A Multi-Function Provable Data Possession Scheme in Cloud Computing

Xiaojun Yu, Qiaoyan Wen

State key laboratory of networking and switching technology,
Beijing University of Posts and Telecommunications, *Beijing*, *100876*, *China*
{yuxiaojun@bupt.edu.cn, wqy@bupt.edu.cn}

Abstract: In order to satisfy the different requirements of provable data possession in cloud computing, a multi-function provable data possession (MF-PDP) is proposed, which supports public verification, data dynamic, unlimited times verification, sampling verification. Besides, it is security in RO model and it is verification privacy under half trust model and can prevent from replacing attack and replay attack. The detail design is provided and the theory analysis about the correct, security and performance are also described. The experiment emulation and compare analysis suggest the feasibility and advantage.
Keywords: multi-function; provable data possession; cloud computing;

1 Introduction

With the cost and technology advantage, the cloud computing is considered as the best solution for information development requirement. However, there are many security problems in cloud computing. One of the basic security problems is data security and privacy.. Thus, many security technologies were used, such as Searchable encryption [1](SE) for data search with confidentiality , proof of deletion [2](POD) for data clear up in shared storage, provable of data possession (PDP)[3-15] for data integrity audit.

This paper focus on the PDP, which is used to check data integrity without downloading the real data from the server. The PDP can also be used as security service to improve the trust level of cloud service provider. However, this technology is faced on many challenges in some cloud, where (a)the data is dynamically updated (b) the client is performance limited.(c) the cloud service provider is untrusted and may take attacks to avoid responsibility if the real data was broken. Thus, a good PDP in such environment should meets both low computing complex and high security. Though there are many work of PDP, Most of which have limitation in performance or security. According to the current works, especially work [3], this paper proposed an improved version of PDP, named MF-PDP, which is better than most of works in function, performance and security.

The paper structure is as followings: chapter 2 is related work; chapter 3 is preliminaries; chapter 4 is the Design detail; chapter 5 is the analysis detail; chapter 6 is the evaluation; the last chapter is conclusion.

This work is supported by NSFC (Grant Nos. 61300181, 61272057, 61202434, 61170270, 61100203, 61121061), the Fundamental Research Funds for the Central Universities (Grant No. 2012RC0612, 2011YB01)

The main contributions of this paper are summarized as followings:

(1) We proposed how to design the PDP that can simultaneously support multi-function requirements in cloud computing environment, including public verification, data dynamic, unlimited times verification; sampling verification, verification privacy.

(2) We give a theory analysis in security and performance. Under assumptions of KEA1-r and RSA, the proposed PDP scheme is security in Random Oracle model. The Complexity of communication is $O(n)$, Storage complexity are $O(1)$, $O(n)$ and $O(1)$ for the data owner, the cloud service provider and the third party auditor respectively. Computing complexity are $O(n)$, $O(c)$, $O(c)$ for the data owner, the cloud service provider and the third party auditor respectively.

(3) We give an evaluation of the proposed scheme. The experiment evaluation show the consistency with theory analysis and high efficiency in performance .The comparison with other works in function shows the advantage.

2 Related work

Ateniese[3]firstly proposed the provable data possession model named PDP and two schemes named SPDP and EPDP based on RSA homomorphic verifible tag were designed. The EPDP is the specific of the SPDP and both support sampling verification and unlimited times verification. The SPDP can't support the public verification and data dynamic. The EPDP support public verification but not verification privacy. In the following work, Ateniese proposed an efficient verification scheme based on Hash function [4]. Though it supports sampling verification, it doesn't support public verification, unlimited times verification.

Sebe[5]designed a Proof of Retrievability based on homomorphic linear authentic which not only can be used to verify the data integrity but also to retrieve the data if some error has occurred. However, this scheme can't support public verification and have data leakage problem in verification progress.

Erway[6]proposed the DPDP model for supporting the data dynamic. He also designed two DPDP scheme, named DPDP-I and DPDP-II, which can support data dynamic, sampling verification, unlimited verification. However, the DPDP-I have replay attack issue and the DPDP-II cannot support verification privacy. Chen[15] add the robustness to DPDP through a combination of technologies such as RS codes based on Cauchy matrices, but the performance need further improved.

Wang[7] designed a integrity verification scheme with public verification and data dynamic, which uses the merkle hash tree to build the authentication structure for data dynamic requirement. Wang[8]proposed the distributed integrity verification scheme base RS code and homomorphic tokens, which support sampling verification and unlimited times verification, but it can't support public verification and data dynamic. Wang[9]proposed a integrity verification scheme with verification privacy and public verification. However, XU[10]indicates that there are security problem, such as Known Plain text attack. Wang[11]proposed a integrity authentication scheme based on HLA and bilinear pair, which has provable data possession property, sample

verification, unlimited times verification, publicly verification and verification privacy, but cannot support data dynamic.

Zhuo[13]designed a PDP scheme based on homomorphic linear code, which support data dynamic, unlimited verification, publicly verify and privately verify, but can't support sampling verification that means all data must be used in each verification process. Furthermore, this scheme can't prevent from replacing attack that the prover uses other data and data verification tag to build the integrity proof of the challenged data.

3 Preliminaries

Definition 1. (KEA1-r)Knowledge of Exponent Assumption[16]

For any adversary A that take input (N, g, g^s) and returns group elements (C, Y), such that $Y = C^s$,there exists an extractor ε which, given the same inputs as A ,returns x ,such that $C = g^x$.

Definition 2. Privacy against third party verifier[17].

We say provable data possession scheme $\prod$ is privately computes f_v, if there exist a probabilistic polynomial-time algorithm emulator S_v , such that $\{S_v(x, f_v(x, y))\}_{x,y\in\{0,1\}^*} \overset{C}{\equiv} \{view_V^{\prod}(x, y)\}_{x,y\in\{0,1\}^*}$ Where f_v is the verification function and a deterministic functionality. $view_V^{\prod}(x, y)$ is *the* view of verifier *during an execution of* $\prod$ *on* (x, y) . $view_V^{\prod}(x, y) = (x, r, m_1, ...m_t)$, *where* r *represents the outcome of the internal coin tosses, and* m_i *represents the* i *-th message it has received*. $\overset{C}{\equiv}$ denotes computational indistinguishability by (non-uniform) families of polynomial-size circuits.

Definition 3. RSA problem

For $y \in Z_N^*$, find the root x ,such that $x^e = y \bmod n$. Where $N = p * q$ be a multiplication of two big primes, $\Phi(N)$ is Euler function of N, $e \in Z_N^*$ is an integrity number and be prime to $\Phi(N)$.

Definition 4. RSA assumption

For the product of any two big prime $n = p \times q$, the RSA problem is hard to solve or may be the same hard as decomposing n.

4 MF-PDP Design

4.1 Assumption

4.1.1 System model

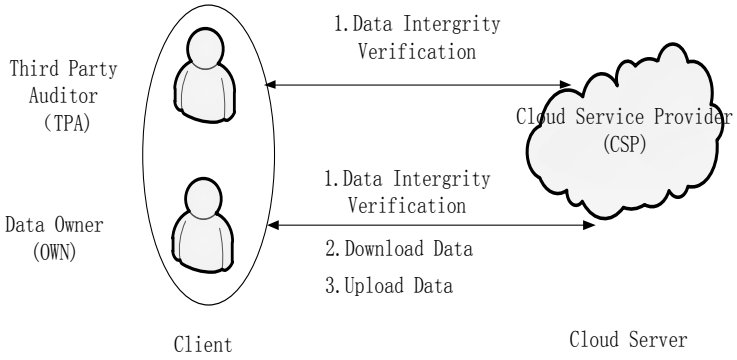

Figure.1 system model

Figure.1 is the system model, including two parts: the Client and the Cloud Server. And the Client can be two types that are data Owner (OWN) and Third Party Auditor (TPA).

The cloud server is managed by cloud service provider (CSP), which provides data storage service and data integrity verification service. The OWN is the customer of cloud storage service and participate in the data integrity verification. The TPA is the independent part who is represent the OWN to take data integrity audit task.

4.1.2 Threat model

We assume that the CSP may replace proof or replay proof in order to pass the verification. The TPA also may record related information in order to discover the user's data. The TPA is independent of the CSP, thus they will not take collusion attack. The client environment is safe and the OWN is trust for the data is belong to him.

4.2 Notations

For convenient, some notations are given in Table 1.

Table. 1: notations description

F	Data File $F=\{m_1,m_2,\ldots m_n\}$	*chal*	Challenge
m_i	data block.	$\|x\|$	Length operation
n	Data block number of each file.	$x\|\|y$	Link operation
l	The length of data block.	$f_k(x)$	pseudo-random function
c	Number of challenged data block	$\pi_k(x)$	pseudo-random permutation
k	Random number key	$H(x)$	hash function
pk, sk	Public key, Secret key	V	Proof
T_m	the tag of data block m	R	Random sequence $R=\{r_1,r_2\ldots r_n\}$
Σ	The set of metadata of data blocks	H_R	Digital signature of R

4.3 Design Idea

The MF-PDP is an improved version of PDP[3],but It supports some new requirements in cloud environment by using new ideas. See Table 2 .

In function requirement: it supports data dynamic by remove the data block index when computing the data block tag and public verification by using public key in verification stage.

In performance requirement, it uses the homomorphic verifiable tag[3] to reduce the communication cost and the random sampling verification to reduce computing cost.

In security requirement, the verification procedure is dependent on two parts proof information that one part comes from the prover and the other comes from the verifier, this idea not only make sure the valid of proof but also resisting the proof replacing attack.

Table 2: Design idea

function	Design idea
Data Dynamic	The data block tag rule is $T_{m_x} = (g^{m_x} \times r_x)^d \bmod N$, which is only depend on the data block content, thus the update operation on x-th data block has no effect on other data block tag.
Sampling verification	in each verification, the challenged data blocks are chosen based on the random challenge $chal = (k_1, k_2, g_s, c)$, we use AES as the random method to ensure the result valid with high probability and low computing cost.
Public verification	the data integrity proof is verified with public key, that is $CheckProof(pk, chal, V)$,thus anyone who get the pk can verify the proof.
Verification privacy	we use SHA-1 as one-way hash function to compute the data integrity proof. In RO model, it ensures the TPA can't find any information of the challenged data except the output of hash function.
Unlimited times verification	in each verification, the challenge $chal = (k_1, k_2, g_s, c)$ and proof are created randomly which make the each challenge and proof independently and security for statistical attack.
Anti-replacing attack	the data integrity proof is computed based on two parts information ,one part is stored on the prover, such as data block tag, the other part is provided by verifier, such as random numbers in challenge. The verifier will check the proof with random numbers in challenge. Thus, if the prover uses other data block tag or data content, the verify result will be failure.
Anti-replay attack	the proof is computed based on the challenged data block tag and random number. Thus, the two proofs for the same data content are different.

4.4 MF-PDP Scheme

Based on the definition model of Provable Data Possession Scheme[3], we propose our MF-PDP definition.

Definition 5 : The Multi-Function Provable Data Possession Scheme, MF-PDP, is composed of five probability algorithms, MF-PDP={ $KeyGen, TagBlock, GenChal$, $GenProof, CheckProof$, $Update$ },where:

- $KeyGen(1^k) \to (pk, sk)$: The OWN computes $pk = (e, g, N)$, $sk = (d, N)$, $N = pq$, $\Phi(N) = (p-1)(q-1)$, $p = 2p' + 1$, $q = 2q' + 1$, p , q , p' , q' are primes. e is a primes and d is the inverse of e , $ed = 1 \bmod \Phi(N)$, g is the generator of QR_N ,which is multiplication cycle group composed of the quadratic residues modulo N .The pk is published in from of certificate, sk is kept by the OWN.
- $TagBlock(pk, sk, m) \to T_m$:The OWN computes tag of each data block $T_{m_i} = (g^{m_i} \times r_i)^d \bmod N$,in which, $pk = (e, g, N)$, $sk = (d, N)$, $1 \le i \le n$, $r_i \in Z_N$; R = $\{r_1, r_2, .r_n\}$ is the random sequence, H_R = $(H(r_1 \| r_{2|} \mid r_3 ... \| r_n))^d \bmod N$ is the digital signature of random sequence. The OWN sends M= (F , Σ ={ T_{m_i} }, R , H_R) to CSP.
- $GenChal(pk) \to chal$: The OWN (or TPA) randomly chooses $(k_1, k_2, s) \in Z_N$, $c \in Z_n$ and computes $g_s = g^s \bmod N$, then the $chal = (k_1, k_2, g_s, c)$ is sent to the CSP.
- $GenProof(pk, F, chal, \Sigma) \to V$:The CSP computes $P_h = H(g_s^{a_1 m_{i_1} + a_2 m_{i_2} + .. a_c m_{i_c}} \bmod N)$ $= H(g_s^{\sum_{j=1}^{c} a_j m_{i_j}} \bmod N)$, $P_t = \prod_{j=1}^{c} T_{m_{i_j}}{}^{a_j} = \prod_{j=1}^{c} (g^{m_{i_j}} \times r_{i_j})^{d * a_j} \bmod N$,where $i_j = \pi_{k_1}(j)$, $a_j = f_{k_2}(j)$, $pk = (e, g, N)$. The proof $V = (P_h, P_t, R, H_R)$. Then the CSP return V to OWN （or TPA）.
- $CheckProof(pk, chal, V) \to \{"success", "failure"\}$: The OWN （ or TPA ） check $H(r_1 \| r_2 \| r_3 ... \| r_n) \bmod N \stackrel{?}{=} (H_R)^e$,if not equal, meaning the random sequence is not integrity, output “failure”, else, compute $P_r = \prod_{j=1}^{c} r_{i_j}^{a_j} \bmod N$, C = $H((\frac{P_t^e}{P_r})^s)$,where,

$i_j = \pi_{k_1}(j)$, $a_j = f_{k_2}(j)$, $r_{i_j} \in R$,then check $C \stackrel{?}{=} P_h$,if not equal, then the verification failure, output "$failure$"; if equal, it means the CSP possess the OWN's data and output "$success$".

- $Update(m_x, i, optype, T, F, R, H_R) \rightarrow (T', F', R', H_{R'})$: where m_x is the new data block for insert or target data block for modification and deletion. i is the data block operation index. $optype$ = { "Insert", "Modify", "Delete" }. The OWN checks the valid of R with H_R, if failure, output warning and exit update process, or else according to the $optype$, the OWN run the functions with the CSP, the CSP updates (T, F, R, H_R) to $(T', F', R', H_{R'})$.

4.5 MF-PDP System

Based on the definition model of Provable Data Possession System [3], we propose MF-PDP System definition.

Definition 6: The Multi-Function Provable Data Possession System based on MF-PDP scheme consists of three stages, MF-PDP System={ Setup stage, Challenge stage ,update stage},where:

- Setup stage: The OWN runs $KeyGen(1^k) \rightarrow (pk, sk)$, and then execute $TagBlock(pk, sk, m_i) \rightarrow T_{m_i}$ for $1 \le i \le n$. The OWN publishes the pk and keep sk, and then the OWN sends message M= (F, Σ ={T_{m_i}}, R, H_R) to the CSP for storage and delete local file F and Σ.
- Challenge stage: The OWN（or TPA）runs $GenChal(pk) \rightarrow chal$ to generate a challenge $chal = (k_1, k_2, g_s, c)$,and sends $chal$ to the CSP. The CSP runs $GenProof(pk, F, chal, \Sigma) \rightarrow V$ to generate proof of possession $V = (P_h, P_t, R, H_R)$ and sends V to the OWN（or TPA）. Finally, The OWN （ or TPA ） checks the validity of proof V by running $CheckProof(pk, sk, chal, V) \rightarrow \{"success", "failure"\}$.
- Update stage: The OWN runs $Update(m_x, i, optype, T, F, R, H_R) \rightarrow (T', F', R', H_{R'})$, check the valid of R, and execute sub functions according to the optype value with the CSP. The result is updates (T, F, R, H_R) to $(T', F', R', H_{R'})$.

5 MF-PDP Analysis

5.1 Correctness

Theorem 1: if both the verifier and the prover follow the MF-PDP scheme, the prover can success passed the verification stage.

Proof: according the returned information p_t, the verifier computes:

$$P_t^e = (\prod_{j=1}^{c} T_{i_j}^{a_j})^e = (\prod_{j=1}^{c} (g^{m_{i_j}} \times r_{i_j})^{d*a_j} \bmod N)^e = (g^{\sum_{j=1}^{c} a_j * m_{i_j}} \times \prod_{j=1}^{c} r_{i_j}^{a_j}) \bmod N$$

As $p_r = \prod_{i=1}^{c} r_{i_j}^{a_j} \bmod N$,then $H((\frac{P_t^e}{P_r})^s) = H((\frac{(g^{\sum_{j=1}^{c} a_j * m_{i_j}} \times \prod_{i=1}^{c} r_{i_j}^{a_j}) \bmod N}{\prod_{i=1}^{c} r_{i_j}^{a_j} \bmod N})^s)$

$$= H((g^{\sum_{j=1}^{c} a_j * m_{i_j}} \bmod N)^s = H(g^{s\sum_{j=1}^{c} a_j * m_{i_j}} \bmod N) = H(g_s^{\sum_{j=1}^{c} a_j * m_{i_j}} \bmod N) = P_h$$

5.2 Detection Probability of Data Integrity Broken

The detection probability of data Integrity Broken P_x is an important evaluation part,which has relation to the total number of data block n ,data block loss number t , challenge data block number c , as follows:

$$P_x = P\{X \geq 1\} = 1 - P\{X = 0\} = 1 - \frac{n-t}{n} \times \frac{n-1-t}{n-1} \times \ldots \times \frac{n-c+1-t}{n-c+1}$$

Because $\frac{n-i-t}{n-i} \geq \frac{n-i-1-t}{n-i-1}, 0 \leq i \leq c-1$, thus $1 - \left(\frac{n-t}{n}\right)^c \leq P_x \leq 1 - \left(\frac{n-c+1-t}{n-c-1}\right)^c$

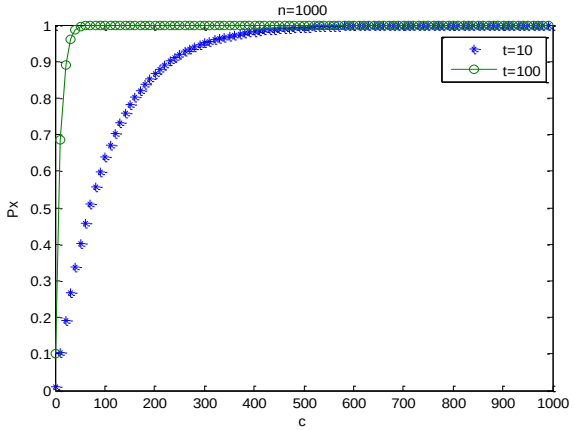

Figure 2. Detection Probability of Data Integrity Broken

Figure 2 show the relation of P_x , t , c , when n =1000. We can see that (1) with the same c ,

the P_x is increased as the t; (2) under the same P_x, c is decreased as the t's increase, e.g. When P_x =99%, n=1000, t=10, c is about 458, while t=100, c is about 43. Thus, the MF-PDP can find the data integrity broken status with appropriate parameters.

5.3 Security

5.3.1 **Provable data possession**

Theorem 2: Under assumptions of KEA1-r(Definition 1)and RSA(Definition 4) , the MP-PDP scheme is provable data possession in Random Oracle model.

Proof: We use the data possession game [3] to explain the security of MF-PDP. We assume that if the adversary A wins the PDP game with challenger C, then there is an extractor who helps the challenger C to extract the challenged data block or else the challenger can break the RSA problem or integer factoring.

We first look at the case when all the coefficients $a_1\ a_2 \ldots a_c$ are equal to 1. This case proves the challenger C possesses the sum of the requested data blocks. Then the coefficients $a_1\ a_2 \ldots a_c$ are random and distinct, which corresponds to the case that proves the challenger C possesses all the challenged data block.

The Challenge C and Adversary A play the PDP game as followings:
Setup:

C computes $g = y^2 \bmod N$, sets the public key $pk = (N, g)$ and sends pk to A.

Query:

C makes the tag queries adaptively: C selects a block m_1 and the position index i_1,then sends (m_1, i_1) to A. According to the (m_1, i_1),A generates the data block metadata T_{m_1,i_1} and sends it back to C who continues to query A for the metadata $T_{m_2,i_2}, T_{m_3,i_3} \ldots T_{m_n,i_n}$ on the blocks $m_2, m_3, \ldots m_n$ and index $i_2, i_3, \ldots i_n$.The only restriction is that C cannot make tag queries for two different blocks using the same index.
C answers the A's tag oracle queries as followings:

When A receives a tag query for a data block m and index i ,If a previous tag query has been made for the same m and index i, then A retrieves the recorded tuple(m, i, r_i, v_i), and returns $T_{m_1,i_1} = r_i$, else C randomly chooses $r_i \in QR_N$, $v_i \in Z_N$,and saves the data tuple (m, i, r_i, v_i) and return $T_{m_1,i_1} = r_i$.

Challenge:

C generates the $chal=(g_s, i_1, ..i_c)$,where $g_s = g^s \bmod N$, $s \in Z_N^*$, $i_1, ..i_c$ are the challenged data block index. $1 \le i_j \le n, 1 \le j \le c, 1 \le c \le n$,C sends $chal$ to A.

Forge:

According to the queried data block, A generates a data proof $V=(T,\rho)$,where $T = T_{m_{i_1}+m_{i_2}+..m_{i_c},\{i_1,..i_c\}}$,let $M = m_{i_1} + m_{i_2} .. m_{i_c}$,then return $V=(T,\rho)$ to C.

Analysis:

Case(1): $a_1\ a_2 \ldots a_c$ **are equal to 1**

As H is the random Oracle, with overwhelming probability, C can extract the pre-image value p_ρ that A utilized to calculate ρ . In the PDP game, the challenger C sends (g, g_s) to attack A who returns $V=(T,\rho)$, by which we can get $\tau = \dfrac{T^e}{\prod_{j=1}^{c} \upsilon_{i_j}}$ .Assume the adversary A can pass the verification, then there is $\tau^s = p_\rho$,thus, we can say adversary A returns implicitly the (τ, p_ρ) .

According to the KEA1-r, there exists a extractor ε help the challenger C extract a valid value M^* ,which satisfy $\tau = g^{M^*}$ (if $-g^{M^*} = \tau$,then Challenger sets $T = -T \bmod N$).

If $M^* = M$, then the challenger can successfully extract the challenged data block sum M .

If $M^* \neq M$,then there exits $g^{M^*} = g^M \bmod \Phi(N)$, that means the Challenger can compute a multiple of $\Phi(N)$ that is $g^{M^*} - g^M = K \times \Phi(N)$ for some $K \in Z - \{0\}$,the factorization of N can be efficiently computed[18].

We now could say our MF-PDP scheme support the possession of the sum of all the data blocks, which means the Challenger either get the challenged data block or break the factorization of big integrity.

Case (2): $a_1\ a_2 \ldots a_c$ **are random and pairwise distinct**

Note that each time in executing the MF-PDP scheme, there is a knowledge extractor ε can get a linear equation of the form $M_k = a_1^k m_{i_1} + a_2^k m_{i_2} .. + a_c^k\ m_{i_c}$ .By choosing independent coefficients $a_1^k \ldots a_c^k$ in c times on the same blocks $m_{i_1} \ldots m_{i_c}$, the ε obtains c independent linear equations, such that :

$$\begin{cases} M_1 = a_1^1 m_{i_1} + a_2^1 m_{i_2} .. + a_c^1 m_{i_c} \\ M_2 = a_1^2 m_{i_1} + a_2^2 m_{i_2} .. + a_c^2 m_{i_c} \\ .... \\ M_c = a_1^c m_{i_1} + a_2^c m_{i_2} .. + a_c^c m_{i_c} \end{cases}$$

By resolving above equation group, the $\mathcal{E}$ can extract the challenged data blocks $m_{i_1} \ldots m_{i_c}$.

Above all, under the assumption of ERA1-r and RSA, the MF-PDP satisfied the security property of provable data possession.

5.3.2 **Verification privacy**

Theorem 3: Under half honest model, the verification operation of the MF-PDP scheme is privacy against to the third party auditor.

Proof: we first build emulation as the view of verifier, and then we show the output of emulation is indistinguishable with output of verifier. The Input of emulation is come from the verifier.

The input of verifier is $\{N, g, e, R, H_R\}$, the output of verifier is bit b ,standing for " success" or "failure" of the *CheckProof* result. In the half-honest model, we assume the service provider (the prover) is honest, thus, the value of b is always 1, besides the set of $\{N, g, e, R, H_R\}$ is public.

Emulation E is composed of steps as follows:

Step1: E create random number s^1, k_1^1, k_2^1 ,and compute $g_s^1 = g^{s^1} \bmod N$.

Step2: E create random sequences $i_j = \pi_{k_1^1}(j)$ and challenge number c^1 ,and computes random coefficient $a_j = f_{k_2^1}(j)$

Step3: E sends challenge $< k_1^1, k_2^1, g_s^1, c^1 >$ to the prover who returns proof massage< P_t^1, P_h^1, R , H_R >.

Step4: using R , H_R, $\{a_j\}$, $pk = \{N, g, e\}$, E computes P_r^1 and compute

$$\tau = \left(\frac{\left(P_t^1\right)^e}{P_r^1} \right)^{s^1} = H((g_s^1)^{\sum_{j=1}^{c^1} a_i * m_{i_j}} \bmod N)$$ based on P_r^1, P_t^1.

Step5: let $x = \{N, g, e, R, H_R\}$ to be the input of verifier, then the output of E can be expressed as $output^E = (x, s^1, k_1^1, k_2^1, g_s^1, c^1, \tau^1)$.

Analysis: according to definition 2, the view of verifier is $view_v^p = (x, s, k_1, k_2, g_s, c, P_h)$, where

the distribution of $p_h = H(g_s^{\sum_{j=1}^{c} a_i * m_{i_j}} \bmod N)$ is determined by k_1, k_2, g_s, c.Under the same input x, as the distribution between $< s^1, k_1^1, k_2^1, g_s^1, c^1 >$ and $< s, k_1, k_2, g_s, c >$ is same, so the distribution between τ^1 and p_h is same too, this means the output of $E \quad output^E$ and $view_v^p$ are indistinguishable.

According to definition 2, the verifier know nothing about real data in executing the MF-PDP scheme except the input and output of the scheme, thus the MF-PDP scheme can keep the privacy against the third party auditor.

5.3.3 **Anti-replacing attack**

The MF-PDP scheme uses two policies to avoid the proof replacing attack.

(1) Each data block T_i is banded with random number r_i.

(2) The verification result is determined by the proof information from both the prover and the verifier.

Analysis:

For the first policy, without the random number r_i in computing data tag T_{m_i}, the data tags T_{m_1} for m_1 and T_{m_2} for m_2 are the same when the $m_1 = m_2$. The prover can modify the m_1 but use the m_2 and the corresponded T_{m_2} to replace the m_1 and T_{m_1} when computing proof. Thus the verifier can't find the modification of m_1. However, if using the random number r_i, the same content of different data block will make different data tags.

For the second policy, If the proof information is only provided by the prover, there may be proof replacing problem. Assume the prover returns directly the proof $V = (P_h, P_t, P_r)$ and the CheckProof is to compute whether $H((\frac{P_t^e}{P_r})^s)$ is equal to P_h, where $P_h = H(g_s^{a_1 m_{1_1} + a_2 m_{i_2} + .. a_c m_{i_c}} \bmod N) = H(g_s^{\sum_{j=1}^{c} a_j m_{i_j}} \bmod N)$, $P_t = \prod_{j=1}^{c} T_{m_{i_j}}{}^{a_j} = \prod_{j=1}^{c} (g^{m_{i_j}} \times r_{i_j})^{d * a_j} \bmod N$, $P_r = \prod_{j=1}^{c} r_{i_j}^{a_j} \bmod N$. For $chal = (k_1, k_2, g_s, c)$ which is response to data m_k block and data tag

$T_{m_{kj}}$, we can see that the prover can use other data block m_i and $T_{m_{ij}}$ to compute $V=(P_h,P_t,P_r)$ and this operation will not change the correctness of CheckProof, thus the proof replacing attack happened. The second policy migrate the computing task of P_r to the verifier, thus the returned proof information is $V=(P_h,P_t,R,H_R)$.

According to $chal=(k_1,k_2,g_s,c)$,the verifier choose the random number r_i from (R,H_R) and compute P_r with r_i. Thus, if the prover want to pass the CheckProof, he must compute the (P_h,P_t) with the challenged data block and data tag instead of other data block and data block tag. we use the digital signature to ensure the intergrity proof of random sequence R which is used to ensure the integrity of data block.

5.3.4 **Anti-replay attack**

The MF-PDP scheme uses the dynamically proof from $GenProof(pk,F,chal,\Sigma)\rightarrow V$ based on random $chal=(k_1,k_2,g_s,c)$ to resist replay attack with high probability.

Analysis:

If the integrity proof is static that means for the same data block the proof is same. Dynamic. Given to the $GenProof(pk,F,chal,\Sigma)\rightarrow V$, the proof is different if the $chal=(k_1,k_2,g_s,c)$ is different. The same value probability of two proofs is decided by the random function.

From $(k_1,k_2,s)\leftarrow Z_N$ and $c\leftarrow Z_n$, the equality probability of two different proof is $P=(\frac{1}{N}\times\frac{1}{N})^3\times(\frac{1}{n}\times\frac{1}{n})=\frac{1}{N^6\times n^2}$, when N and n big enough, the equality probability trend zero . Thus, the proposed scheme can resist the replay attack.

5.4 Performance

The MF-PDP performance analysis include three parts: (1)the Storage Cost : S_{OWN} , S_{CSP} , S_{TPA} and Storage Complexity: $O_{S_{OWN}}$, $O_{S_{CSP}}$, $O_{S_{TPA}}$ (2) Communication Cost: C_{com} , $O_{C_{com}}$, C_{Setup} , $C_{challange}$, C_{update} and Communication Complexity: $O_{C_{setup}}$, $O_{c_{challenge}}$, $O_{S_{update}}$ (3)Computing Cost: T_{OWN} , T_{CSP} , T_{TPA} , and Computing Complexity: $O_{T_{OWN}}$, $O_{T_{CSP}}$, $O_{T_{TPA}}$

5.4.1 **Storage cost**

（1） The OWN stores nothing except the public key pair, the cost of storage is independent of the data block number, so the complexity of storage is $O_{S_{OWN}} = O(1)$.

（2） The CSP stores the OWN's data file $F = \{m_i\}$, data block tags $T = \{T_{m_i}\}$, random numbers $R = \{r_i\}$ used to computing data block tags and the digital signature H_R of R. Thus, the cost of storage $S_{CSP} = |F| + |T| + |R| + |H_R|$, the max is $n \times l + (2n+1)|N|$. As the cost of storage is mainly dependent on the data block number, so the storage complexity is $O_{S_{CSP}} = O(n)$.

（3） The TPA stores nothing except the public key for verifying. The cost of storage is $S_{TPA} = |e| + |g| + |N|$, the max is $3|N|$ and the complexity of storage is $O_{S_{TPA}} = O(1)$.

5.4.2 **Communication cost**

（1） Setup stage

The partners of this stage are the OWN and the CSP. The content of communication include data file $F = \{m_i\}$, data block tags $T = \{T_{m_i}\}$, random numbers sequence $R = \{r_i\}$, the digital signature H_R. The communication cost of setup stage is $C_{Setup} = |F| + |T| + |R| + |H_R|$, the max is $n \times l + (2n+1)|N|$, the communication complexity is $O_{C_{setup}} = O(n)$.

（2） Challenge stage

The partners of this stage are the TPA and the CSP. The content of communication include the challenge $chal = (k_1, k_2, g_s, c)$ and the proof information $V = (P_h, P_t, R, H_R)$. The communication cost of Challenge stage is $C_{challange} = |k_1| + |k_2| + |g_s| + |c| + |P_h| + |P_t| + |R| + |H_R|$, the max is $(7+n)|N|$, the communication complexity is $O_{C_{challenge}} = O(n)$.

（3） Update stage

The partners of this stage are the OWN and the CSP. The content of communication include the random numbers $R = \{r_i\}$, the digital signature H_R. The updated data block

tag T_{m_x} (it is not used in data block deletion operation), the updated random numbers $R^{'}$ and The updated digital signature $H_{R^{'}}$,thus the cost of communication is C_{update} =$|R|+|H_R|+|R^{'}|+|H_{R^{'}}|+|T_{m_x}|$, the max is $(2n+3)|N|$,the communication complexity is $O_{S_{update}}=O(n)$.

Above all, the total communication cost is $C_{com}=C_{Setup}+C_{challange}+C_{update}$, the max is $n\times l+(2n+1)|N|+(7+n)|N|+(2n+3)|N|$ and the total communication complexity is $O_{C_{com}}=O(n)$.

5.4.3 **Computing cost**

(1) the OWN

The OWN's computing cost include the setup stage cost $t_{Own_{Setup}}$ and the data update stage $t_{Own_{update}}$.

In setup stage, the OWN computing cost include a public key pair creation t_{KeyGen}, n data block tag computing, $n\times t_{TagBlock}$,n random number creation $n\times t_{rand}$, a digital signature t_{sign} ,thus the setup stage computing cost is $t_{Own_{Setup}}=t_{KeyGen}+n\times t_{TagBlock}+n\times t_{rand}+t_{sign}$,the computing complexity is $O_{Own_{Setup}}=O(n)$.

In data update stage, the OWN cost include a data block tag creation T_{m_x} , a random number creation t_{rand} and a digital signature t_{sign}, thus the data update stage cost is $t_{Own_{update}}=t_{TagBlock}+t_{rand}+t_{sign}$, the computing complexity is $O_{Own_{Update}}=O(1)$.

Above all, the total computing cost of the OWN is $T_{own}=t_{Own_{Setup}}+t_{Own_{update}}=t_{KeyGen}+(n+1)\times t_{TagBlock}+(n+1)\times t_{rand}+2\times t_{sign}$ and the total computing complexity is $O_{T_{OWN}}=O(n)$.

(2) the CSP

The CSP's computing cost mainly from challenge stage as $t_{CSP_{challenge}}$.

In each challenge, the CSP's work include c pseudo random permutations $c\times t_{PSP}$, c

pseudo random functions $c \times t_{PSF}$, a p_h creation and a p_t creation t_{P_t}, the computing cost of the CSP is $T_{CSP} = c \times t_{PSP} + c \times t_{PSP} + t_{P_h} + t_{P_t}$, the computing complexity of the CSP is $O_{T_{CSP}} = O(c)$.

(3) the TPA

The TPA's computing work is in challenge stage. For each challenge, the TPA computing cost includes the challenge creation $t_{GenChal}$, c pseudo random permutations $c \times t_{PSP}$, c pseudo random functions $c \times t_{PSF}$,the digital signature verification t_{very}, the p_r creation t_{P_r},the comparison of proof $t_{compare}$,the total computing cost is $T_{TPA} = t_{GenChal} + c \times t_{PSP} + c \times t_{PSF} + t_{very} + t_{P_r} + t_{compare}$ and the computing complexity is $O_{T_{TPA}} = O(c)$.

6 Evaluation

6.1 Performance experiments

We implement the MF-PDP scheme with C++ and the MIRACAL. The test environment is a computer with configure 2.1GHz CPU,4G memory. The test parameters: N=143, p=11, q=13, g=4, file size is 4MB (=2^22 B), all the data is average value of ten times test.

Table 3. Pre-computing time of the OWN

Data block number n	Time (ms) of setup
64(2^6)	7.982
128(2^7)	20.245
256(2^8)	37.554
512(2^9)	102.167
1024(2^{10})	351.012

Table.4 Computing Cost of the OWN(or TPA) and The CSP ($l = 2^{12}$)

Challenged data block number c	Time of OWN (or TPA) (ms)	Time of CSP (ms)
10	19.281	3.877
20	41.487	5.013
50	72.322	19.338
100	97.528	23.783
1000	568.319	54.126

Table 3 shows that the computing cost of setup stage, which is increased with n and this result is in consisting to the performance analysis. Table 4 shows that the computing cost in challenge stage is low when the challenge is small. The computing cost of challenge stage is increase as the challenge number increase which is in consisting to the performance analysis before.

6.2 Comparison with other works

Table 3 show the comparison result between the MF-PDP and other PDP schemes. In functions support side, the MF-PDP is better. In the performance, the MF-PDP has some advantage in storage complexity of the OWN and computing complexity of the CSP. The MF-PDP has some disadvantage in communication complexity because it has download the random sequence R and the response digital signature H_R in each verification or data update stage. However, we can use some method to relieve, such as the data cache scheme that means the once the random sequence R and digital signature H_R is download, then the verifier only need to download the H_R. If the current download H_R is same to the previous saved one, the verifier will not need to download the R, thus the communication cost will be decrease much.

Table.5. Comparison of different works

works / Evaluation Index	[3] EPDP	[4]	[5]	[6] DPDP	[7]	[9]	[13]	MF-PDP
Type of Proof	P	P	D	P	P	P	D	P
Public verification	Yes	No	No	No	Yes	Yes	Yes	Yes
Data Dynamic	Yes*	Yes*	Yes	Yes	Yes	No	Yes	Yes
Verification Privacy	No	Yes	Yes	Yes	No	Yes	Yes	Yes
Sampling Verification	Yes	Yes	No	Yes	Yes	Yes	No	Yes
Unlimited times verification	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes
Anti-Replacing attack	Yes	Yes	Yes	No	Yes	Yes	No	Yes
Anti-Replay attack	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes
Communication Complexity	*O (1)*	*O (1)*	*O (1)*	*O (clogn)*	*O (clogn)*	*O (c)*	*O (1)*	*O (n)*
Storage complexity of the OWN	*O (1)*	*O (1)*	*O (n)*	*O (1)*	*O (1)*	*O (1)*	*O (n)*	*O (1)*
Computing complexity of the CSP	*O(c)*	*O (c)*	*O (n)*	*O(clogn)*	*O(clogn)*	*O (c)*	*O (n)*	*O (c)*
Computing Complexity of the TPA	*O (c)*	*O (1)*	*O (n)*	*O(clogn)*	*O (clogn)*	*O (c)*	*O (n)*	*O (c)*

Yes*: support partly operations P: Probabilistic D: Deterministic

n: data block number of one file. c: challenged data block number

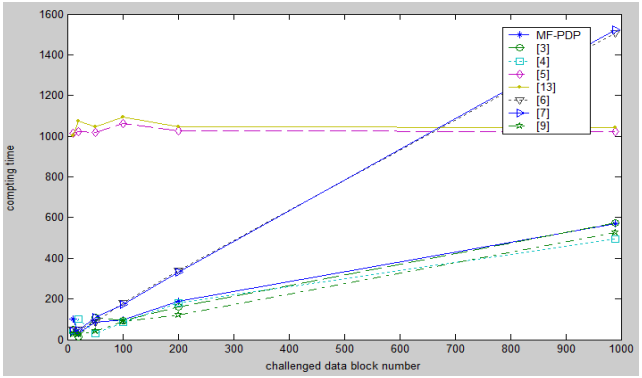

Figure 3. The challenged data block number and the TPA computing cost

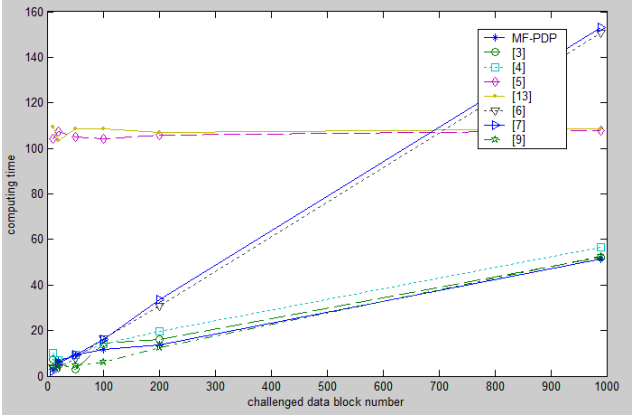

Figure 4. The challenged data block number and the CSP computing cost

Figure 3 and Figure 4 are the emulated results. It show the computing cost is consistent to the analysis result of table 2.The emulated result also suggest that the propose PDP is low complex and more effective than most of works.

7 Conclusion

For the data security in untrusted cloud, the PDP has been a tool to check data integrity. However, in cloud environment, most of the current works have limitation in performance or security. Based on current works and some new ideas, we designed a multi-function PDP scheme, which can meet different requirements of cloud computing environment. The further work is to study the relation among different functions and the multi-copy support where the data copy may store in different places.

Reference

[1]Hyun-A Park,Jae Hyun Park and Dong Hoon Lee. PKIS: practical keyword index search on cloud datacenter. EURASIP Journal on Wireless Communications and Networking 2011.2011:64. pp.1-16.

[2]Mithun Paul and Ashutosh Saxena Proof Of Erasability For Ensuring Comprehensive Data Deletion In Cloud Computing Recent Trends in Network Security and Applications Communications in Computer and Information Science, Springer, V.89, 2010, pp.340-348

[3]G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Pe-terson, and D. Song, Provable data possession at untrusted stores .In Proceedings of the 14th ACM conference on Computer and communications security , New York, USA, 2007,pp.598–609

[4]G.Ateniese, R.D. Pietro, L.V. Mancini, and G. Tsudik, Scalable and Efficient Provable Data Possession. In Proceedings of the Fourth Int'l Conf. Security and Privacy in Comm. Networks .SecureComm'08, New York: ACM, 2008,pp.1-10.

[5]Sebé F, Domingo J F, Martinez A B, et.al. Efficient remote data possession checking in critical information infrastructures. IEEE Trans on Knowledge and Data Engineering, 2007,20(8):1034-1038

[6] C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia, Dynamic provable data possession.In Proceedings of the 16th ACM conference on Computer and communications security , New York, USA, 2009, pp.213–222

[7] Q. Wang, C. Wang, J. Li, et al. Enabling public verifiability and data dynamics for storage security in cloud computing. LNCS 5789: Proceedings of 14th European Symposium on Research in Computer Security , Berlin:Springer, September 2009,pp.355–370

[8] C. Wang, Q. Wang, K. Ren ,et al, Ensuring Data Storage Security in Cloud Computing.//Proc. 17th Int'l Workshop Quality of Service (IWQoS '09),Piscataway,NJ:IEEE,2009,pp.1-9

[9] C.Wang, Q. Wang, K. Ren ,et al, Privacy-preserving public auditing for data storage security in cloud computing, In Proc of INFOCOM 2010.Piscataway,NJ:IEEE, March 2010,pp.1-9

[10] C.XU , HE Xiao-hu, Daniel Abraha.Cryptanalysis of auditing protocol proposed by Wang et al. for data storage security in Cloud Computing.2012, http://eprint.iacr.org/2012/115.pdf

[11]C. Wang, S. S.-M. Chow, Q. Wang, et.al , Privacy-preserving public auditing for secure cloud storage. Cryptology ePrint Archive, Report 2009/579, 2009. http://eprint.iacr.org/2009/579.pdf

[12]Y. Zhu, H. Wang, Z. Hu,et.al. Efficient Provable Data Possession for Hybrid Clouds. Cryptology ePrint Archive, Report 2010/234. http://eprint.iacr.org/2010/234.pdf

[13]Zhuo Hao, Sheng Zhong, Nenghai Yu.A Privacy-Preserving Remote Data Integrity Checking Protocol with Data Dynamics and Public Verifiability. IEEE TRANSAC TIONS ON KNOWLEDGE AND DATA ENGIN EERING, SEPTEMBER 2011, 23(9):1432-1437.

[14]P.Syam Kumar, R. Subramanian RSA-based dynamic public audit service for integrity verification of data storage in cloud computing using Sobol sequence Int. J. Cloud Computing, 2012,1(2/3): 167-200.

[15]Bo Chen, Reza Curtmola .Robust Dynamic Provable Data Possession.In Proceeding ICDCSW '12 Proceedings of the 2012 32nd International Conference on Distributed Computing Systems Workshops (ICDCSW '12). 2012.pp.515-525.

[16]M.Bellare and A.Palacio. The knowledge-of-exponent assumptions and 3-round zero-knowledge protocols.In CRYPTO 2004, LNCS 3152, Springer, 2004,pp.273–289.

[17]Oded.Goldreich.Foundations.of.Cryptography.Volume.2.Basic. Applications , New York: Cambridge University Press.2004.

[18]G.L.Miller. Riemann’s hypothesis and tests for primality. JCSS, 1976,13(3):300-317.