Server Re-Tag Provable Data Possession in Public Cloud

Yongjun Ren[1,2], Jiang Xu[1,2], Jin Wang[1,2], Jeong-Uk Kim[3]

[1]Jiangsu Engineering Center of Network Monitoring，Nanjing University of Information Science & Technology, Nanjing 210044, China

[2]Computer and Software School, Nanjing University of Information Science & Technology, Nanjing 210044, China

[3]Department of Energy Grid, Sangmyung University, Seoul 110-743, Korea

Abstract. Integrity checking becomes imperative to secure data in a cloud environment. In this paper, we propose server re-tag provable data possession (SRT-PDP). By utilizing proxy re-signatures, we allow the cloud severs as the group manager to re-sign blocks after the client sign the shared data, so that existing clients do not have to do anything even if some clients leave the group. In addition, a public verifier is always able to audit the integrity of shared data without retrieving the entire data from the cloud servers. SRT-PDP enable the cloud server takes the most of the work, and reduce the burden of client, which is suit for mobile device.

Keywords: cloud computing, data storage auditing, provable data possession, proxy re-signature

1 Introduction

In the cloud paradigm, data owners move the large data files from their local computing systems to the remote servers. It is of critical importance for the data owners can avoid the initial investment of expensive infrastructure setup, large equipment, and daily maintenance cost, which is particularly true for small and medium-sized businesses. Moreover the data owners can rely on the Cloud to provide more reliable services, so that they can access data from anywhere and at any time.

Storing the data in cloud environment becomes natural and also essential. But, security becomes one of the major concerns for all entities in cloud services. Data owners need to be convinced that their data are correctly stored in the Cloud. It is desirable to have data integrity verification to assure data are correctly stored in the Cloud. In order to solve the problem of data integrity verification, many schemes are proposed under different systems and security models [1-13].

With the widespread use of mobile electronic devices, cloud storage platform has the incomparable advantage. In such environment, mobile devices often join or leave the group of the shared data. When some members leave the group, the shared data blocks signed by they are hard to audit. Moreover there is such scenario: Small companies usually stores their data to the cloud server, over a period of time the company goes bankrupt, the company does not deal with these data. However, these data are likely to be very valuable to the public.

To solve the problems, we propose server re-tag provable data possession (SRT-PDP). We allow the cloud severs to re-sign blocks after the client sign the shared data, so that existing clients do not have to do anything even if some clients leave and discarded the stored data. In addition, a public verifier is always able to audit the integrity of the data without retrieving the entire data from the cloud servers. SRT-PDP enable the cloud server takes the most of the work, and reduce the burden of client, which is suit for mobile device.

2 Related Work

Based on the pre-computed MACs stored on the verifier, the protocols proposed by Lilli bridge et al.[2] and Naor et al.[3]can detect any data loss or corruption with high probability. Sha cham et al. [4] proposed a MAC-based batch verification for multiple data blocks. In 2007 Ateniese, et al [5] proposed a PDP model to solve the storage problems of files. They divided the file into blocks, and computed a homomorphic tag [6] for each block, completed the proof of the data integrity by sampling and verifying the correspondence of the tags and blocks randomly. Havav Shacham and Brent Waters [4] proposed an improved POR model under the security model defined in [7], and had a very complete proof. Kevin D. Bowers et al [8] and Yevgeniy Dodis et al [9] made some theory and application extensions based on [4][7]. Zheng and Xu also present a dynamic POR model in [10]. Ateniese improved PDP model to apply to public authentication in [11]. They replaced the homomorphic tags in [5] with homomorphic tags supported public authentication [12].C.Erway[13] proposed dynamic PDP model based on PDP model. It maintained a skip-list for tags, and stored the root metadata in Client's hand to prevent replay attack. Qian Wang, et al [14] use the tags based on [4] to apply the data integrity verification of dynamic files. Its computation and communication were both smaller than DPDP model. Zhu et. al present a cooperative PDP (CPDP) scheme based on homomorphic verifiable response and hash index hierarchy [15]. Recently Wang et. al proposes a public auditing scheme with client revocation, but the scheme assumes that there is secure secret channel prior to execution of the scheme between the every communicated pairs, which is too high and difficult to achieve in the real environment.

3 Server Re-Tag Provable Data Possession

3.1System model

SRT-PDP system consists of three different network entities: Client, CCS, Third Party Auditor (TPA). They can be identified below.

1) Client: an entity, which has massive data which will be moved to CPS for maintenance and computation, can be either individual consumer or organization;

2) Cloud Storage Server (CSS): an entity, which is managed by cloud service provider, has significant storage space and computation resource to maintain the clients' data;

3) Third Party Auditor: an entity, which has expertise and capabilities that clients do not have, is trusted to assess and expose risk of cloud storage services on behalf of the clients upon request.

3.2 Our SRT-PDP

Let and be two groups of order , is a generator of , $e: G_1 \times G_1 \rightarrow$ is a bilinear map, is a different random element from in ,two hash functions $H_1: \{0,1\}^* \rightarrow$, $H_2: \{0,1\}^* \rightarrow$ .The public parameters are ($e, q, g, G_1, G_2, u, H_1,$). We assume that shared data file *F* (potentially encoded using Reed-Solomon codes [11]) is divided into *n* blocks $\{m_1, m_2, \cdots, m$.The procedure of our basic scheme execution is as follows:

KeyGen$(1^k) \rightarrow (sk, pk)$

The client *A* chooses a random $a \in$and compute $pk_A =$.The secret key is and the public key is . The cloud server randomly selects $s \in$ as private key , and compute $spk =$ as public key.

ReKey $\rightarrow$

The cloud server generates a re-signing key as follows: The cloud server generates a random $r \in$ and sends it to the client *A*.The client *A* sends r to cloud server.The cloud server computes $rk_{A \rightarrow S} = s/a \in$, where is private key of the cloud server.

TagGen$(sk_A, F) \rightarrow T_A$

Given*F*= $\{m_1, m_2, \cdots, m$, the client *A* with the private key generates the tag of the block : $\sigma_i = (H_1(f_i)u^m$, where block identifier of is , then denotes the set by $\Phi = \{$, $1 \leq i n$ as the tag for block .The clients ends ={ $F\}$ to the cloud server and deletes them from its local storage.

ConvertTag$(T_A, rk_{A \rightarrow s}) \rightarrow T_s$

When the cloud server receives , it first checks the following formula: $e(\sigma_i, g) = e(H_1(f_i)u^{m_i}, pk$. If the verification fails, reject by the cloud server; otherwise it convert tag of the client *A* into its tag on the same block as follows. The cloud server computes = $\sigma_i^{rk_{A \rightarrow S}} = (H_1(f_i)u^{m_i})^{a \cdot s/a} = (H_1(f_i)u^m$. Let $\Phi' = \{$.Then and *F* are stored.

GenChal$(k) \rightarrow chal$

The TPA can verify the integrity of the shared data. It picks a random *c*-element subset *I* of the set [1,*n*]. For $i \in$($1 \leq i \leq$), the TPA chooses a random element $v_i \in$ and sends the message *chal*= $\{(i, v_i)\}$ to the cloud server.

Genproof $(F, \phi', chal) \rightarrow TPA$

Upon receiving the challenge, the cloud server computes:

$$\sigma = \prod_{i=1}^{c} \sigma \ , \qquad \mu = \sum_{i=1}^{c} v_i$$

The cloud server outputs *pf*={ , μ}and sends it to the verifier.

$Verify\ \Pr oof\ (pk, chal, pf) \rightarrow \{true, false\}$

Upon receiving the response *pf* from the CSS, the TPA checks the correctness of the auditing proof as

$$e(\sigma, g) = e(\prod_{i=1}^{c} H_1(f_i)^{v_i} \cdot u^{\mu}, sp) \quad (1)$$

If so, output "*true*"; otherwise "*false*".

3.3 Security Analysis

The correctness analysis and security analysis of our SRT-PDP scheme can be given by the following theorems.

Theorem 1: If the proposed procedures follow as above, any TPA is able to correctly check the integrity of shared data.

Proof: According to our scheme procedures, we know that

$$e(\sigma, g) = e(\prod_{i=1}^{c} \sigma_i^{v_i}, g)$$

$$= e(\prod_{i=1}^{c} (H_1(f_i) u^{m_i})^{v_i}, g)$$

$$= e(\prod_{i=1}^{c} (H_1(f_i) u^{m_i})^{v_i}, g^s)$$

$$= e(\prod_{i=1}^{c} H_1(f_i)^{v_i} \cdot \prod_{i=1}^{c} u^{m_i v_i}, sp)$$

$$= e(\prod_{i=1}^{c} H_1(f_i)^{v_i} \cdot u^{\mu}, spk)$$

Theorem 2: The proposed SRT-PDP scheme is computational infeasible to generate a forgery of an auditing proof under our mechanism.

Due to limited space, here we omit the proof. The proof will appear in the full version of the paper.

4 Performance Evaluation

In this section, we first discuss the communication and computation cost of our mechanism.

Communication Cost

According to the description, the size of an auditing message $\{(i, v_i)\}$ is $c \cdot (|n| + |$ bits, where *c* is the number of selected blocks, is the size of an element[1,*n*] and is the size of an element of .The size of an auditing proof{ , μ} is 2 $|p| + c \cdot |$ bits, where is the size of an element of or , is the size of a block identifier. Therefore, the total communication cost of an auditing task is 2 $|p| + c \cdot (|f| + |n| + |$ bits.

Computation Cost

As shown in Convert Tagof our mechanism, the cloud first verifies the correctness of the original signature on a block, and then computes a new signature on the same block with a re-signing key. The computation cost of re-signing a block in the cloud is 2 Exp + Mul +2Pair+ Hash, where Exp denotes one exponentiation in , Muldenotes one multiplication in ,Pair denotes one pairing operation, and Hashdenotes one hashing operation in .The cloud can further reduce the computation cost of the re-signing on a block to Expby directly re-sign in git without verification. The public auditing performed by theTPA ensures that the re-signed blocks are correct. Moreover Convert Tag algorithm is performed by cloud server. In generally we assume that the computable ability of the cloud server is unlimited. Thus our scheme reduces the burden of the clients.

5 Conclusions

In this paper, we propose server re-tag provable data possession (SRT-PDP). By utilizing proxy re-signatures, we allow the cloud severs as the group manager to re-sign blocks after the client sign the shared data, so that existing clients do not have to do anything even if some clients leave the group. In addition, a public verifier is always able to audit the integrity of shared data without retrieving the entire data from the cloud servers. SRT-PDP enable the cloud server takes the most of the work, and reduce the burden of client, which is suit for mobile device.

Acknowledgement. This work was supported by Jiangsu Province Universities Natural Science Research Program (NO.11KJB510010) and Jiangsu Province Research and Innovation Project for College Graduates (NO.CXZZ12_0515). This work was also supported by the Industrial Strategic Technology Development Program (10041740) funded by the Ministry of Trade, Industry and Energy (MOTIE, Korea). Prof. Jeong-Uk Kim is the corresponding author.

References

1 Kan Yang, XiaohuaJia. Data storage auditing service in cloud computing: challenges, methods and opportunities.The journal of World Wide Web. July 2012, Volume 15, pp 409-428.
2 Lillibridge, M., Elnikety, S., Birrell, A., Burrows, M., Isard, M.: A cooperative internet backup
3 Naor, M., Rothblum, G.N.: The complexity of online memory checking. In: Proceedings of the46th Annual IEEE Symposium on Foundations of Computer Science, FOCS ’05, pp. 573–584.
4 IEEE Computer Society, Washington, DC, USA (2005)H. Shacham and B. Waters. Compact proofs of retrievability.In ASIACRYPT '08, pp. 90-107, 2008.
5 G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song. Provable data possession at untrusted stores.In CCS '07, pp.598-609, 2007.

6 R.Johnson, D.Molnar, D.song, and D.wagner. Homomorphic signature schemes.In Proc. of CT-RSA, volume 2271 of LNCS, pp. 244-262, 2002.
7 A.Juels and B. Kaliski. PORs: Proofs of retrievability for large files. In CCS '07, pp.584-597, 2007.
8 K. Bowers, A. Juels, and A. Oprea. Proofs of retrievability: Theory and implementation. Technical Report 2008/175, Cryptology ePrint Archive, 2008.
9 Y. Dodis, S. Vadhan, and D. Wichs. Proofs of retrievability via hardness application. In TCC, vol.5444 of LNCS, pp. 109-127, 2009
10 QingjiZheng and ShouhuaiXu.Fair and Dynamic Proofs of Retrievability.CODASPY'11, February 21–23, 2011, San Antonio, Texas, USA.
11 G.Ateniese, S.Kamara, and J.Katz. Proofs of storage from homomorphic identification protocols. ASIACRYPT'09, LNCC, 2009.
12 DBoneh, B Lynn, H Shacham. Short signatures from the weil pairing. ASIACCRYPT 2001.LNCS, vol. 2248, pp. 514-532, 2001.
13 C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia. Dynamic provable data possession.In CCS '09, pp. 213-222, 2009.
14 Q.Wang, C.Wang, J. Li, K. Ren, and W. Lou. Enabling public verifiability and data dynamics for storage security in cloud computing. IEEE Transactions on Parallel and Distributed Systems (TPDS), Vol. 22, No. 5, pp. 847-859, May, 2011.
15 Yan Zhu,HongxinHu,Gail-JoonAhn and Mengyang Yu. Cooperative Provable Data Possessionfor Integrity Verification in Multicloud Storage. IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 23, NO. 12, 2012