemes is much less than that of the MR-DDP; the way we construct our schemes enables us to aggregate the responses from the servers on which the copies are stored. For the challenge phase, the communication cost of our DEMC-DDP is 10 times less than that of the MR-PDP, and it is 5 times less for our PEMC-PDP. Moreover or just only 5 copies we compress the

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

response of our DEMC-DDP to about 1/19 of the MR-DDP response, and the compression ratio between the response of our PEMC-DDP to that of the MR-PDP is about 1:6. Hence, our schemes are efficient and much more practical especially when the available bandwidth is limited.

4.2 Experimental Results

In this section, we present and discuss the experimental results of our research. The experiments are conducted using C++ on a system with an Intel(R) Xeon (R) 2-GHZ processor and 3 GB RAM running Windows XP. In our implementation we use MIRACL library version 5.4.2 and 64-MB file. To achieve 80-bit security level, the elliptic curve group we work on has a 160-bit group order and the size of the RSA modulus N is 1024 bits. In our implementation we do not consider the time to access the file blocks, as the state-of-the-art- hard drive technology allows as much as 1 MB to be read in just few nanoseconds. Hence, the total access time is unlikely to have substantial impact on the overall system performance. Our DEMC-PDP has the strongest guarantee that all blocks of all copies are actually being stored by the CSP and they are intact, but this strongest guarantee is at the expense of the storage overhead.

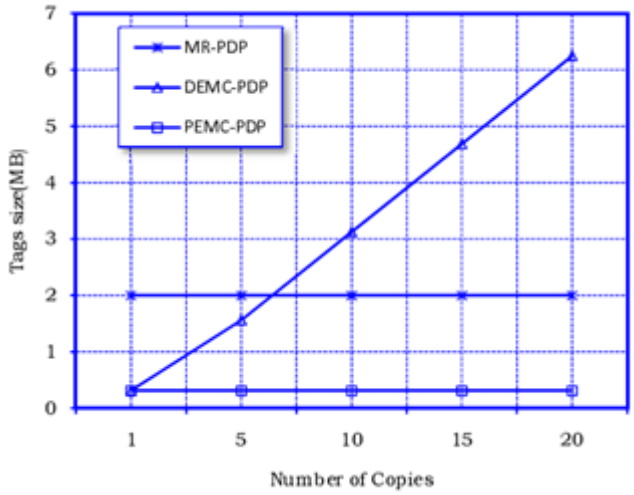


Fig7: Proposed schemes and the MR-PDP model

The tags generation time of the MR-DDP is the lowest one, and this is because it generates a single set of tags for all copies. But as mentioned earlier, this reduced computation of tags generation resulted in precluding the authorized users from seamlessly accessing the owner's files. Moreover, the tags generation time is unlikely to have significant impact on the overall system performance; tags generation task is done only once during the files life time which may be for tens of years. As noted from figure 12, the time of our PEMC-PDP scheme to generate tags is slightly higher than that of our DEMC-PDP model. This slight difference is due to the additional nmM operations that are used to aggregate the tags of the blocks at the same indices

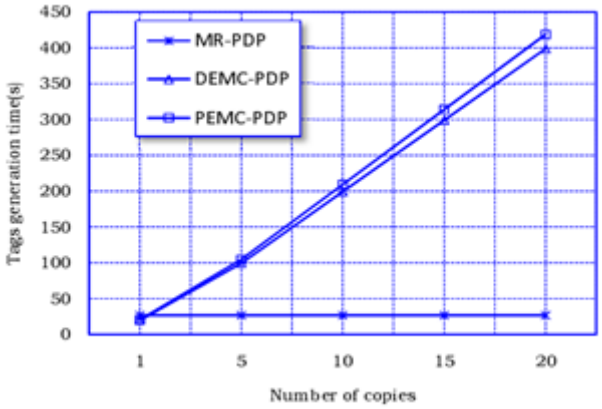


Fig8: Proposed schemes and the MR-DDP model.

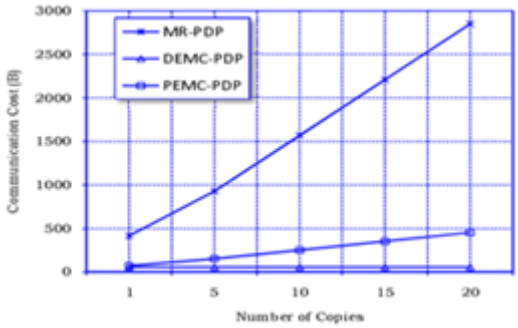


Fig9: proposed schemes and the MR-DDP model

# of Copies	DEMC-PDP	PEMC-PDP	MR-PDP [15]	*Speed-up*
1	10485.76	294.4	394.57	25.39%
5	52428.8	323.76	440.13	26.44%
10	104857.6	360.43	497.08	27.49%
15	157286.4	395.9	554.03	28.54%
20	209715.2	430.19	610.98	29.59%

CSP computation times (ms) of the three schemes for different number of copies

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

CONCLUSION

Outsourcing data to remote servers has become a growing trend for many organizations to alleviate the burden of local data storage and maintenance. In work we have studied the problem of creating multiple copies of dynamic data file and verifying those copies stored on unfrosted cloud servers. We have proposed a new DDP scheme referred to as MB-PMDDP), which supports outsourcing a multi-copy dynamic data, where the data owner is capable of not only archiving and accessing the data copies stored by the CSP, but also updating and scaling these copies on the remote servers. To the best of our knowledge, the proposed scheme is the first to address multiple copies of data. The interaction between the authorized users and the CSP is considered in our scheme, where he authorized users can seamlessly access a data copy received from the C. using a single secret key shared w. the data owner. Moreover, the proposed scheme supports public verifiability, enables arbitrary number of auditing, and allows verification where the verifier has the ability to verify the data integrity even though he neither nor thieves the file bloc from the server. Through performance analysis a. experimental results, we have demonstrated the proposed MB-PMDDP scheme outperforms the TB-PMDDP approach derived from a class of dynamic single-copy PDP models. The TB-PMDDP leads to high storage overhead on the remote servers and high computations on both the CSP and the verifier sides. The MB-PMDDP scheme significantly reduces the computation time during the challenge-response phase which makes it more practical for applications where a large number of verifiers are connected to the CSP causing a huge computation overhead on the servers. Besides, it has lower storage overhead on the CSP, and thus reduces the fees p. by the cloud customers. The dynamic block operations of the map-based approach are done with communication cost than that of die tree-based approach. A slight modification can be done on the proposed scheme to support the feature of identifying the indices of corrupted copies. Through security analysis, we have shown that the proposed scheme is provably secure.

Literature Survey

Demonstrable Data Possession (DDP)

Demonstrable data possession (DDP) is a methodology for validating the integrity of data in outsourcing storage service. The fundamental goal of the PDP scheme is to allow a verifier to efficiently, periodically, and securely validate that a remote server — which supposedly stores 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. First, the data owner computes a message authentication code (MAC) of the whole file before outsourcing to a remote server. Then, the owner keeps only the computed MAC on his local storage, sends the file to the remote server, and deletes the local copy of the file. 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. Alternatively, instead of computing and storing the MAC of the whole file, the data owner divides the file F into blocks $\{b1, b2, \ldots, bm\}$, computes a MAC σi for each block sends both the data file F and the MACs $\{\sigma i\}1 \leq i \leq m$ to the remote/cloud server, deletes the local copy of the file, and stores only the secret key sk. 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 [7]. The rationale behind the second approach is that checking part of the file is much easier than the whole of it. However both approaches suffer from severe drawback; the communication complexity is linear with the queried data size which is impractical especially when the available bandwidth is limited.

There are two main limitations in the protocol of Descartes

• In each verification, the remote server has to do the exponentiation over the entire file. Thus, if we are

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

dealing with huge files, e.g., in order of Terabytes (as most practical applications require) this exponentiation will be heavy.

• Storage overhead on the verifier side; it has to store some metadata for each file to be checked.

This could be a challenge for the verifier if it uses small devices, e.g., a PDA or a cell phone with limited storage capacity.

DDP Schemes: Proposed a scheme to verify data integrity using the RSA-based Homomorphism hash function. A function H is Homomorphism if, given two

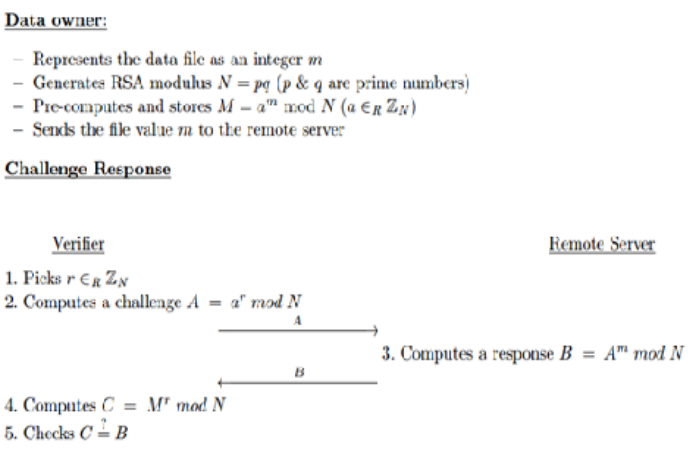
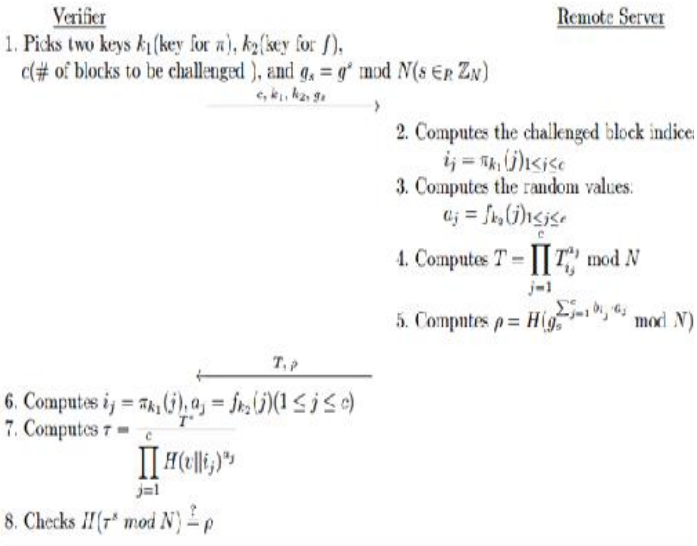


Fig2: DDP Protocol



II. E-PDP scheme

The only difference between the E-PDP and the S-PDP is that : $\{a_j\}_{1 \le j \le c} = 1$, and thus

- Step 4 : $T = \prod_{j=1}^{c} T_{i_j} \bmod N$
- Step 5 : $\rho = H(g_s^{\sum_{j=1}^{c} b_{i_j}} \bmod N)$
- Step 7 : $\tau = \frac{T^s}{\prod_{j=1}^{c} H(v \| i_j)}$

Fig3: The S-PDP and E-PDP protocols by Attendees.

Proof of Irretrievability (POR)

A Proof of Irretrievability (POR) scheme is an orthogonal/a complementary approach to a Provable Data Possession (PDP) system. A POR scheme is a challenge-response protocol which enables a remote server to provide evidence that a verifier can retrieve or reconstruct the entire data file from the responses that are reliably transmitted from the server. The main idea of the POR schemes is to apply erasure code to data files before outsourcing to allow more error resiliency. Thus, if it is a critical demand to detect any modification or deletion of tiny parts of the data file, then erasure code could be used before outsourcing. The work done by Juels and Kaliski was from the first papers to consider formal models for POR schemes. In their model, the data is first encrypted then disguised blocks (called sentinels) are embedded into the cipher text. The sentinels are hidden among the regular file blocks in order to detect data modification by the server. In the auditing phase, the verifier requests for randomly picked sentinels and checks whether they are corrupted or not. If the server corrupts or deletes parts of the data, then sentinels would also be influenced with a certain probability. The main limitation of the scheme in [23] is that it allows only a limited number of challenges on the Comparison of PDP schemes for a file consisting of m blocks, c is the number of challenged blocks, and N e is a finite number of random challenges.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Scheme	[8]	[9]	[10]	[11]	[12, 13]	[6]
Owner pre-computation	EXF	$O(1)$	$O(m)$	$O(m)$	$O(1)$	$O(m)$
Verifier storage overhead	$O(1)$	$O(1)$	$O(m)$	$O(1)$	$O(\tilde{N})$	-
Server storage overhead	-	-	-	-	-	$O(m)$
Server computation	EXF	EXF	$O(c)$	$O(m)$	$O(1)$	$O(c)$
Verifier computation	$O(1)$	$O(1)$	$O(c)$	$O(1)$	$O(1)^{\dagger}$	$O(c)$
Communication cost	$O(1)$	$O(1)$	$O(1)$	$O(1)$	$O(1)$	$O(1)$
Unbounded challenges	✓	✓	✓	✓	×	✓
Fragmentation	×	×	✓	✓	×	✓
Type of guarantee	DET	DET	DET/ PRO	DET	DET	PRO‡
Prove data possession	✓	✓	✓	×	✓	✓

Data files, which is specified by the number of sentinels embedded into the data file. This limited number of challenges is due to the fact that sentinels and their position within the file must be revealed to the server at each challenge and the verifier cannot reuse the revealed sentinels. Schwartz and Miller proposed using algebraic signature to verify data integrity across multiple servers using error-correcting codes data files, which is specified by the number of sentinels embedded into the data file. This limited number of challenges is due to the fact that sentinels and their position within the file must be revealed to the server at each challenge and the verifier cannot reuse the revealed sentinels. Schwartz and Miller proposed using algebraic signature to verify data integrity across multiple servers using error-correcting codes.

CC deployment model in which an organization provides and handles some internal and external resources. For example, an organization can use a public cloud service as Amazon to perform the general computation, while the data files are stored within the organizations local data center in a private cloud

7. REFERENCES

1. G. Ateniese et al, "Provable data possession at untrusted stores," in Proc. 14d: ACM Conf. Comput. Commun. Secur. (CCS), New Y(rk, NY, USA, 2007, pp. 598-609.
2. K. ang, "Publicly verifiable remote data integrky," in P. 10th Int. Conf Inf. Commun. Secur.(ICICS), 2008, pp.419-434.
3. Y. Deswarte, J.-J. Quisquater, and A. Saidane, "Remote integrity checking," in Prvc. 6th Working Conf. Thtegr. Internal Control Inf. Syst. (IICIS),2003, pp. I—II.
4. D. L G. Filho and P. S. L. M. Barreto, "Demonstrating data possession and uncheatab le data transfer," IACR (International Association for Cryptologic Research) ePrint Archive, Tech. Rep. 2006/150, 2006.
5. F. Sebe, J. Domingo-Ferrer, A. Martinez-Balleste, Y. Deswarte, and J.-J. Quisquater, "Efficient rem. data possession cheating in critical information inftustructures," IEEE Trans. Knowl. Data Eng., voL 20, no. 8, pp. 1034-1038, Aug. 2008.
6. P. Golle, S. Jarecici, and L Mironov, "Cryptographic primitives enforcing communication and storage complexity," in Proc. 6A Int. Conf. Financial Cryptograph. (FC),Be., Germany, 2003, pp. 120-135.
7. M. A. Shah, M. Baker, J. C. Mogul, and R. Swaminathan, "Auditing to keep online storage services honest," in P. 1 1 th USENIX Workshop Hot Topics 09:er. Syst. (HOTOS), Berkeley, CA, USA, 2007, pp. 1-6.
8. M. A. Shah, R. Swaminathan, and M. Baker, "Privacy-preserving audit and extraction of digital contents," Cryptology ePrint Archive, Report 2008/186, 2008.
9. E. Mykletun, M. Narasimha, and G. Tsudik, "Authentication and integrity in outsourced databases," Trans. Storage, vol. 2, no. 2, 2006.
10. R. Curtmola, O. Khan, R. Burns, and G. Ateniese, "MR-PDP: Multiple-Replica Provable Data Possession," in 28th IEEE ICDCS, 2008, pp. 411–420.
11. R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," Commun. ACM, vol. 26, no. 1, 1983.
12. D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," in ASIACRYPT 'Proceedings of the 7th International Conference on the Theory and

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Application of Cryptology and Information Security, London, UK, 2001, pp. 514–532.

13. F. Li, M. Hadjieleftheriou, G. Kollios, and L. Reyzin, "Dynamic authenticated index structures for outsourced databases," in SIGMOD '06: Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data, New York, NY, USA, 2006, pp. 121–132.
14. H. Shacham and B. Waters, "Compact proofs of retrievability," Cryptology ePrint Archive, Report 2008/073, 2008, http://eprint.iacr.org/.
15. G. Ateniese, S. Kamara, and J. Katz, "Proofs of storage from homomorphic identification protocols," in ASIACRYPT '09: Proceedings of the 15th International Conference on the Theory and Application of Cryptology and Information Security, Berlin, Heidelberg, 2009, pp. 319–333.