Designated-Verifier Provable Data Possession in Public Cloud

YongjunRen[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 designated verifier provable data possession (DV-PDP). In public clouds, DV-PDP is a matter of crucial importance when the client cannot perform the remote data possession checking. We study the DV-PDP system security model. We use ECC-based homomorphic authenticator to design DV-PDP scheme. The scheme is very suitable for mobile clouds because expensive bilinear is not needed, which result in small amount of calculation and Communications.

Keywords: cloud computing, data storage auditing, provable data possession

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].

In this paper, we propose the concept of Designated-Verifier Provable Data Possession (DV-PDP). Then, we give DV-PDP system model and formal DV-PDP security model. In DV-PDP, data owners can designate a verifier to verify data integrity of his data. The verifier is stateless and independent from CSP, which solves the problem that the verifier can be controlled by the malicious CSP. In our design, we propose to use ECC-based homomorphic authenticator to design PDP scheme, which does not compute expensive bilinear and consume small amount of calculation and Communications. Our scheme is very suitable for mobile clouds.

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. Shacham et al. [4] proposed a MAC-based batch verification for multiple data blocks. In 2007 Ateniese, et al [5] proposed a PDPmodel 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.

Recently Shen et al. presented delegable provable data possession scheme [14], in which data owner generates the delegation key for delegated verifier and store the key in CPSs for verification. A malicious CPS can control the delegation key and lead to the failure of the subsequent validation work. Wang et al. [15] also proposed a proxy provable data possession (PPDP) model and provided a construction for it. In PPDP data owner can delegate its remote data possession checking capability to the proxy by sending it a warrant. The warrant will be stored both in the proxy and CPS. Before the verification of the data, the both warrant are checked for consistency. If the CPS is malicious, it can reject all queries from the proxy and interrupt the implementation of the scheme. The problem is that the proxy or delegated verifier is not stateless to CSP. While the verifiers should be stateless, since such state is difficult to maintain if the verifier's machine crashes or if the verifier's role is delegated to third parties or distributed among multiple machine [4][11].

3 Designated Verifier Provable Data Possession

3.1System model Security model

DV-PDP system consists of three different network entities: Client, PCS, DV.

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) Designated Verifier (DV): an entity, which is trusted and designated to assess and expose risk of cloud storage services on behalf of the Clients.

3.2 Security model

Definition 1 (Unforgeability): DV-PDP protocol is secure if for any (probabilistic polynomial) adversary *A* (i.e., malicious PCS) the probability that *A* wins the DV-PDP game is negligible. The DV-PDP game between the challenger *C* and the adversary *A* can be depicted as follows:

(1) SetUp: Suppose the system parameter is *params*. *KeyGen* is a private/public key pair generating algorithm. By running *KeyGen*, *C* can get the client's private/public key pair *(x,X)*, the designated verifier's private/public key pair *(y,Y)*. *C* keeps *x*, *y* confidential and sends (*X*,*Y*) to *A*.

(2) Queries: *A* adaptively makes a number of different queries to *C*. Each query can be one of the following.

• Hash query. *A* makes Hash function queries adaptively. *C* responds the Hash values to *A*;

• Proof query. *A* chooses challenge *chal* and obtains a valid proof with the *chal*.

(3) Challenge: *C* generates a challenge *chal* which defines a ordered collection. *C* is required to provide a possession proof for the blocks.

(4) Answer: *A* computes a data possession proof *pf* for the blocks indicated by chal and returns *pf* .

We say that the success probability of the adversary *A* is negligible. i.e.

$$Adv_A(\Pr[Verifyproof = true]) \le \varepsilon$$

where is negligible.

3.3 Our DV-PDP

Let be a cyclic multiplicative group on ECC generated by *g*, two hash functions $H_1, H_2: \{0,1\}^* \rightarrow$, viewed as a random oracle. The procedure of our scheme execution is as follows:

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

The client choose a random $x \in$ and compute $X =$.The secret key is *x* and the public key is *X*. The client designates a trust verifier DV. DV run the *KeyGen* and randomly choose $y \in$ as his private key and computes $Y =$ as his public key.

$TagGen(x, Y, m) \rightarrow T_m$

Given *F*= $\{m_1, m_2, \cdots, m$, the client generates the tag T_m of the block m_i. Let i and i are random integer in .The client computes them as follows: $k_{i1} \| k_{i2} = H_1(Y^x, n$ .Client compute $\sigma_{i,1} = (Y^{H_2(m_i)k_{i1}+k_{i2}}$ $\sigma_{i,2} = X$, $\sigma_{i,3} = X^{|}$, then denote the set by $\Phi = \{\sigma_{i,1}, \sigma_{i,2}, \sigma, \quad 1 \le in$ as the tag for block .

The client sends T_m ={ F} to the CSS and deletes them from its local storage.

$GenChal(k) \rightarrow chal$

Verifier picks a random subset *I* of the set [1,*n*], For $i \in$($1 \le i \le$), and chooses a random element $v_i \in$. Then sends the message *chal*= $\{(i, v_i)\}$ to the CSS.

$Genproof(F, \phi, chal) \rightarrow DV$

Upon receiving the challenge, the CSS computes

$$\sigma = \prod_{i=1}^{c} \epsilon , \qquad \delta = \prod_{i=1}^{c} \sigma_{i,2}^{H_2(m_i}, \qquad \eta = \prod_{i=1}^{c} \epsilon$$

The CSS outputs *pf*={ , , }and sends *pf* to the verifier as the response.

$VerfyP\,\mathrm{r}oof(X, y, pf, chal) \rightarrow \{true, false\}$

Upon receiving the response *pf* from the CSS, the designated verifier checks whether the following formula holds.

$$\sigma = (\delta\eta)^y$$

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

3.4 Security Analysis

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

Theorem 1: If Client and CSS are honest and follow the proposed procedures, then any challenge-response can pass verifier's checking, i.e., DV-PDP satisfies the correctness.

Proof: According to our scheme procedures, we know that

$$\begin{aligned}
\sigma &= \prod_{i=1}^{c} \epsilon \\
&= \prod_{i=1}^{c} (Y^{(H_2(m_i)k_{i1}+k_{i2})x} \\
&= \prod_{i=1}^{c} (g^{(H_2(m_i)k_{i1}+k_{i2})xy} \\
&= \prod_{i=1}^{c} (X^{(H_2(m_i)k_{i1}+k_{i2})y} \\
&= \prod_{i=1}^{c} (X^{H_2(m_i)k_{i1}y})^{v_i} \prod_{i=1}^{c} (X^{k_{i2}y} \\
&= \prod_{i=1}^{c} (\sigma_{i,2}{}^{H_2(m_i)y})^{v_i} \prod_{i=1}^{c} (\sigma_{i,2}{}^{y} \\
&= (\delta\eta
\end{aligned}$$

Theorem 2: If a (t',)-algorithm *A*, operated by an adversary, can generate a forgery tag under our DV-PDP scheme after making at most hash queries, at most tag queries and requesting setup, then there exists a ( ) –algorithm *B* that can solve the CDH problem in with $t \le t' + q_{H_1} T_G + q_T$ and $\varepsilon \ge \varepsilon' / q_K$, where one exponentiation on takes time .

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

4. Conclusions

We propose the designated verifier provable data possession model that provides authorized verification on remote data. DV-PDP enables a designated trusted third party to check data integrity under data owner's permission. Moreover the DV-PDP is more efficient because expense pairing is not calculated.

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 Shiuan-Tzuo Shen, Wen-Guey Tzeng.” Delegable Provable Data Possession for Remote Data in the Clouds” ICICS 2011, LNCS 7043, pp. 93–111, 2011.
15 HuaQun Wang. Proxy Provable Data Possession in Public Clouds. IEEE TRANSACTIONS ON SERVICES COMPUTING, Volume: PP , Issue: 99,2012.