TIMER: Secure and Reliable Cloud Storage against Data Re-outsourcing

Tao Jiang[1], Xiaofeng Chen[1], Jin Li[2], Duncan S. Wong[3], Jianfeng Ma[1], and Joseph Liu[4]

[1] State Key Laboratory of Integrated Service Networks (ISN), Xidian University, Xi'an, P.R. China
jiangt2009@gmail.com, {xfchen,jfma}@xidian.edu.cn
[2] School of Computer Science, Guangzhou University, China
lijin@gzhu.edu.cn
[3] Department of Computer Science, City University of Hong Kong, Hong Kong
duncan@cityu.edu.hk
[4] Institute for Infocomm Research, Singapore
ksliu@i2r.a-star.edu.sg

Abstract. The semi-trusted servers in cloud environment may outsource the files of their clients to some low expensive servers to increase their profit. To some extent, such behavior may violate the wishes of cloud users and impair their legitimate rights and interests. In this paper, a probabilistic challenge-response scheme is proposed to prove that the clients' files are available and stored in a specified cloud server. In order to resist the collusion of cloud servers, common cloud infrastructure with some reasonable limits, such as rational economic security model, semi-collusion security model and response time bound, are exploited. These limits guarantee that a malicious cloud server could not conduct a t-round communication in a finite time. We analyze the security and performance of the proposed scheme and demonstrate that our scheme provides strong incentives for economically rational cloud providers against re-outsourcing the clients' data to some other cloud providers.

Keywords: Cloud storage, Economical server collusion, Storage security, Probabilistic scheme.

1 Introduction

In the cloud computing environment, the Cloud Storage Providers (CSPs) offer paid storage space on its infrastructure to store customers' data. Since the CSPs are not necessarily trusted in cloud storage system, efficient and secure schemes should be built to constrain their malicious activities.

For sensitive data, legitimate concerns are necessary when using cloud storage services. The failure of cloud storage server at Amazon results in the permanent loss of customer data [4]. Also, there are a variety of economical and legal restrictions that may compel a customer to choose to store data in a specific cloud storage provider. For example, many companies are willing to store

X. Huang and J. Zhou (Eds.): ISPEC 2014, LNCS 8434, pp. 346–358, 2014.

their sensitive data in the same cloud storage server and many privacy laws in Nova Scotia, British Columbia, Australia and EU [8] require personal data stored within a political border or other nations with comparable protections. Further, the cross server deduplication will greatly reduce the storage overhead of cloud servers, which will reduce the costs of the service providers and enhance their competitiveness. However, the data deduplication may violate the intention of users and undermine the interests of them. Therefore, we see that it is necessary to constrain the activity of the CSPs and verify that their activity meet the storage obligations. Since the clients data is stored in remote server without a local copy, it is very difficult to provide transparency to the users that their sensitive data is correctly handled by the cloud provider. We need to use challenge-response scheme to provide an efficient method to prove the malicious storage re-outsourcing activity. However, the existing challenge-response scheme could not provide a proof that the data of clients stored in a semi-trusted remote cloud storage is not re-outsourced in the economical server collusion network model [6, 7].

In this paper, we demonstrate that it is possible to design a challenge-response protocol which imposes a strong incentive onto the cloud providers to store their clients' data at rest. In particular, we present a probabilistic challenge-response scheme where semi-collusion bound, communication and computation bound and response time bound are adopted. A malicious cloud server S who has re-outsourced its client data to some other cloud server S' should conduct a t-round communication with S' to generate a correct response. If t is large enough, the malicious server could not generate the response in time even if with unlimited computation power. It is demonstrated that our scheme is secure under cryptography assumption and our analysis shows that as long as the designed communication round t is large enough, TIMER scheme will provide a strong incentive for the rational economic cloud providers to store the data of their clients in their storage servers.

2 Related Works

Provable Data Possession: To protect the availability of the clients' files stored in remote data storage server, Ateniese et al. [1] proposed a formalized model called Provable Data Possession (PDP). Unlike the low efficiency deterministic schemes [10, 14, 23] and probability scheme [22], PDP could efficiently check whether the clients' files stored in remote server have been tampered or deleted with very high probability. Several variations of their proposed scheme, such as static PDP schemes [11,18,24] and limited dynamic or dynamic schemes [2,13], are proposed to achieve efficient proof of remote data availability.

Secure Deduplication: Conducting deduplication will reduce the data storage burden and maintenance cost, which can promote price reductions of data storage service and enhance the competition of CSPs. Recent researches on storage deduplication [12, 25] show that deduplication achieves a higher level of scalability, availability, and durability. However, Harnik et al. [16] point out that

client-side deduplication introduces security problems that an attacker is able to get the entire file from the server by learning just a small piece of the hash value about the file. Therefore, Halevi et al. [15] proposed a scheme called Proofs of Ownership (PoWs), where a client proves to the server that it actually holds a copy of the file and not just some short information about it based on Merkle trees [20] and specific encodings.

Location Sensitive Services: In data storage system, users' data is important for some location sensitive services. Some schemes [19, 26] proposed to use semi-trusted landmarks to provide geolocation solutions for data storage. Also, to provide the security against the colluding of adversaries, hidden landmarks are used in geolocation system [5] in wireless networks. Bowers et al. [4] proposed an hourglass scheme to verify a cloud storage service provider is duplicate data from multiple drives through the measurements of network delay. Gondree and Peterson proposed a provable data geolocation and they detect the network delay of different distance and they point that their system could be built on any exiting PDP scheme.

The PDP relevant solutions are proposed to realize efficient data availability check on remote data storage servers. The data storage deduplication relevant solution PoWs is proposed to protect against an attacker from gaining access to potentially huge files of other users based on a very small amount of side information. However, all these schemes focus on the authentication of data integrity and availability problems between clients and servers, which could not prevent a semi-trusted server from re-outsourcing clients' data to some other servers to save its storage space or increase its profit. Such behavior may reduce the security and availability guarantee of clients' data and the benefit of the clients may be violated in this situation.

3 Problem Statement and Design Goals

3.1 System and Threat Model

In cloud storage environment, the clients' data may be re-outsourced multiple times and stored in some unknown servers with low quality of service, which will cause some serious economical and security problems in cloud storage outsourcing service.

Conspiracy to Profit: A CSP may offload the clients' data to some other CSPs when the sum of their payment is lower than that from its clients. Thus, on one hand, the CSP will be able to enlarge its profit by the difference between the payment of itself and the sum payment of all the other CSPs. Also, data re-outsourcing may be used to save the storage space to store the data from other clients. On the other hand, the colluded CSPs will get payment from the CSP. As a result, the conspiracy to profit model will promote the collusion of CSPs driven by the interests.

Storage Location Security: In the multiple time storage re-outsourcing scenario, the data owner will not be able to control the data re-outsourcing behavior of the malicious CSP and the location of its data is uncontrollable.

Therefore, the clients' data may be stored in some servers controlled by its competitors or in some servers beyond the scope of legal protection. Then, some data security and privacy issues will arise.

Low Service Quality: If the cloud storage service is provided in a multi-hop mode in storage re-outsourcing scenario, the CSPs may not be able to respond the request from their clients in time. Worse still, the CSPs will not be able to respond the clients when any CSP in the storage re-outsourcing chain is out of service. Also, the client's data will be stored in a lower payment data store which usually provides lower data security and quality of service guarantee.

3.2 Design Goals

To design a secure and practical TIMER scheme, our system should achieve the following security and performance guarantees.

1. Correctness: Any cloud server that faithfully follows the mechanism must produce an output that can be decrypted and verified by the customer.
2. Soundness: No cloud server can generate an incorrect output that can be decrypted and verified by the customer with non-negligible probability.
3. Efficiency: The local computations done by customer should be substantially less than the whole data.

4 Proposed Scheme

In this section, we first present the bounds of our scheme. Then, we introduce the definition and designing detail of our scheme.

4.1 Construction Overview

As the first idea, we have to make a cryptography design where the challenges from the client C could not be responded correctly in time, when a cloud server S storing the client's files F colludes with some other cloud server S' and re-outsources F to it. Thus, we propose probabilistic TIMER scheme based on communication time delay to prove that the client data is available and stored in specified data store. The proposed scheme adopts cryptographical assumption and network delay to restrict the collusion re-outsourcing behavior of cloud servers. We properly parameterize some bounds on the protocol as follow:

Semi-Collusion Bound (SC-Bound): *In TIMER scheme, every cloud provider runs the public key generation algorithm and produces a pair of keys* (pk, sk)*. The cloud provider then publicizes its public key* pk *and keeps its private key* sk *secret. It should be emphasized that a cloud provider will not conduct a full proxy signature delegation activity [28] with any other cloud server even in the collusion situation.*

Communication and Computation Bound (CC-Bound): *The cloud providers are rational. They would not like to sign for every possible combination*

of u tags, chosen from n tags, beforehand and outsourcing all the clients' files with these singed tags.(Actually, it is impossible for a cloud provider to conduct this kind of activity when u and n are relative large.)

Time Bound (T-Bound): *The time for a cloud storage provider to compute the proof TC is much less than the time to conduct a 1-round challenge-response commutation TT. The maximum time delay for an honest server to response the client in the TIMER scheme is Δt. S and S' could not conduct t-round communication in Δt, even if they collude with each other.(Since multi-hop re-outsourcing needs much more response time than the 1-hop re-outsourcing, we only need to analyze the 1-hop re-outsourcing security in our scheme.)*

With the above bounds, we could provide an explanation of our TIMER scheme based on network delay. On one hand, Papagiannaki et al. [21] showed that the single-hop average communication delay of packet in the backbone experience is around 0.1ms. On the other hand, Jansma et al. [17] showed that 10ms is needed to compute an RSA signature on an Intel P4 2.0GHz machine with 512MB of RAM. As in $T - Bound$, we assume that the cloud storage providers have much powerful computation ability which makes the time to generate a proof TC much less than the time TT. Thus the maximum time for an honest server S to response the challenge in TIMER scheme is $Time_1 = TT + tTC + \Delta t$ where TC is the proof computation time of S. The minimum time for a dishonest server S, who has re-outsourced file F to another server S', to response the challenge is $Time_2 = (t+1)TT + tTC'$ where TC' is the joint computation time of server S and S'. According to the $T - Bound$, TC and TC' is much less than TT and Δt is smaller than tTC. We obtain that $\frac{Time_2}{Time_1} = \frac{(t+1)TT+t\times TC'\times TC+\Delta t}{TT+t}$ and the challenger C will be able to prove that its file F is not stored in the data storage server S.

In general, TIMER scheme is a challenge-response scheme based on PDP and it forces S to conduct a t-round communication with S' when the file F is offloaded from S and stored at S'. The colluded servers S and S' would not be able to generate a correct proof in a time delay Δt if t is chosen properly.

4.2 TIMER Scheme

In this section, we present the constructions of TIMER scheme. We start by introducing some additional notations used by the constructions. Let $p = 2p' + 1$ and $q = 2q' + 1$ be secure primes and let $N = pq$ be an RSA modulus. Let g be a generator of QR_N, the unique cyclic subgroup of Z_N^* of order $p'q'$. We can obtain g as $g = a^2$, where $a \overset{R}{\leftarrow} Z_N^*$ such that $\gcd(a \pm 1, N) = 1$. All exponentiations are performed modulo N, and we sometimes omit writing it explicitly for simplicity. Let $h : \{0,1\}^* \rightarrow QR_N$ be a secure deterministic hash function that maps strings uniformly to QR_N. Let k, l, λ be security parameters (λ is a positive integer) and let H be a cryptographic full domain hash function as used in the provably secure FDH signature scheme [3, 9]. We get $H : \{0,1\}^k \rightarrow \mathbb{Z}_N^*$. In addition, we make use of a pseudo-random function (PRF) f and a pseudo-random permutation (PRP) π that $f : \{0,1\}^k \times \{0,1\}^{log_2(n)} \rightarrow \{0,1\}^l$ and $\pi : \{0,1\}^k \times \{0,1\}^{log_2(n)} \rightarrow \{0,1\}^{log_2(n)}$.

We write $f_k(x)$ to denote f keyed with key k applied on input x. The algorithms of TIMER scheme are described in Algorithm 1. We are able to maintain 1-round communication cost between C and S with a combined value ρ, and verification materials T_l and $\rho_l (0 \leq l \leq t-1)$.

As previously defined, let f be a pseudo-random function, π be a pseudo-random permutation and H be a cryptographic hash function.

According to the TIMER algorithms in Algorithm 1, We construct the TIMER system in two phases, Setup and Challenge:

Setup: The client C runs $\mathsf{Gen}_C(1^k) \rightarrow (pk_C, sk_C)$, stores (sk_C, pk_C) and sets $(N_C, g) = pk_C, (e_C, d_C, v) = sk_C$. C then runs $\mathsf{Tag}(pk_C, sk_C, b_i, i) \rightarrow (T_{i,b_i}, W_i)$ for all $1 \leq i \leq n$ and sends pk_C, F and $TAG = (T_{1,b_1}, ..., T_{n,b_n})$ to S for storage. C may now delete F and TAG from its local storage.

Challenge: C requests proof of possession for $c = ut$ distinct blocks of the file F (with $1 \leq c \leq n$):

1. C generates the challenge $CHAL = (r, k_0, k', g^s, u, t)$, where $k_1 \overset{R}{\leftarrow} \{0,1\}^k, k' \overset{R}{\leftarrow} \{0,1\}^k, g_s = g^s \bmod N, s \overset{R}{\leftarrow} \mathbb{Z}_N^*, CT_1$ is the machine time when C sends the challenge, u and t are used to decide the number of blocks to verify and the round number that S has to sign the intermediate results and Δt is the upper bound of time for S to respond a challenge. C sends $CHAL$ to S and stores the current system time CT_1.
2. S runs $\mathsf{Gen}_S(1^k) \rightarrow (pk_S, sk_S)$ and then runs $\mathsf{Prof}(pk_C, sk_S, F, CHAL, TAG) \rightarrow \mathcal{V}$ and sends to C the proof of possession $\mathcal{V}$.
3. When C receives the response from S, it stores the current system time CT_2. Then C sets $CHAL = (k_1, k', u, t, s, CT_1, CT_2, \Delta t)$ and checks the validity of the proof $\mathcal{V}$ by running Vrfy $(pk_S, pk_C, sk_C, CHAL, \mathcal{V})$.

It is obvious that, the additional tags do not change the storage requirements for the server, since the size of the file is $O(n)$. Considering the efficiency of the proposed scheme, we need to remark that $2t + 1$ values are needed among the communication between C and S. It means that the client needs to conduct t times signatures verification in each request. In the TIMER system, we consider a 1024-bit modulus N. In the Challenge phase, C sends to S 5 value with total 298 bytes (r and g_s are both 128 bytes, k_0 is 16 bytes, k' is 20 bytes, u is 4 byte and t is 1 byte). The values contained in the server's response are related with the communication round t and the total length is $(148t + 20)$ bytes ($T_l (0 \leq l \leq t-1)$ is 128 bytes, $\rho_l (0 \leq l \leq t-1)$ is 20 bytes and ρ is 20 bytes). The communication rounds t is decided according to Δt in full data re-outsourcing situation. However, in partial data re-outsourcing, it will grow when the allowed percent of the re-outsourced data becomes smaller, and we will provide a detailed analysis in the next section. According to out TIMER system above, we only need to send a small number of values and the server does not need to send back to the client the file blocks. The storage of a client is $O(1)$, and the communication overhead and computation overhead of a client are both $O(t)$.

Algorithm 1. The TIMER Algorithms

$\mathsf{Gen}_C(1^k)$:

1. Generate $pk_C = (N, g)$ and $sk_C = (e_C, d_C, v)$, such that $e_C d_C \equiv 1(\bmod p'_C q'_C)$, $e_C > \lambda$ is a large secret prime and $d_C > \lambda$, g is a generator of QR_N and $v \xleftarrow{R} \{0,1\}^k$.
2. Output (pk_C, sk_C).

$\mathsf{Gen}_S(1^k)$:

1. Generate $pk_S = (N, e_S)$ and $sk_S = (N, d_S)$, such that $ed \equiv 1(\bmod p'_S q'_S)$, $e_S > \lambda$ is a large secret prime and $d_S > \lambda$, g is a generator of QR_N.
2. Output (pk_S, sk_S).

$\mathsf{Tag}(pk_C = (N, g), sk_C = (d_C, v), b, i)$:

1. Generate $W_i = v||i$. Compute $T_{i,b} = (h(W_i) \cdot g^b)^{d_C} \bmod N$.
2. Output (T_i, b, W_i).

$\mathsf{Prof}(pk_C = (N, g), sk_S = d_S, F = (b_1, ..., b_n), CHAL = (r, k_0, k', g_s, u, t), TAG = (T_{1,b_1}, ..., T_{n,b_n}))$:

1. Let $c = ut$.
   for $0 \le l \le t-1$ **do**
   - **for** $1 \le j \le u$ **do**
     - Compute coefficients:$a_{l,j} = f_{k'}(ul + j)$; Compute the indices of the blocks for which the proof is generated: $i_{l,j} = \pi_{k_l}(ul + j)$;
   - Compute $T_l = (h(W_{i_{l,1}})^{a_{l,1}} \cdot ... \cdot h(W_{i_{l,u}})^{a_{l,u}} \cdot g^{a_{l,1}b_{i_{l,1}} + \cdot ... \cdot + a_{l,u}b_{i_{l,u}}})^{d_C}$;
   - Compute $\rho_l = (H(T_l||r))^{d_S}$. let $k_{l+1} = \rho_l$;
2. Compute $\rho = H(\prod_{l=0}^{t-1} g_s^{a_{l,1}b_{i_{l,1}} + \cdot ... \cdot + a_{l,u}b_{i_{l,u}}} \bmod N)$;
3. Output $\mathcal{V} = (\rho, T_0, ..., T_{t-1}, \rho_0, ..., \rho_{t-1})$.

$\mathsf{Vrfy}(pk_S = e_S, pk_C = (N, g), sk_C = (e_C, v), CHAL = (r, k_0, k, u, t, s, CT_1, CT_2, \Delta t), \mathcal{V} = (\rho, T_0, ..., T_{t-1}, \rho_0, ..., \rho_{t-1}))$:

if $CT_2 - CT_1 < \Delta t$ **then**
- Compute $T = T_0 \cdot ... \cdot T_{t-1} = (\prod_{l=0}^{t-1} h(W_{i_{l,1}})^{a_{l,1}} \cdot ... \cdot h(W_{i_{l,u}})^{a_{l,u}} g^{a_{l,1}b_{i_{l,1}} + ... + a_{l,u}b_{i_{l,u}}})$; **for** $0 \le l \le t-1$ **do**
  - Compute $H(T_l \parallel r) = \theta_l$ Let $k_{l+1} = \rho_l$ and $\tau = T^{e_C}$. **for** $1 \le j \le u$ **do**
    - Compute $a_{l,j} = f_{k'}(ul + j)$;// Compute $i_{l,j} = \pi_{k_l}(ul + j), W_{i_{l,j}} = v||i_{l,j}$, and ;
- **if** $\theta_l = (\rho_l))^{e_S} (0 \le l \le t-1)$ *and* $H(\tau^s \ mod \ N) = \rho$ **then**
  - Output Accept.

else
- Output reject.

5 Security and Performance Analysis

In this section, we present the security and performance analysis of our TIMER scheme.

5.1 Security Proof of TIMER Scheme

We suppose that the maximum time delay that a client allows the server to respond the proof is Δt. According to the $T - Bound$, when the file F is re-outsourced to S' from S, the collusion servers would not conduct a t round communication between each other. However, S will be able to forge a proof of possession $\mathcal{V}$ for the blocks indicated by $CHAL$ without conducting a t-round interaction with S'. Thus, we have to prove that the colluded servers could forge a valid proof in each phase $Ph_i(0 \leq i \leq t-1)$ with a negligible probability.

The initial key k_0, used to choose the blocks, is from the client while the phase key used to choose the blocks in each phase is generated from the result of the previous phase of the current phase. Thus, we have to prove that a Probability Polynomial Time (PPT) adversary will forge each phase key $k_l(1 \leq l \leq t-1)$ with only a negligible probability. If the PDP scheme [1] adopted in our scheme is secure under the RSA and KEA-r assumption [27], we could start the security proof for TIMER system by the phrase key unforgeability. We construct the phase key generation scheme according to our TIMER scheme as follow:

The Phrase Key Generation Scheme

Let f and π be the pseudo-random function and pseudo-random permutation respectively as defined before and some other parameters $r, k_0\ k'$, F and TAG are as defined in TIMER algorithms.

Key Generation: Compute$(N, e, d) \leftarrow \mathsf{GenRSA}\ (1^k)$ and the public key is (N, e) and the secret is d. Let $H : \{0,1\}^k \rightarrow \mathbb{Z}_N^*$ be a hash function.

Phrase Key Generation: When $k_l(1 \leq l \leq t-1)$ is needed to compute, the parameter T_{l-1} has been computed as defined in TIMER algorithms. Then compute $\rho_l = (H(T_l||r))^{d_S} \bmod N$ and $k_{l+1} = \rho_l$. At last, output $(r, k_0, k', \{T_0, T_1, ..., T_{t-1}\}, \{k_1, k_2, ..., k_{t-1}\})$.

Phrase Key Verification: Input $(r, k_0, k', \{T'_0, T'_1, ..., T'_{t-1}\}, \{k'_1, k'_2, ..., k'_{t-1}\})$ and check whether all the $\rho_l'^{e_S} \stackrel{?}{=} H(T'_l \parallel r)$ and $k'_{l+1} = \rho'_l$ where $0 \leq l \leq t-1$.

Theorem 1. *If the PDP scheme is provably secure under the RSA and KEA1-r assumption, and H and h are modeled as random oracles, the construction of the Phase Key Generation Scheme is unforgeable under adaptive chosen-message attack.*

Proof. On one hand, the PDP scheme is secure under RSA and KEA1-r assumption, which assures that an adversary can forge T_0 with only a negligible probability. On the other hand, the RSA-signature is a trapdoor permutation

and H is assumed to be a full-domain hash function, which guarantees that the output of the signature scheme in the phrase key generation scheme has a unique signature. Then, we get that the probability for an adversary to forge a phase key $k'_1 = \rho'_0$ and ${\rho'_0}^{e_S} = H(T'_0 \parallel r)$ is negligible. As a result, if a PPT adversary could forge the key k'_l with only a negligible probability, the probability that k'_{l+1} can be correctly generated is negligible. Consequently, the phase key $\{k_1, k_2, ..., k_{t-1}\}$ is unforgeable when the phase key seed k_0 is determined.

Theorem 2. *For any phase $Ph_l(0 \leq l \leq t-1)$, on input d_S, $\{k_0, k_1, ..., k_{l-1}\}$ and $\{T_0, T_1, ..., T_{l-1}\}$, the probability that S could forge a signature $T_l = (h(W_{i_{l,1}})^{a_{l.1}} \cdot ... \cdot h(W_{i_{l,u}})^{a_{l,u}} \cdot g^{a_{l,1}b_{i_{l,1}}+...+a_{l,u}b_{i_{l,u}}})^{d_C}$ is negligible without interacting with S'; On input $F = b_1, ..., b_n$, W and $\{k_0, k_1, ..., k_{l-1}\}$, the probability that S' could forge the phase key k_{l+1} is negligible without interacting with S.*

Proof. We model h as a random oracle. Then, the block identity $i_{l,j} = h_{k_{l-1}}(j)(1 \leq j \leq u)$ is uniform distribution. Since S has re-outsourced $F = b_1, ..., b_n$ to S', S could not compute a value $T_l = (h(W_{i_{l,1}})^{a_{l.1}} \cdot ... \cdot h(W_{i_{l,u}})^{a_{l,u}} \cdot g^{a_{l,1}b_{i_{l,1}}+...+a_{l,u}b_{i_{l,u}}})^{d_C}$ from a random set of u blocks without communicating with S'. Then, we have to prove that S' could not forge $T_\alpha(l+1 \leq \alpha \leq t-1)$ without communicating with S.

According to the $SC - Bound$, S would not conduct a full proxy signature with S'. Thus, S' will not get the secret key d_S of S. Under the assumption $SC - Bound$, We consider a game in which a challenger generates an RSA key (N, e), chooses random $m \in \{0,1\}^k$ and $r \in Z_N^*$, and sends $(N, e, y = h(m||r) \in Z_N^*)$ to adversary $\mathcal{A}$. The goal is for $\mathcal{A}$ to compute $y^{1/e}$ mod N. Assume that $\mathcal{A}$ can query the random oracle $H : Z_N^* \to Z_N^*$ at any sequence of points $x_1, ..., x_\ell \in Z_N^*$ receiving in return the output values $y_1 = H(x_1), ..., y_\ell = H(x_\ell)$ and, without loss of generality, these points are distinct. The challenger then gives to $\mathcal{A}$ the value $y^{1/e}$ mod N for all i, assuming that the challenger knows the factorization of N and can compute these values. We claim that $\mathcal{A}$ still can not compute a signature $\sigma = y^{1/e}$ mod N except with negligible probability. We construct the following adversary $\mathcal{A}'$ which computes $y^{1/e}$ mod N with the same probability at $\mathcal{A}$, but without any additional help from the challenger:

Algorithm 2. Algorithm $\mathcal{A}'$

Given (N, e, y) as input, and its goal is to compute $y^{1/e}$ mod N.

1. Run $\mathcal{A}$ on input (N, e, y).
2. Oracle Query:
   for *each random oracle query $H(x_i)$ of $\mathcal{A}$* **do**
   Choose $R_i \leftarrow Z_N^*$. Answer the query using $y_i := R_i^e$ mod N.
3. Give $R_1, ...R_\ell$ to $\mathcal{A}$, output whatever value is output by $\mathcal{A}$.

According to Theorem 1 and Theorem 2, if server S has re-outsourced F to S', it must conduct a t-round interaction with S' to generate a correct proof. Combining with $T - Bound$, they will not respond in time and the client will detect the malicious behavior of server S.

5.2 Probabilistic Analysis of Data Re-outsourcing

Our TIMER scheme is a probability scheme, where server S may re-outsource x blocks of the n-block file F to S' to break through the $T - Bound$. Let u be the number of different blocks for which client C asks proof in each phase of a challenge. Let X be a discrete random variable that is defined to be the number of blocks chosen by C that matches the blocks deleted by S. We could compute the probability P_X that at least one of the blocks picked by C matches one of the blocks re-outsourced to S' by S in each phases. We have:

$$\begin{aligned} P_X =& P\{X \geq 1\} = 1 - P\{X = 0\} \\ =& 1 - \frac{n-x}{n} \cdot \frac{n-x-1}{n-1} \cdot \ldots \cdot \frac{n-u+1-x}{n-u+1}. \end{aligned} \tag{1}$$

Since $\frac{n-i-x}{n-i} \geq \frac{n-i-1-x}{n-i-1}$, we get:

$$1 - (\frac{n-x}{n})^u \leq P_X \leq 1 - (\frac{n-u+1-x}{n-u+1})^u \tag{2}$$

If x blocks of F are offloaded to S' from S, P_X indicates the probability that a challenger C will detect the misbehavior of server S, when it asks proof for u out of n blocks. Since secure parameter t is the maximum communication rounds in a finite time delay Δt according to $T - Bound$, we have to guarantee that server S should conduct the communication rounds not less than t from the viewpoint of probability. Then, we get the relation between ideal communication rounds t and the actual communication rounds t' as $t' \times P_X = t$. That is:

$$t' = \frac{t}{P_X} \geq \frac{t}{1 - (\frac{n-x}{n})^u}. \tag{3}$$

Then, we assume that S re-outsources y out of n blocks of file F. Let Y be a discrete random variable that is defined to be the number of blocks chosen by C and matches the blocks destroyed by S. The probability P_Y that at least one of the blocks picked by C matches one of the blocks destroyed by S in each phase can be computed. We have:

$$\begin{aligned} P_Y =& P\{Y \geq 1\} = 1 - P\{Y = 0\} \\ =& 1 - \frac{n-y}{n} \cdot \frac{n-y-1}{n-1} \cdot \ldots \cdot \frac{n-u+1-y}{n-u+1}. \end{aligned} \tag{4}$$

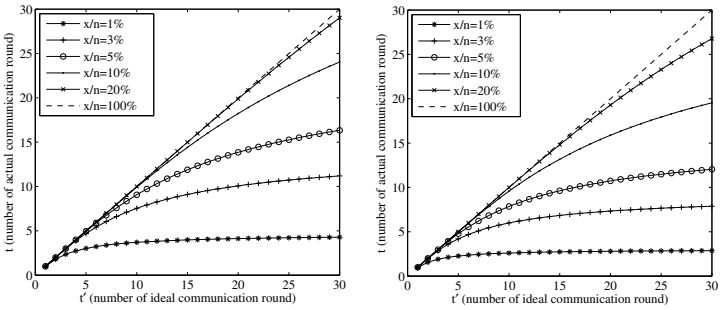
We define the probability P that, in all the t' phases, at least one of the blocks picked in each phase matches one of the blocks destroyed by S. we get:

$$P = 1 - [1 - P_Y]^{t'}. \tag{5}$$

Then we have:

$$1 - (\frac{n-y}{y})^{ut'} \leq P \leq 1 - (\frac{n-u+1-y}{n-u+1})^{ut'}. \quad (6)$$

Let $c = ut'$ be the number of blocks chosen for the proof of data availability in PDP [1]. The lower bound of inequalities (5) is the same as detection probability expressed in PDP. If y equals to 1% of n, then C needs to ask for $ut = 460$ blocks and $ut = 300$ blocks in order to achieve P of at least 99% and 95%, respectively. Fig.1 shows the relation between t' and t when the total blocks asked by C are 460 and 300 respectively.



(a) $ut = 460$ (b) $ut = 300$

Fig. 1. Relationship between the ideal communication round t and the actual communication round t' when different percents of file are re-outsourced

According to Fig. 1, it is obvious that $t' = t$ when F is re-outsourced from S to S'. However, when only a part of F is re-outsourced from S to S', the TIMER scheme should also be able to tolerate a t-round communication delay. Thus, t' rounds communication needs to be adopted to prevent from partial data re-outsourcing and the relation between t and t' is shown in Fig. 1. We need to choose appropriate t' to prevent against different percent of data re-outsourcing, because the smaller t' is , the more efficient our scheme will be. If t' is relatively small, it means the allowed response delay Δt is not very large compared with a 1-round communication time delay in the network. From this point of view, the efficiency of our scheme, to some extent, depends on precise measurement of the maximum response delay Δt.

TIMER scheme is efficient when t' is relative small and the total challenge block number $c = ut'$ is fixed. However, to detect the malicious activity of S, when only a small percent data is re-outsourced, 1 percent or even smaller, t' may become too large compared to t. As a result, $u = c/t'$ may become a relative small number and the package composition number C_n^u may not be large enough as a secure parameter. In this situation, we need to fix u and compute the relative t'. Therefore, the total number of random blocks that C challenges will be linear correlation with the actual communication rounds t'.

6 Conclusion

Server side clients' data re-outsourcing may cause some security problems in cloud storage environment. In this paper, the proposed probabilistic TIMER scheme will provide an efficient way to detect this malicious behavior of cloud servers. It adopts cryptographic assumptions and network delay to prevent servers from collusion in cloud storage, which will provide a strong incentive for the economically rational cloud server to store clients' data in their stores. We provide a security and performance analysis of our scheme. The analysis shows that our scheme is secure and efficient. The storage overhead of clients in TIMER scheme is $O(1)$ and the computation and communication overhead are both $O(t)$ in full data re-outsourcing scenario and the client storage overhead, computation and communication overhead become $O(1)$, $O(t')$ and $O(t')$, respectively. However, t and t' will become relative large when only a small percent of clients' data is re-outsourced. In the future, we will explore some new methods to construct a scheme with constant computation and communication overhead.

Acknowledgement. This work is supported by the National Natural Science Foundation of China (Nos. 61272455 and 61100224), Doctoral Fund of Ministry of Education of China (No. 20130203110004), Program for New Century Excellent Talents in University (No. NCET-13-0946), China 111 Project (No. B08038), and the Fundamental Research Funds for the Central Universities (No. BDY151402).

References

1. Ateniese, G., et al.: Provable data possession at untrusted stores. In: Proc. of ACM CCS, Virginia, USA, pp. 598–609 (October 2007)
2. Ateniese, G., et al.: Scalable and efficient provable data possession. In: Proc. of SecureComm, VA, USA, pp. 1–10 (September 2008)
3. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing effficient protocols. In: Proc. of ACM CCS, VA, USA, pp. 62–73 (November 1993)
4. Bowers, K.D., et al.: How to tell if your cloud files are vulnerable to drive crashes. In: Proceedings of the ACM Conference on Computer and Communications Security, IL, USA, pp. 501–514 (October 2011)
5. Capkun, S., Cagalj, M., Srivastava, M.: Secure localization with hidden and mobile base stations. In: Proceedings of the IEEE International Conference on Computer Communications, Catalunya, Spain, pp. 1–10 (April 2006)
6. Chen, X., Li, J., Susilo, W.: Efficient fair conditional payments for outsourcing computations. IEEE Transactions on Information Forensics and Security 7(6), 1687–1694 (2012)
7. Chen, X., Li, J., Ma, J., Tang, Q., Lou, W.: New algorithms for secure outsourcing of modular exponentiations. In: Foresti, S., Yung, M., Martinelli, F. (eds.) ESORICS 2012. LNCS, vol. 7459, pp. 541–556. Springer, Heidelberg (2012)
8. Commission, E.: Regulation of the european parliament and of the council on the protection of individuals with regard to the processing of personal data and on the free movement of such data. general data protection regulation, directive 95/46/EC (2012)

9. Coron, J.-S.: On the exact security of full domain hash. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 229–235. Springer, Heidelberg (2000)
10. Deswarte, Y., Quisquater, J.J., Saidane, A.: Remote integrity checking. In: Proc. of Conference on Integrity and Internal Control in Information Systems (IICIS 2003), Lausanne, Switzerland, pp. 1–11 (November 2003)
11. Dodis, Y., Vadhan, S., Wichs, D.: Proofs of retrievability via hardness amplification. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 109–127. Springer, Heidelberg (2009)
12. Dubnicki, C., et al.: Hydrastor a scalable secondary storage. In: Proc. of the 7th USENIX Conference on File and Storage Technologies, CA, USA, pp. 197–210 (February 2009)
13. Erway, C., Kupcu, A., Papamanthou, C., Tamassia, R.: Dynamic provable data possession. In: Proc. of ACM CCS. pp. 213–222. Illinois, USA (November 2009)
14. Filho, D.L.G., Baretto, P.S.L.M.: Demonstrating data possession and uncheatable data transfer. IACR ePrint archive 2006 (2006), http://eprint.iacr.org/2006/150
15. Halevi, S., Harnik, D., Pinkas, B., Shulman-Peleg, A.: Proof of ownership in remote storage system. In: Proc. of ACM CCS, Illinois, USA, pp. 491–500 (October 2011)
16. Harnik, D., Pinkas, B., Shulman-Peleg, A.: Side channels in cloud services: Deduplication in cloud storage. In: Proc. of IEEE Security & Privicy, CA, USA, pp. 40–47 (November 2010)
17. Jansma, N., Arrendondo, B.: Performance comparison of elliptic curve and rsa digital signatures. Tech. Rep. MI, University of Michigan, Ann Arbor (May 2004)
18. Juels, A., Kaliski, B.S.: Pors: Proofs of retrievability for large files. In: Proc. of ACM CCS, Virginia, USA, pp. 584–597 (2007)
19. Laki, S., et al.: A detailed path-latency model for router geolocation. In: The International Conference on Testbeds and Research Infrastructures for the Development of Networks Communities and Workshops, DC, USA, pp. 1–6 (April 2009)
20. Merkle, R.C.: A certified digital signature. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 218–238. Springer, Heidelberg (1990)
21. Papagiannaki, K., et al.: Provable data possession at untrusted stores. In: Proc. IEEE INFOCOM 2002, NY, USA, pp. 535–544 (June 2002)
22. Schwarz, T.S.J., Miller, E.L.: Store, forget, and check: Using algebraic signatures to check remotely administered storage. In: Proceedings of ICDCS 2006, Lisboa, Portugal, pp. 1–12 (July 2006)
23. Sebe, F., et al.: Time-bounded remote file integrity checking. Tech. Rep. 04429, Universitat Rovira i Virgili, Tarragona, Spain (July 2004)
24. Shacham, H., Waters, B.: Compact proofs of retrievability. In: Proc. of ASIACRYPT, Melbourne, Australia, pp. 90–107 (December 2008)
25. Ungureanu, C., et al.: Hydrafs A high-throughput file system for the hydrastor content-addressable storage system. In: Proc. of the 8th USENIX Conference on File and Storage Technologies, CA, USA, p. 17 (February 2010)
26. Wong, B., Stoyanov, I., Sirer, E.G.: Octant: A comprehensive framework for the geolocalization of internet hosts. In: Proceedings of the USENIX Networked Systems Design and Implementation, MA, USA, pp. 313–326 (April 2007)
27. Yamamoto, G., Fujisaki, E., Abe, M.: An efficiently-verifiable zero-knowledge argument for proofs of knowledge. IEICE Technical Report ISEC2005-48 105, 41–45 (July 2005)
28. Zhang, F., Kim, K.: Efficient id-based blind signature and proxy signature from bilinear pairings. In: Safavi-Naini, R., Seberry, J. (eds.) ACISP 2003. LNCS, vol. 2727, pp. 312–323. Springer, Heidelberg (2003)