Comments and Corrections

Comments on "A Public Auditing Protocol With Novel Dynamic Structure for Cloud Data"

Xiong Li, Shanpeng Liu, and Rongxing Lu, *Senior Member, IEEE*

***Abstract*— In this paper, we discuss a security weakness of Shen *et al.*'s public auditing protocol for cloud data [IEEE Transactions on Information Forensics and Security, 12(10): 2402-2415, 2017.]. Specifically, we point out their protocol is vulnerable to a data privacy breach attack, *i.e.*, an adversary, once he compromises the third-party auditor latently, can also obtain all data owners' outsourced data by constructing appropriate challenges. As a result, it breaks the property of "privacy preserving". We hope that by identifying this design flaw, similar weaknesses can be avoided in future designs.**

***Index Terms*— Cloud storage, auditing protocol, privacy preserving, data breach.**

I. Introduction

PUBLIC auditing protocol has been regarded as an important mechanism to check the integrity of outsourced data stored in cloud service provider, and the "third-party based auditing", due to its advantages of heavily reducing the burden on data owners, has received considerable attention. However, recent reports show that the data breaches through third parties are getting more serious and awful today [1]. For example, according to the BDO and AusCERT 2018/19 Cyber Security Survey [2], data breaches experienced through third-party providers and suppliers rose by 74.3 percent in Australia. Therefore, the property of "privacy preserving" becomes essential for public auditing protocol, *i.e.*, the data owners' data should be still secure even if the third-party is malicious or compromised. Recently, Shen *et al.* [3] proposed a new public auditing protocol for cloud data, which assumes that the third-party auditor (TPA) is trustworthy for data owners (DOs). However, being trustworthy to the users does not mean that the TPA itself is reliable, and it may be compromised by adversaries. To the best of our knowledge, most related works do not assume that TPA is trustworthy in public auditing protocols. In this paper, we point out that Shen *et al.*'s protocol [3] is vulnerable to a data privacy breach attack, which breaks the property of "privacy preserving", *i.e.* an adversary, once he compromises the TPA, can also recover all data owners' outsourced data by constructing some suitable challenges.

Manuscript received September 18, 2019; revised February 21, 2020; accepted February 27, 2020. Date of publication March 5, 2020; date of current version March 27, 2020. This work was supported in part by the China Scholarship Council and in part by the Scientific Research Fund of Hunan Provincial Education Department under Grant 18A178. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Mauro Conti. *(Corresponding author: Xiong Li.)*

Xiong Li is with the Institute for Cyber Security, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China, and also with the Faculty of Computer Science, University of New Brunswick, Fredericton, NB E3B 5A3, Canada (e-mail: lixiongzhq@163.com).

Shanpeng Liu is with the School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan 411201, China (e-mail: liushanpeng0@gmail.com).

Rongxing Lu is with the Faculty of Computer Science, University of New Brunswick, Fredericton, NB E3B 5A3, Canada (e-mail: rlu1@unb.ca).

Digital Object Identifier 10.1109/TIFS.2020.2978592

II. Review of Shen *et al.*'s Protocol

In this section, we briefly review Shen *et al.*'s public auditing protocol [3] for cloud data, which contains three entities, *i.e.* the DO (data owner), the CSP (cloud service provider) and the TPA (third-party auditor), where TPA is a trustworthy third-party auditor between DO and CSP. Since the dynamic structure of their protocol is not relevant to our analysis, we omit it, and the detailed information about their dynamic structure can be found in [3].

For ease of description, some notations used in Shen *et al.*'s protocol [3] are introduced as follows. G and G_T are two multiplicative cyclic groups of large prime order p, and g and u are two generators of group G. $e : G \times G \rightarrow G_T$ is an efficiently computable bilinear pairing defined over G and G_T, and $h(\cdot) : \{0, 1\}^* \rightarrow G$ is a secure one-way hash function. m_i is the ith block of a data file F, and Loc_i is the specific location of m_i. Their protocol mainly composed of two phases, *i.e.* the setup phase and verification phase.

A. Setup Phase

This phase contains three algorithms as follows.

KeyGen: DO first selects a random key pair (ssk, spk) for signature, and randomly selects two generators g, u of group G. Then DO chooses a secret key $a \in Z_p^*$, and computes $v = g^a$. Finally, the KeyGen algorithm outputs the secret/public key pair $(sk, pk) = \{(a, ssk), (u, g, v, spk)\}$.

Filepro2C: DO divides the file F into n blocks $F = \{m_1, m_2, \ldots, m_n\}$. Then, DO generates a signature $\sigma_i = \left(h\left(V_i \| T_i\right) \cdot u^{m_i}\right)^a$ for each data block m_i $(i \in [1, n])$, where V_i represents the version of m_i, and T_i is the current timestamp. All the signatures form a set $\sigma = \{\sigma_i\}_{i \in [1,n]}$. To ensure the integrity of the file information, DO generates a tag $\vartheta = U_{ID} \| F_{ID} \| SIG\left(U_{ID} \| F_{ID}\right)_{ssk}$ by using ssk, where U_{ID} is the DO's identifier and F_{ID} is the file identifier. Finally, DO uploads $\{F, \sigma, \vartheta\}$ to CSP.

Filepro2T: DO runs this algorithm to store some related information to TPA for data auditing. In their protocol, the file and the data block information are stored in the doubly linked info table (DLIT), and the specific address of each data block is stored in the location array (LA). For more details of DLIT and LA, please refer to [3]. DO sends the parameters $\{F_{ID}, U_{ID}, V_i, T_i, Loc_i\}$ to TPA, where Loc_i is the specific address of the data block m_i. Upon receiving the parameters, TPA establishes the corresponding DLIT and LA to store the related information.

B. Verification Phase

This phase also contains three algorithms as follows.

ChalGen: TPA first asks CSP for the appropriate file tag ϑ and verifies the correctness of it by spk. Then, TPA selects s

random elements $a_{x,y}$ in the index table LA. Secondly, TPA gets the information of the corresponding data blocks in the DLIT according to the addresses $a_{x,y}$. Finally, TPA sends a challenge $chal = \{i, r_i\}_{i\in[1,s]}$ to CSP, where $s \in [1, n]$ and the random number $r_i \in Z_p$.

ProofGen: When receiving the challenge, CSP runs the ProofGen algorithm to generate a proof for the challenged blocks. The proof contains two parts, one is named the tag proof $T = \prod_{i\in[1,s]} \sigma_i^{r_i}$ and the other is named the data proof $D = \sum_{i\in[1,s]} m_i \cdot r_i$. Subsequently, CSP sends the proof (T, D) to TPA as a response to this challenge.

VerifyProof: After obtaining the proof from CSP, TPA runs the VerifyGen algorithm to determine whether the proof returned by CSP is reasonable, and the result shows whether the outsourced data is integrity. Firstly, TPA calculates $DI_i = e\left(h\left(V_i \| T_i\right), \upsilon\right)$ according to the challenged blocks based on the parameters stored in the DLIT. Secondly, TPA aggregates the data information $DI = \prod_{i\in[1,s]} DI_i$ for $i \in [1, s]$. Finally, TPA checks $e(T, g) \stackrel{?}{=} DI \cdot e\left(u^D, \upsilon\right)$. If the equality holds, it means that the outsourced data is integrity. Otherwise, the outsourced data is corrupted.

III. Correction and Security Analysis of Shen *et al.*'s Protocol

In this section, we first give a correction note on Shen *et al.*'s protocol [3], which is caused by the authors' typo or negligence. Next, we show their protocol is vulnerable to a data privacy breach attack if the TPA was compromised by an adversary.

A. Correction Note of Shen et al.'s Protocol

Due to the authors' typos or negligence [3], the equation (5) $DI_i = e\left(h\left(V_i \| T_i\right), \upsilon\right)$ in the verification phase and batch auditing phase of Shen *et al.*'s protocol [3] is not correct, and the correct equation would be $DI_i = e\left(h\left(V_i \| T_i\right)^{r_i}, \upsilon\right)$. Then the correctness analysis of their protocol in Section VI (A) should be revised as follows:

$$\begin{aligned}
e(T, g) &= e(\textstyle\prod_{i\in[1,s]} \sigma_i^{r_i}, g) \\
&= e(\textstyle\prod_{i\in[1,s]} (h(V_i \| T_i) \cdot u^{m_i})^{a\cdot r_i}, g) \\
&= e(\textstyle\prod_{i\in[1,s]} (h(V_i \| T_i)^{r_i} \cdot u^{m_i \cdot r_i}), g^a) \\
&= e(\textstyle\prod_{i\in[1,s]} (h(V_i \| T_i)^{r_i} \cdot u^{m_i \cdot r_i}), \upsilon) \\
&= e(\textstyle\prod_{i\in[1,s]} h(V_i \| T_i)^{r_i} \cdot u^{\sum_{i\in[1,s]} m_i \cdot r_i}, \upsilon) \\
&= e(\textstyle\prod_{i\in[1,s]} h(V_i \| T_i)^{r_i}, \upsilon) \cdot e(u^{\sum_{i\in[1,s]} m_i \cdot r_i}, \upsilon) \\
&= \textstyle\prod_{i\in[1,s]} DI_i \cdot e(u^D, \upsilon) \\
&= DI \cdot e(u^D, \upsilon)
\end{aligned}$$

Besides the above, the verification of both equations (11) and (12) in Section IV(A) of Shen *et al.*'s protocol [3] should be modified accordingly, and we here omit its description.

B. Data Privacy Breach Attack on Shen et al.'s Protocol

As reported in [1], [2], data breaches via third parties are a growing problem in today's society. Therefore, it is a big challenge for a TPA based public auditing protocol to protect the DOs' data privacy against the TPA, *i.e.* DOs' outsourced data cannot be recovered by the TPA from the proofs in any case. As we can see from the **ProofGen** algorithm of Shen *et al.*'s protocol [3], the data proof D is a linear combination of the challenged data blocks and its corresponding random numbers, and D is also transmitted via the public channel. Based on above observation, we found that an adversary, once he compromises the TPA, can also recover all DOs' outsourced data by constructing appropriate challenges. As a result, Shen *et al.*'s protocol [3] is vulnerable to a data breach attack and cannot achieve the property of "privacy preserving".

To facilitate the description of this attack, we assume that an adversary $\mathscr{A}$ has compromised the TPA and can act as the TPA. Take DO's file $F = \{m_1, m_2, \ldots, m_n\}$ as an example, $\mathscr{A}$ can recover the original outsourced data F by submitting n suitable challenges, and the data breach attack of their protocol is shown as follows.

1) $\mathscr{A}$ constructs a suitable invertible square matrix of order n:
$$A = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ r_{n1} & r_{n2} & \cdots & r_{nn} \end{bmatrix},$$
where each $r_{ij} (i, j \in [1, n])$ is a random element in Z_p, and each row of matrix A contains s non-zero elements. Then, $\mathscr{A}$ calculates its inverse matrix A^{-1} of modulo p.
2) For the ith ($i \in [1, n]$) row vector of matrix A, $\mathscr{A}$ extracts the column numbers of all s non-zero elements. Then $\mathscr{A}$ generates a challenge $chal_i = \{j, r_{ij}\}_{j\in[1,s]}$ using the s non-zero elements of row i, and forwards the challenge to CSP.
3) For each challenge $chal_i (i \in [1, n])$, CSP calculates the tag proof $T_i = \prod_{j\in[1,s]} \sigma_j^{r_{ij}}$ and data proof $D_i = \sum_{j\in[1,s]} m_j \cdot r_{ij} \mod p$ according to the **ProofGen** algorithm. Then, CSP sends the proof (T_i, D_i) back to $\mathscr{A}$.
4) When obtaining all n data proofs of the n challenges, $\mathscr{A}$ constructs a system of linear equations:
$$\begin{cases} \sum_{j\in[1,s]} m_j \cdot r_{1j} \mod p = D_1 \\ \sum_{j\in[1,s]} m_j \cdot r_{2j} \mod p = D_2 \\ \cdots \\ \sum_{j\in[1,s]} m_j \cdot r_{nj} \mod p = D_n \end{cases},$$
and it can be represented as:
$$A \cdot \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_n \end{bmatrix} \mod p = \begin{bmatrix} D_1 \\ D_2 \\ \vdots \\ D_n \end{bmatrix}.$$
Next, $\mathscr{A}$ can recover all data blocks $m_i (i \in [1, n])$ by calculating
$$\begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_n \end{bmatrix} = A^{-1} \cdot \begin{bmatrix} D_1 \\ D_2 \\ \vdots \\ D_n \end{bmatrix} \mod p.$$

Based on the above analysis, we can see that $\mathscr{A}$ has recovered DO's outsourced file $F = \{m_1, m_2, \ldots, m_n\}$ by constructing n suitable challenges, and Shen *et al.*'s protocol [3] is vulnerable to a data privacy breach attack. In addition, in order to make the attack look like normal audit challenge behavior, $\mathscr{A}$ can generate some normal random challenges, and then mix the n constructed challenges with these random challenges to form a larger challenge set. After receiving the corresponding proofs from the CSP, $\mathscr{A}$ picks up the

proofs corresponding to the n constructed challenges, and then the DO's outsourced file can be recovered.

IV. Conclusion

Data breaches via third parties has been reportedly common in terms of the current security situation [1], [2]. Therefore, even if a third-party auditor (TPA) is assumed trustworthy, we should still ensure TPA cannot read the outsourced data. Follow this line, in this paper, we have examined Shen *et al.*'s public auditing protocol for cloud data [3]. Concretely, we have showed that their protocol is vulnerable to a data breach attack, which breaks the property of "privacy preserving", *i.e.*, an adversary, once he compromises the TPA latently, can also recover all DOs' outsourced data by constructing suitable challenges. We hope that by identifying this design flaw, similar weaknesses can be avoided in future designs for resilient to third-party data breaches.

References

[1] L. Nate. *The Third Party Data Breach Problem*. Accessed: Jul. 27, 2017. [Online]. Available: https://digitalguardian.com/blog/third-party-data-breach-problem

[2] L. Fouche. *The BDO and AusCERT 2018/19 Cyber Security Survey*. Accessed: May 1, 2019. [Online]. Available: https://www.bdo.com/insights/business-financial-advisory/global/2018-2019-cyber-security-survey-results

[3] J. Shen, J. Shen, X. Chen, X. Huang, and W. Susilo, "An efficient public auditing protocol with novel dynamic structure for cloud data," *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 10, pp. 2402–2415, Oct. 2017.