Comment on id-based remote data integrity checking with data privacy preserving

Jianhong Zhang[12], Hongxin Meng[1]

[1]College of sciences, North China University of Technology,beijing 100144, China

[2]Guangxi Key Lab of Multi-source Information Mining & Security, Guilin China

Abstract. Recently, an ID-based remote data integrity checking protocol with perfect data privacy preserving (IEEE Transactions on Information Forensics and Security, doi: 10.1109/ TIFS.2016.2615853) was proposed to achieve data privacy protection and integrity checking. Unfortunately, in this letter, we demonstrate that their protocol is insecure. An active hacker can modify the stored data without being detected by the verifier in the auditing. And we also show malicious cloud server can convince the verifier that the stored data are kept intact after the outsourced data blocks are deleted. Finally, the reasons to produce such attacks are given.

1 Introduction

As a promising computing model, Cloud computing can provide convenient and on-demand network access service, storage service and resource sharing service for clients. Storage service is one of most popular cloud services, it allows that data users move its local data into the cloud and makes data users can flexibly access these data from the cloud servers.It provides data users with enormous storage space to outsource their data in an economical and scalable manner. Thus, a large number of individuals and organizations are tending to outsource their data storage to professional cloud services providers (CSP).

Compared with the traditional storage technology, cloud storage has a lot of advantages, however, as a new cutting edge service, it faces many new security issues since data user no longer possesses their data locally. For data user, the most concern issue is whether their data are deleted or corrupted. What is worse, cloud storage providers may deliberately delete rarely accessed data files which belong to an ordinary user to save money and storage space [11]. However, it claims that the data are still correctly stored in the cloud yet. Therefore, it is urgent and significant to design efficient public auditing protocols to strengthen trust and confidence of data owners to cloud storage service. To ensure the integrity of the outsourced data, the user needs to periodically check data integrity in order to be convinced that the data are correctly stored in the cloud. For the users, the biggest challenge is how to perform periodical integrity checking without the local copy of data files. And it is impractical to download the whole data file to check data integrity for a source-constrained data user.

In order to solve the problem of data integrity checking, many solutions have been proposed under the different systems and security models in [4], [7], [8], [9], [10], [12], [14]. However, most of existing solutions are mainly based on public key infrastructure (PKI) system. It is well known that PKI-based auditing system exists key management problem, data users need to manage its public key certificate. Thus, for a source-constrained cloud user, the key management of PKI-based data integrity checking scheme might become a difficult problem. Meanwhile it also spends more storage space than identity information to locally keep its public private key pairs. For an auditor, to audit data integrity, it needs to first retrieve the certificate of public key from CA, then to check the validity of public key's certificate. It will bring heavy burden to the auditor in terms of computation cost and communication cost.

The choice for ID-based cryptography (IBC) [2] is motivated by several reasons. (1) we benefit from an easier key management mechanism due to the certificate-free feature of IBC. That is, the public keys of a data user do not require the deployment of a Public Key Infrastructure (PKI) and the distribution of certificates. (2) IBC allows data user to obtain public keys without the corresponding private keys. That is, contrary to traditional public key derivation schemes, IBC does not require to compute the private key before producing the public key. Indeed, data users can directly use ID-based public keys to encrypt data before storage at no extra cost of communication. (3) IBC permits to data user to use the same ID-based public key under the different PKG, That is, a ID-based public key corresponds to multiple private keys. Thus, it alleviates data user's storage burden to public keys. (4) IBC does not need to obtain public key certificate from certificate authority (CA) and verify the validity of public key certificate, it saves computation cost and communication overhead.

Contributions: Recently, to realize data privacy protection in the auditing phase, Yu et al. proposed an ID-based remote data integrity checking protocol in [11]. In this letter, we show that their protocol [11] suffers from the hacker attack and malicious cloud server attack. Namely, the hacker/ malicious cloud server can alter/delete the out sourced data block, however, the verifier is fooled to trust that the stored data in cloud are well maintained.

2 System model and security requirement

In this section, we first present the system model of ID-based auditing protocol for cloud storage, then we define the corresponding's security model.

2.1 System model

For an ID-based auditing system for cloud data storage, its network system architecture is illustrated in Fig.1. The system involves four entities: data users, the cloud server, the third-party auditor and private key generator (PKG). Their roles are identified as follows:

- Data user: it is an entity which has a large amount of data files to be outsourced to the cloud storage for data maintenance and computation. In general, it is a resource-constrained entity
- The Cloud Server: it is an entity which has unlimited storage space and computation capability. And it is responsible for storing and maintaining the outsourced data and can provide the data access to the data user.
- The auditor: it is a trusted third-party which has expertise and capabilities to provide data auditing service on behalf of data users with cloud servers.
- Private key Generator: it is responsible to set up the whole system parameter and issue private key for each data users.

Cloud storage paradigm is to let the data users upload the large data files to the cloud servers in order to relieve of the burden of storage and computation of data users. However, it results in a potential problem: data user no longer possesses their data locally. Thus, it is of very importance for the data user to ensure that their data are being correctly stored and maintained. That is a reason why data users should be equipped with certain security measures so that they can periodically verify the integrity of the outsourced data even without the existence of local copies.

Definition 1. (ID-based auditing Protocol) An ID-based auditing protocol for cloud storage consists of the following algorithms.

1) Setup(1^k)→(params,mpk,msk) The algorithm takes a security parameter k as input and outputs system public parameters params, the master public-secret key pair (mpk; msk) of PKG.

2) KeyExtract(1^k; params; mpk; msk; ID) → (sk_{ID}).

The algorithm takes a security parameter *k*, system parameters *params*, the PKG's secret key msk , the user's identity information ID and a random element which is chosen by the user as inputs, and outputs the private key sk_{ID} corresponding to the user with identity ID.

3) TagGen(M,sk_{ID})→δ. The tag generation algorithm takes an outsourced data file M and the private key sk_{ID} as inputs, for each data block mi, it computes a data authentication tag δi. Finally, it outputs a set of data authentication tag $\delta = (\delta_1,\delta_2,\ldots, \delta n)$.

4）Integrity-Challenging(M_{info}) → C. For executing integrity checking, the challenging algorithm takes the abstract information of the data M_{info} (e.g., data file name, total number of data blocks, the challenged index subsect, etc.) as input and outputs a challenge information C.

5) Proof(M; δ;C) → P. To produce integrity proof information, the algorithm takes the data file M, the authentication tags δi, and the challenge information C from the auditor as inputs, and outputs a valid proof information P.

6) Verifying(C; P; mpk;Minfo)→1. The algorithm takes the proof information P , the public key mpk of PKG, and the abstract information of the data M_{info} as inputs, and outputs the auditing result as 0 or 1.

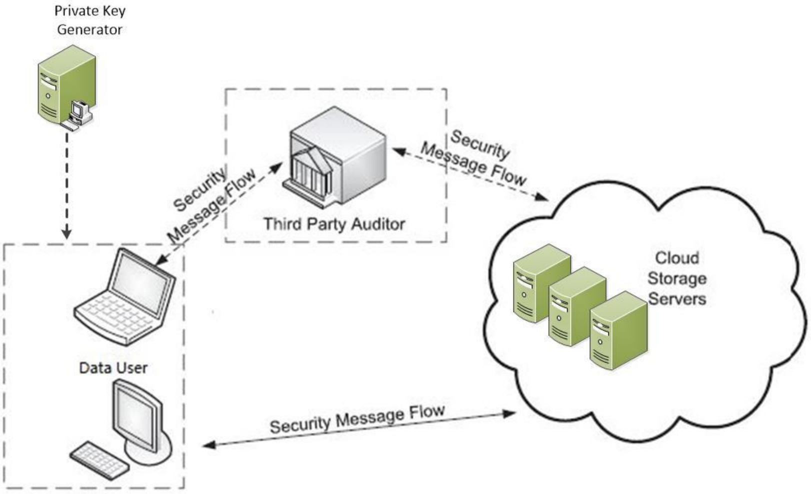

Fig1. System model of ID-based data auditing

2.2 Security requirement

For a third auditor, it is regarded as a honest-but-curious entity, that is to say, it can perform honestly the auditing, but might be curious about the stored data. Furthermore, the cloud server is considered as an untrusted entity. It may hide the fact of some data being deleted or corrupted for self-interest. Thus, for a cloud server, it may launch the following attacks to successfully cheat the third party auditor.

(1) Forge attack. During the auditing procedure, cloud server may forge a proof information P or an authentication tag of a data block to deceive the auditor.

(2) Replacing attack. If the challenged data blocks were corrupted, to pass the verification, cloud server may replace the corresponding pair of data block and data tag (m_j,δj) by choosing a valid uncorrupted pair of data block and data tag (m_i,δi).

(3) Reply attack. To energetically get through the auditing verification, cloud server may produce a proof information P by using the previous proof information or other former information, under the condition of not retrieving the challenged data of data user.

3 Reviews of Yu et al.'s ID-based auditing scheme

In their scheme, five algorithms are included. In the following, we will briefly review these algorithms. Please the interested readers refer to [11] for the details..

Setup. The system parameters are built as follows: G_1 and G_2 are two multiplicative cyclic groups of large prime order *p*, g_1 is a generator of group G_1. $e : G_1 \times G_1 \rightarrow G_2$ is an admissible bilinear pairing.

H_1 : $\{0,1\}^* \rightarrow G_1$,$H_2$: $\{0,1\}^* \rightarrow G_1$, and H_3 : $G_2 \rightarrow \{0,1\}^l$ are three hash functions. The KGC randomly chooses $\alpha \in Zp$ as its master secret key and computes public key $P_{pub} = g_1^{\alpha}$.

Key Extract. For an user's identity $ID \in \{0,1\}^*$, the KGC inputs its master secret key α to compute the private key of this user as $s = H_1(ID)$.

TagGen. For an outsourced file M, it is divided into n blocks, namely, $M = m_1||\ldots||m_n$. Then data owner randomly selects $\eta \in Z_p$ to compute $r = g_1^{\eta}$. For $i = 1$ to authentication tag of each block m_i is computed as

$$\sigma_i = s^{mi} \cdot H_2(fname||i)^{\eta}$$

Finally, the data owner uploads the file M together with (r, $\{\sigma_i\}$, $IDS(r||fname)$) to the cloud, where IDS ($r||fname$) denotes an ID-based signature on the value $r||fname$ which is from the data owner.

Challenge. The verifier randomly selects a c-element challenge subset I of the set [1, n]. And it also chooses random number $v_i \in Z_p$ for $i \in I$. Let $Q = \{i, v_i\}_{i \in I}$. To produce a challenge, the verifier chooses a random number $\rho \in Z_p$ to perform the procedures below:

1) compute $Z = e(H_1(ID), P_{pub})$, $c_1 = g_1^{\rho}$ and $c_2 = Z^{\rho}$;
2) produce a knowledge proof: $pf = POK\{\rho: c_1 = g_1^{\rho} \wedge c_2 = Z^{\rho}\}$

Finally, the challenge information $chall = (c_1, c_2, Q, prf)$ is sent to cloud server.

Proof. On receiving the challenge information $chall$, cloud server first computes $Z = e(H_1(ID), P_{pub})$ and verifies whether proof pf is valid. If it is invalid, the auditing is aborted. Otherwise, cloud server computes $\mu = \sum_{i \in I} v_i m_i$, $\sigma = \prod_{i \in I} \sigma_i^{vi}$ and $m' = H_3(e(\sigma, c_1) \cdot c_2^{-\mu})$. Finally, it sets $Prf = (m', r, IDS(r||fname))$ as proof information and responses Prf to the verifier.

Verify. After receiving proof information $Prf = (m', r, IDS(r||fname))$ is valid signature of data owner. If it not, the proof is invalid. Otherwise, it verifies whether the following equation holds

$$m' = h_2(\prod_{i \in I} e(H_2(fname||i)^{vi}, r^{\rho}))$$

If the equation above holds, the verifier accepts the proof.

4 Security analysis on protocol

In remote integrity checking, cloud server is a malicious entity. For his own benefits, it may corrupt the outsourced data. Furthermore, stronger adversaries may exist in the real life. For example, it is a malicious programmer or a hacker who plants bugs in the software and network protocols running on the cloud. In the following, we will show Yu et al.'s scheme [11] is insecure. Their scheme suffers hacker attack and malicious cloud server attack. The detail attacks are given as follows:

Attack 1: In this attack, we will show how a hacker attacks cloud server.

1) Assume that (M, r, $\{\sigma_i\}_{i \in [1, n]}$, $IDS(r||fname)$) is the uploaded information of the outsourced file M, where $M = m_1||\ldots||m_n$.
2) After attacking cloud server, a hacker alters the stored data by the following process. (a) it randomly chooses $k \in Z_p$ to alter M into $M' = m_1'||\ldots||m_n'$, where $m_i' = km_i$. And the corresponding authentication tag σ_i is altered into $\sigma_i' = \sigma_i^k$ for $i \in [1, n]$. (b) it intercepts the challenge information $chall = (c_1, c_2, Q, prf)$ and revises $Q = \{i, v_i\}$ into $Q' = \{ i, v_i '\}$ where $v_i' = k^{-1} \cdot v_i$. Then it sends $chall' = (c_1, c_2, Q', prf)$ to cloud server.
3) After cloud server receives the challenge information $chall'$, it honestly executes the protocol to compute $\mu' = \sum_{i \in I} v_i' m_i'$, $\sigma' = \prod_{i \in I} \sigma_i^{\prime vi'}$ and

$$\overline{m}' = H_3(e(\sigma', c_1) \cdot c_2^{-\mu})$$

Finally, proof information $Prf' = (\overline{m}', r, IDS(r||fname))$ is returned to the verifier.

In the following, we show that the proof information $Prf' = (\overline{m}', r, IDS(r||fname)$) can pass **Verify** algorithm since

$$\overline{m}' = H_3\left(e(\sigma', c_1) \cdot c_2^{-\mu'}\right)$$
$$= H_3((e(\prod_{i \in I} \sigma_i^{\prime v_i'}, c_1) \cdot c_2^{\sum_{i \in I} v_i' m_i'}))$$

$$= H_3((e(\prod_{i\in I} \sigma_i'^{v_i}, c_1) \cdot c_2^{\sum_{i\in I} v_i m_i}))$$
$$= H_3(e(\prod_{i\epsilon I} H_2\,(fname||i)^{v_i}, r^{\rho})$$

It is easy to see that by doing such a simple modification, the hacker can convince the verifier that the data in the cloud are well maintained, while the data have been corrupted.

Attack 2:

In such attack, we will show that the malicious cloud server can deceive the verifier that the outsourced files are intact after data blocks are deleted. The detail attack is given as follows:

1) Assume that (*M*, *r*, $\{\sigma_i\}_{i\in[1,n]}$, *IDS*(*r*||*fname*)) is the uploaded information of the outsourced files *M*.
2) To delete data block m_i of file *M*, the cloud works as follows:
   (a). It choose two data blocks m_j and m_l which are co-prime with m_i, namely, $a_1{\cdot}m_i + b_1{\cdot}m_j = 1$ and $a_2{\cdot}m_i + b_2{\cdot}m_j = 1$
   (b). Then it computes
   $$\sigma_i' = \frac{\sigma_i^{a_1}\cdot\sigma_j^{b}}{\sigma_i^{a_2}\cdot\sigma_l^{b_2}}$$
   $$= H_1\,(Fname||i)^{(a1-a2)\eta} \cdot \frac{H_1(Fname||j)^{b_1\eta}}{H_1(Fname||j)^{b_2\eta}}$$
3) After cloud server receives the challenge information *chall* = (c_1, c_2 ,*Q*, *prf*), assume that block *i* is included, it computes $\mu' = \sum_{i\in I/\{i\}} v_i m_i$, $\sigma' = (\sigma_i')^{v_i(a_1-a_2)^{-1}} \cdot \prod_{t\in I/\{i\}} \sigma_t^{v_t}$ and $\overline{m}' = H_3(e(\sigma', c_1) \cdot c_2^{-\mu} \cdot \frac{d_l^{v_i b_2/(a_1-a_2)}}{d_l^{v_i b_1/(a_1-a_2)}})$
   Where $d_j = e(\sigma_j, c_1) \cdot (c_2)^{-m_j}$ and $d_l = e(\sigma_l, c_1) \cdot (c_2)^{-m_j}$
4) Finally, the forged proof information *Prf'* = ($\overline{m}'$, *r*, *IDS*(*r*||*fname*)) is returned to the verifier.

It is easy to see that the returned proof information can pass verification since

$$e((\sigma_i')^{v_i(a_1-a_2)^{-1}}, c_1) \cdot \frac{d_l^{v_i b_2/(a_1-a_2)}}{d_j^{v_i b_1/(a_1-a_2)}}$$
$$= e(H_1(Fname||i)\frac{H(Fname||j)^{\frac{b_1}{a_1-a_2}}}{H_1(Fname||l)^{\frac{b_2}{a_1-a_2}}}, g_1^{\rho})^{v_j\eta} \cdot \frac{d_l^{\frac{v_i b_2}{a_1-a_2}}}{d_j^{\frac{v_i b_1}{a_1-a_2}}} = e(H_1(Fname||i), r^{\rho})^{v_i}$$

The reasons to produce the above attacks are that cloud server only returns a hash value of $e(\sigma, c_1){\cdot}c_2^{-\mu}$ in the proof information, and the integrity of Q in the challenge information can not been guaranteed. These factors result in some potential attack chances for malicious cloud server and the hackers.

5 Conclusion

In this letter, we analyse Yu et al.'s ID-based remote data integrity checking protocol and demonstrate that the malicious cloud server and the hacker can convince the verifier that the stored data files in cloud are pristine, while the data files have been corrupted.

Acknowledgements

This work was supported by Beijing Municipal Natural Science Foundation (No: 4162020,4132056) and Research Fund of Guangxi Key Lab of Multi-source Information Mining $\&$ Security (No:MIMS16-01).

References

[1] Yong Yu, Man Ho Au, Giuseppe Ateniese, Xinyi Huang, Willy Susilo, Yuan shun Dai, and Geyong Min, “Identity-based Remote Data Integrity Checking with Perfect Data Privacy Preserving for Cloud Storage”, IEEE Transactions on Information Forensics and Security, DOI10.1109/TIFS.2016.2615853.

[2] G. Ateniese, R. C. Burns, R. Curtmola, J. Herring, L. Kissner, Z. N. J. Peterson, and D. X. Song, “Provable data possession at untrusted stores”, in Proc. of ACM Conference on Computer and Communications Security 2007: 598-609, 2007.

[3] Yong Yu, Yafang Zhang, Yi Mu, Willy Susilo, Hongyu Liu, “Provably Secure Identity Based Provable Data Possession”, ProvSec 2015: 310- 325

[4] Zhang Jianhong, Dong Qiaocui, “Efficient ID-based Public Auditing for the Outsourced Data in Cloud Storage”, Information Sciences, Vol. 343-344, pp.1-14,2016.

[5] SG. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, “Provable Data Possession at Untrusted Stores”, Proc. 14th ACM Conf. Computer and Comm. Security (CCS07), pp. 598-609, 2007.

[6] G. Ateniese, S. Kamara, and J. Katz, “Proofs of Storage from Homomorphic Identification Protocols”, Proc. Intl Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology, pp. 319-333, 2009.

[7] Huaqun Wang, “Identity-Based Distributed Provable Data Possession in Multicloud Storage”, IEEE T. Services Computing, vol.8(2):328-340 ,2015

[8] E.J.Goh, S.Jarecki, “A signature scheme as secure as the diffie-hellman problem”, EUROCRYPT 2003. LNCS, vol. 2656, pp. 401-415.

[9] K.Ren, C. Wang, Q.Wang, “Security challenges for the public cloud”, IEEE Internet Computing, vol.16 (1): 69-73, 2012.

[10] H. Shacham and B. Waters, “Compact Proofs of Retrievability”, Proc. 14th Intl Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT’08), pp. 90-107, 2008.

[11] Huaqun Wang, Qianhong Wu, Bo Qin, Josep Domingo Ferrer, “Identity-based remote data possession checking in public clouds”,IET Information Security,vol.8(2), 2014: 114-121.