Security Analysis of a Certificateless Public Provable Data Possession Scheme with Privacy Preserving for Cloud-Based Smart Grid Data Management System

Caixue Zhou

(Corresponding author: Caixue Zhou)

School of Information Science and Technology, Jiujiang University
551 Qianjin Donglu, Jiujiang 332005, China
(Email: charlesjjjx@126.com)

(Received Jan. 18, 2019; Revised and Accepted Aug. 8, 2019; First Online Sept. 21, 2019)

Abstract

The certificateless public key cryptosystem not only reduces the high cost of public key management, but also eliminates the private key escrow problem. The cloud-based smart grid data management system can release the burden of big data storage in power enterprises. Provable data possession (PDP) can ensure the integrity of data stored in the cloud with a high probability. Recently, a certificateless public PDP scheme with privacy preserving for cloud-based smart grid data management system was proposed. However, we find the scheme insecure. We give two concrete attacks to the scheme - the first attack shows that a malicious cloud storage provider (CSP) can forge a valid tag of any file block modified at his will, and the second one shows that CSP can produce a valid proof without storing any file blocks. Then, we point out the flaws in their proof and the key reason why their scheme is insecure.

Keywords: Certificateless Cryptosystem; Provable Data Possession; Smart Grid

1 Introduction

With the rapid development of computer network, communication technology and sensor technology, the smart grid [1, 6, 11] is gradually entering people's life as the next-generation power system. Build on the integrated and high-speed two-way communication work, it is aimed to achieve reliable, safe, economic, efficient and environmental-friendly operations. So far, many countries have launched smart grid projects [13, 16].

However, with the application of smart meters and other smart devices, the volume of electric power data is increasing exponentially. As a result, the traditional electric power information management system can no longer be able to process them in real time, prompting the birth of the cloud-based smart grid data management system [3]. It is flexible, scalable and reliable with a high equipment utilization rate and can help the smart grid achieve the storage of massive data. However, data stored in the cloud may be lost or damaged due to soft/hardware failures, human errors or hacker attacks. Thus, it has become an essential step to verify the integrity of data stored in the cloud.

Provable data possession (PDP) [5] can help check the integrity of cloud data without downloading it. It is a lightweight cloud data integrity probabilistic checking model. There are two kinds of auditing methods in PDP, i.e., public auditing [9] and private auditing [15]. In the former, Anyone with public information can audit the data, and therefore the cloud user can delegate the verification process to a third-party verifier (TPV) to ensure that his data is intact in the cloud; while in the latter, the auditor must use some private information to audit the data. At present, public auditing is becoming a popular trend. But in this method, the TPV should not deduce the cloud user's data when they check the integrity of it, and in this case, a public verifiable scheme with privacy preserving can be used [14]. However, all the above schemes are based on the public key infrastructure (PKI), which has the complex public key management problem. In order to reduce the high cost of public key management, the identity-based PDP [12] was proposed. However, the identity-based public key cryptosystem brings a new problem about key escrow - the trusted third party knows all users' private keys.

Regarding this problem, the certificateless public key cryptosystem [10] has great superiority. It not only reduces the high cost of public key management, but also solves the private key escrow problem. Many certifictateless PDP schemes have been proposed in the literature.

For example, in 2017, Kang *et al.* [7] proposed a certificateless public PDP scheme with privacy preserving, and applied it to the cloud-assisted wireless body area network. In the same year, He *et al.* [2] proposed another certificateless public PDP scheme with privacy preserving. Kim *et al.* [8] also proposed a certificateless public PDP scheme. In 2018, He *et al.* [4] proposed a certificateless public PDP scheme, and applied it to the cloud-assisted wireless body area network. However, their scheme does not support privacy preserving. In the same year, He *et al.* [3] proposed another certificateless public PDP scheme with privacy preserving, and applied it to the cloud-based smart grid data management system.

In this paper, we point out that scheme [3] is insecure and give two concrete attacks against it. The first attack shows that a malicious CSP can modify a file block and produce the corresponding tag. The second shows that a malicious CSP can produce a proof to pass the integrity verification without having to hold any data blocks.

The rest of this paper is organized as follows. Section 2 provides the formal definition and security model of certificateless provable data possession. Section 3 describes He *et al.*'s scheme. Section 4 gives two concrete attacks against their scheme, and then it points out the flaws in their proof and the key reason why their scheme is insecure. At last, the conclusion is given in Section 5.

2 Preliminaries

2.1 Formal Definition of Certificateless Provable Data Possession

There are four entities in the system: cloud users, who have huge data to be stored in the cloud; a cloud storage provider (CSP), which provides data storage service; a third party verifier (TPV), which is delegated by the cloud users to verify the cloud data integrity; and a key generation center (KGC), which produces system parameters and cloud users' partial private keys.

A certificateless provable data possession scheme consists of the following five algorithms:

1) Setup: Given a security parameter 1^k, KGC generates a master private key s and a common public parameter $Params$. For cloud user ID, KGC uses s and $Params$ to generate a partial private key PSK_{ID} and sends it to him secretly. Then, the cloud user ID randomly selects a secret value x_{ID}, and computes his public key UPK_{ID}. The cloud user ID's full private key consists of two parts: the partial private key PSK_{ID} and the secret value x_{ID}.

2) Store: Given a file $m_F = \{m_1, m_2, ..., m_n\}$, the cloud user uses his full private key to generate $\{m_i\}(i = 1, 2, ..., n)$'s tag $\{\sigma_i\}(i = 1, 2, ..., n)$. Then, he sends $\{m_i, \sigma_i\}(i = 1, 2, ..., n)$ to CSP, which checks whether $\{\sigma_i\}(i = 1, 2, ..., n)$ are valid. If they are invalid, CSP will ask the cloud user to re-produce them.

3) ChalGen: TPV randomly chooses a subset $I \in \{1, 2, ..., n\}$ and generates a challenge message to CSP.

4) ProGen: CSP produces a proof according to the challenge message, file $\{m_i\}(i = 1, 2, ..., n)$, tags $\{\sigma_i\}(i = 1, 2, ..., n)$ and sends it to TPV.

5) ProVer: TPV checks whether the proof is valid. If it is invalid, TPV will inform the cloud user that his file is corrupted.

Note 1. *Cloud users use the Store algorithm to produce file blocks' tags. With these tags and file blocks, CSP can produce a proof of data possession and cloud users can delete file blocks from their local copies. TPV uses the ChalGen algorithm to produce random file blocks to be audited. CSP uses the ProGen algorithm to produce a proof of data possession, which demonstrates that CSP stores users' files intactly. TPV uses the ProVer algorithm to check whether the proof is valid. If the proof is valid, it demonstrates that the data is intact in the cloud server.*

2.2 Security Model of Certificateless Provable Data Possession

Our security model is exactly the same as He *et al.*'s. There are two types of attackers in the certificateless public key cryptosystem [10]. The type-I attacker A_I can replace anyone's public key, but does not know the master private key. The type-II attacker A_{II} knows the master private key, but cannot replace anyone's public key. A certificateless PDP scheme must be unforgeable under both type-I and type-II adversaries.

Definition 1. *A CL-PDP scheme is unforgeable if no probabilistic polynomial time (PPT) adversary A (A_I or A_{II}) has a non-negligible advantage in the following game:*

Setup: Given a security parameter 1^k, challenger C produces the system's parameters $Params$ and a master private key s. If A is a type-I adversary, C gives the parameters $Params$ to A_I. If A is a type-II adversary, C gives the parameters $Params$ and master private key s to A_{II}.

Queries: A can adaptively make a polynomially bounded number of queries as follows:

1) Create-User Query: A supplies an identity ID. If ID's key pair has not been created, C produces ID's partial private key PSK_{ID} and secret value x_{ID}, and computes ID's public key UPK_{ID}. Then, C returns the public key UPK_{ID} to A.

2) Replace-Public-Key Query: A supplies an already created identity ID and a new public key UPK'_{ID}. C replaces the current public key UPK_{ID} with the new key UPK'_{ID}. If A is a type-II adversary, he cannot make such query.

3) Extract-Partial-Private-Key Query: A supplies an already created identity ID. C returns ID's partial private key PSK_{ID} to A. If A is a type-II adversary, he does not need to make such a query.
4) Extract-Secret-Value Query: A supplies an already created identity ID. C returns ID's secret value x_{ID} to A.
5) Tag-Gen Query: A supplies an already created identity ID and a file block m_i, and C computes the corresponding tag σ_i and returns it to A.

Forgery: At last, A outputs a forged tag σ^* corresponding the cloud user's identity ID^*. A wins the game if σ^* is valid and the following conditions hold.

1) If A is a type-I adversary, A_I cannot extract the partial private key of ID^*. If A is a type-II adversary, A_{II} cannot extract the secret value of ID^*.
2) σ^* is not the output of the Tag-Gen query.

3 He *et al.*'s Scheme

He *et al.*'s scheme consists of the following five algorithms.

Setup Algorithm

Step 1:

1) Given a security parameter k, KGC chooses two cyclic groups G_1 and G_2 of prime order q, a random generator P of G_1, a bilinear map $e : G_1 \times G_1 \to G_2$.
2) KGC randomly chooses $s \in Z_q^*$ as the system private key and computes the system public key $P_{pub} = sP$.
3) KGC chooses five secure hash functions $h_i : \{0,1\}^* \to Z_q^* (i = 1, 2, 3, 4)$ and $H : \{0,1\}^* \to G_1$.
4) KGC publishes the system parameters $\{G_1, G_2, e, P, q, P_{pub}, h_1, h_2, h_3, h_4, H\}$ and saves s secretly.

Step 2:

1) KGC randomly chooses $\overline{y}_{DO} \in Z_q^*$ and computes $\overline{Y}_{DO} = \overline{y}_{DO} \cdot P$.
2) KGC computes $\alpha_{DO} = h_1(ID_{DO}, \overline{Y}_{DO})$, and $y_{DO} = \alpha_{DO} \cdot \overline{y}_{DO} + s \mod q$.
3) KGC sends the partial private key y_{DO} to data owner (DO) secretly.

Step 3:

1) DO randomly chooses $x_{DO} \in Z_q^*$ as his secret value.
2) DO computes his public key $X_{DO} = x_{DO} \cdot P$.

Store Algorithm

Step 1:

1) DO randomly chooses $x_F \in Z_q^*$ and computes $X_F = x_F \cdot P$.
2) DO computes $\beta_{DO} = h_2(ID_{DO}, X_{DO}, \overline{Y}_{DO})$, $\alpha_F = h_3(ID_{DO}, X_{DO}, \overline{Y}_{DO}, X_F)$, and $s_F = \alpha_F \cdot x_F + \beta_{DO} \cdot x_{DO} + y_{DO} \mod q$.
3) DO saves x_F and X_F as a one-time signing key and verification key, respectively.

Step 2:

1) DO computes $V_{pub} = H(P_{pub})$ and $V_i = H(name_F, i)$ for $i = 1, ..., n$.
2) DO computes $\Phi_i = x_F \cdot (m_i \cdot V_{pub} + V_i)$ for $i = 1, ..., n$ and $\Phi_F = x_F \cdot V_{pub}$.
3) DO outputs Φ_i as m_i's tag for $i = 1, ..., n$.

Step 3: DO sends $\overline{F} = \{\{m_i\}_{i=1}^n, \{\Phi_i\}_{i=1}^n, s_F, X_F, \Phi_F\}$ to CSP.

Step 4: CSP checks if the equations $s_F \cdot P = \alpha_F \cdot X_F + \beta_{DO} \cdot X_{DO} + \alpha_{DO} \cdot \overline{Y}_{DO} + P_{pub}$ and $e(\sum_{i=1}^n \Phi_i, P) = e((\sum_{i=1}^n m_i) \cdot V_{pub} + \sum_{i=1}^n V_i, X_F)$ hold.

ChalGen Algorithm

Step 1: TPV randomly chooses a subset $I \in \{1, 2, ..., n\}$.

Step 2: TPV randomly chooses $w_i \in Z_q^*$ for each $i \in I$.

Step 3: TPV outputs $(\{i, w_i\}_{i \in I})$ as the challenge message and sends it to CSP.

ProGen Algorithm

Step 1: CSP randomly chooses $r_{CS} \in Z_q^*$ and computes $R_{CS} = r_{CS} \cdot \Phi_F$, $\Phi_{CS} = \sum_{i \in I} w_i \cdot \Phi_i$, $\alpha_{CS} = h_4(ID_{DO}, X_{DO}, \overline{Y}_{DO}, X_F, R_{CS}, \Phi_{CS})$ and $s_{CS} = \alpha_{CS} \cdot r_{CS} + \sum_{i \in I} w_i \cdot m_i \mod q$.

Step 2: CSP outputs the proof $\{X_F, \Phi_F, R_{CS}, \Phi_{CS}, s_{CS}\}$ and sends it to TPV.

ProVer Algorithm

TPV checks if the equations $s_F \cdot P = \alpha_F \cdot X_F + \beta_{DO} \cdot X_{DO} + \alpha_{DO} \cdot \overline{Y}_{DO} + P_{pub}$ and $e(\alpha_{CS} \cdot R_{CS} + \Phi_{CS}, P) = e(s_{CS} \cdot V_{pub} + \sum_{i \in I} w_i \cdot V_i, X_F)$ hold.

4 Security Analysis of He *et al.*'s Scheme

4.1 Two Concrete Attacks

Attack 1: Tag forging attack. According to Definition 1, in the Queries stage, a malicious CSP makes a Create-User query to ensure that the ID is created. Then he chooses a file block m_i and makes a Tag-Gen

query for (ID, m_i). After that, challenge C computes the corresponding tag $\overline{F} = \{m_i, \Phi_i, s_F, X_F, \Phi_F\}$ and returns it to him. Now, the malicious CSP can modify m_i to m_i^* and forge its corresponding tag Φ_i^* as follows. $\Phi_i^* = \Phi_i - m_i \cdot \Phi_F + m_i^* \cdot \Phi_F = x_F \cdot (m_i^* \cdot V_{pub} + V_i)$. The malicious CSP forges a valid tag Φ_i^* of m_i^* with a probability of 1.

Attack 2: Data loss hiding attack. Setup and Store algorithms are run as normal. After the Store algorithm, CSP gets file blocks and tags $\overline{F} = \{\{m_i\}_{i=1}^n, \{\Phi_i\}_{i=1}^n, s_F, X_F, \Phi_F\}$. Then, TPV runs the ChalGen algorithm to produce a challenge message $(\{i, w_i\}_{i \in I})$. After that, the malicious CSP computes $t_i = \Phi_i - m_i \cdot \Phi_F = x_F \cdot V_i$ for $i = 1, ..., n$. Then, he deletes all the file blocks $\{m_i\}_{i=1}^n$ and runs the ProGen algorithm as follows. He randomly chooses $r_{CS} \in Z_q^*$ and computes $R_{CS} = r_{CS} \cdot \Phi_F$, $\Phi_{CS} = \sum_{i \in I} w_i \cdot t_i$, $\alpha_{CS} = h_4(ID_{DO}, X_{DO}, \overline{Y}_{DO}, X_F, R_{CS}, \Phi_{CS})$, and $s_{CS} = \alpha_{CS} \cdot r_{CS} \mod q$. At last, the malicious CSP outputs the proof $\{X_F, \Phi_F, R_{CS}, \Phi_{CS}, s_{CS}\}$ and sends it to TPV. Obviously, the equation $e(\alpha_{CS} \cdot R_{CS} + \Phi_{CS}, P) = e(s_{CS} \cdot V_{pub} + \sum_{i \in I} w_i \cdot V_i, X_F)$ holds, meaning that the proof can pass the validation of the ProVer algorithm.

Note 2. *In the above attack 2, the malicious CSP can compute a valid proof without the cloud user's file, that is, the malicious CSP can delete all the cloud user's data file blocks.*

4.2 Flaws in the Proof of Lemma 1

In He *et al.*'s scheme, the proof of Lemma 1 is based on the security model which is defined in Subsection 2.2. In the Create-Data-Owner query, they divided it into two cases:

1) $ID_i = ID^*$. C stores $(ID^*, x_i, y_i, \perp, X_i, \overline{Y}_i)$ into L_K;

2) $ID_i \neq ID^*$. C stores $(ID_i, x_i, \perp, \overline{y}_i, X_i, \overline{Y}_i)$ into L_K.

Therefore, C knows the partial private key y_i in (1) and does not know the partial private key y_i in (2). Then, in the Extract-Partial-Private-Key query, C cannot give an answer when $ID_i \neq ID^*$. In other words, in most cases, C cannot answer this query. The same situation happens to Generate-Tag query - in most cases, C cannot answer this query, either.

In addition, in the proof part of C solving the CDH problem, the authors require $ID_i = ID^*$. In fact, our attack 1 shows that a malicious CSP can forge a valid tag when $ID_i \neq ID^*$. In other words, C can never solve the CDH problem.

Therefore, the simulation made by C is distinguishable from a true challenger, indicating that the proof is questionable.

4.3 Key Reason for the Insecurity

The key reason why He *et al.*'s scheme is insecure is that they compute a Φ_F in the Store algorithm. Then, the cloud user sends Φ_F to CSP along with the tags $\{\Phi_i\}_{i=1}^n$ and file blocks $\{m_i\}_{i=1}^n$. Because $\Phi_i = x_F \cdot (m_i \cdot V_{pub} + V_i) = m_i \cdot \Phi_F + x_F \cdot V_i$, CSP can modify file block m_i to m_i^* and compute $\Phi_i^* = \Phi_i - m_i \cdot \Phi_F + m_i^* \cdot \Phi_F = x_F \cdot V_i + m_i^* \cdot \Phi_F = x_F \cdot (m_i^* \cdot V_{pub} + V_i)$, that is, he can forge a valid tag in the above attack 1. At the same time, CSP can compute $t_i = \Phi_i - m_i \cdot \Phi_F = x_F \cdot V_i$ and produce a valid proof in the above attack 2. Obviously, if Φ_F is unknown to CSP, he cannot do the above attack computing.

Meanwhile, the ProGen algorithm must be run by CSP, but if Φ_F is unknown to CSP, it will not be able to run the algorithm. Therefore, He *et al.*'s scheme has a logic error.

5 Conclusions

In this paper, we give two attacks to a recently proposed certificateless public PDP scheme with privacy preserving for cloud-based smart grid data management system. In the first attack, a malicious CSP can forge a valid tag for any modified file block; and in the second one, a malicious CSP can produce a valid proof without storing any file blocks. We also point out the flaws in their proof and the key reason why their scheme is insecure.

Acknowledgments

This study was supported by the National Natural S cience Foundation of China [Grant no. 61462048] and the Natural Science Foundation of Jiangxi Province, China (No. 20181BAB202011). We thank to Ms. Yan Di, who checked our manuscript.

References

[1] C. Chrysoulas, "Shielding the grid world: An overview," *International Journal of Electronics and Information Engineering*, vol. 1, no. 1, pp. 23-28, 2014.

[2] D. B. He, N. Kumar, H. Q. Wang, L. N. Wang, and K. K. R. Choo, "Privacy-preserving certificateless provable data possession scheme for big data storage on cloud," *Applied Mathematic and Computation*, vol. 314, pp. 31–43, 2017.

[3] D. B. He, N. Kumar, S. Zeadally, and H. Q. Wang, "Certificateless provable data possession scheme for cloud-based smart grid data management systems," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 3, pp. 1232–1241, 2018.

[4] D. B. He, S. Zeadally, and L. B. Wu, "Certificateless public auditing scheme for cloud-assisted wireless

body area networks," *IEEE Systems Journal*, vol. 12, no. 1, pp. 64–73, 2018.

[5] W. F. Hsien, C. C. Yang, and M. S. Hwang, "A survey of public auditing for secure data storage in cloud computing," *International Journal of Network Security*, vol. 18, no. 1, pp. 133–142, 2016.

[6] M. Inam, Z. Li, A. Ali, and A. Zahoor, "A novel protocol for vehicle cluster formation and vehicle head selection in vehicular ad-hoc networks," *International Journal of Electronics and Information Engineering*, vol. 10, no. 2, pp. 103–119, 2019.

[7] B. Y. Kang, J. Q. Wang, and D. Y. Shao, "Certificateless public auditing with privacy preserving for cloud-assisted wireless body area networks," *Mobile Information Systems*, 2017. DOI: 10.1155/2017/2925465.

[8] D. M. Kim and I. R. Jeong, "Certificateless public auditing protocol with constant verification time," *Security and Communication Networks*, vol. 2017, no. 5, pp. 1-14, 2017.

[9] C. W. Liu, W. F. Hsien, C. C. Yang, and M. S. Hwang, "A survey of public auditing for shared data storage with user revocation in cloud computing," *International Journal of Network Security*, vol. 18, no. 4, pp. 650–666, 2016.

[10] L. H. Liu, W. P. Kong, Z. J. Cao, and J. B. Wang, "Analysis of one certificateless encryption for secure data sharing in public clouds," *International Journal of Electronics and Information Engineering*, vol. 6, no. 2, pp. 110–115, 2017.

[11] N. S. Nafi, K. Ahmed, M. A. Gregory, and M. Datta, "A survey of smart grid architectures, applications, benefits and standardization," *Journal of Network and Computer Applications*, vol. 76, pp. 23–36, 2016.

[12] S. Peng, F. C. Zhou, J. Li, Q. Wang, and Z. F. Xu, "Efficient, dynamic and identity-based remote data integrity checking for multiple replicas," *Journal of Network and Computer Applications*, vol. 134, pp. 72–88, 2019.

[13] R. Singh and M. S. Manu, "An energy efficient grid based static node deployment strategy for wireless sensor networks," *International Journal of Electronics and Information Engineering*, vol. 7, no. 1, pp. 32-40, 2017.

[14] S. Thokchom and D. K. Saikia, "Privacy preserving and public auditable integrity checking on dynamic cloud data," *International Journal of Network Security*, vol. 21, no. 2, pp. 221–229, 2019.

[15] T. Y. Wu, Y. M. Tseng, S. S. Huang, and Y. C. Lai, "Non-repudiable provable data possession scheme with designated verifier in cloud storage systems," *IEEE Access*, vol. 5, pp. 19333–19341, 2017.

[16] Y. Zhang, W. Chen, and W. J. Gao, "A survey on the development status and challenges of smart grids in main driver countries," *Renewable and Sustainable Energy Reviews*, vol. 79, pp. 137–147, 2017.

Biography

Caixue Zhou received BS degree in Computer Science Department from Fudan University in 1988, Shanghai, China and MS degree in Space College of Beijing University of Aeronautics and Astronautics in 1991, Beijing, China. He is an Associate Professor in the School of Information Science and Technology, Jiujiang University, Jiujiang, China since 2007. He is a member of the CCF (China Computer Federation) and a member of CACR (Chinese Association for Cryptologic Research). His research interests include applied cryptography, security of computer networks.