2021 15th International Conference on Ubiquitous Information Management and Communication (IMCOM) | 978-1-6654-2318-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/IMCOM51814.2021.9377422

Security Analysis of Blockchain-based User Authentication for Smart Grid Edge Computing Infrastructure

Hakjun Lee
Department of Electrical and Computer Engineering
Sungkyunkwan University
Suwon, Korea
hjlee@security.re.kr

Jihyeon Ryu
Department of Computer Science and Engineering
Sungkyunkwan University
Suwon, Korea
jhryu@security.re.kr

Youngsook Lee
Department of Cyber Security
Howon University
Gunsan, Korea
ysooklee@howon.ac.kr

Dongho Won*
Department of Computer Science and Engineering
Sungkyunkwan University
Suwon, Korea
dhwon@security.re.kr

***Abstract*—With the development of IT technology and the generalization of the Internet of Things, smart grid systems combining IoT for efficient power grid construction are being widely deployed. As a form of development for this, edge computing and blockchain technology are being combined with the smart grid. Wang et al. proposed a user authentication scheme to strengthen security in this environment. In this paper, we describe the scheme proposed by Wang et al. and security faults. The first is that it is vulnerable to a side-channel attack, an impersonation attack, and a key material change attack. In addition, their scheme does not guarantee the anonymity of a participant in the smart grid system.**

***Index Terms*—User authentication, Internet of Things, Smart grid, Edge computing, Blockchain**

I. Introduction

The smart grid maximizes energy efficiency through real-time information exchange related to power supply between suppliers and consumers by integrating information and communication technology into the power grid system. By connecting power grid devices to the Internet, intelligent demand management, connection to new and renewable energy, and electric vehicle charging have become possible in our lives [1].

With the development of IT technology, cloud edge computing has been applied to the smart grid system to improve the quality, reliability, and flexibility of energy transmission, enhance the efficiency of smart grid IoT device management, and reduce communication latency between nodes.

By utilizing the inherent characteristics of edge computing, it is possible to cover the heterogeneity, mobility, and geographical distribution of power grid units in an edge computing-based smart grid system.

In the smart grid system, mutual authentication is used to establish a session key between smart grid IoT devices, such as smart meters and units that aggregate total power usage. Using this session key, they can hide sensitive information such as users' identities and allow secure communication.

However, the public key-based certificate system is not suitable in a network environment based on IoT devices with limited resources. There are many computational and communication overheads in issuing, revoking, signing, and verifying certificates [2].

In this paper, we deal with the mutual authentication scheme in the smart grid based on edge computing applying the blockchain proposed by Wang et al. [3] Using blockchain, their scheme is to establish a mutually secure session key and protect end-user (EU)'s identity from being exposed to edge-server (ES). In addition, through smart contracts, their scheme improves efficiency in key management such as updating and discarding key materials. However, we found that their scheme still is not secure, and in this paper, we analyze the weaknesses of their scheme.

The section II introduces the related work. Section III describes the prior knowledge used in Wang et al.'s scheme and we review the scheme in section IV. We perform the security analysis of the scheme of Wang et al. in section V and finally conclude the paper with section VI.

II. Related Work

In the past few years, many authentication schemes in various network environments have been proposed [4]–[6]. In addition, the schemes for establishing session keys in a smart grid environment have been proposed.

Tsai et al. [7] proposed a key exchange scheme for a smart grid environment to provide anonymous access service between smart meters and electricity suppliers using bilinear pairings. However, the attacker can extract the secret key from the smart meter. Therefore, their scheme does not guarantee the privacy of the end-user of smart grid.

Wazied et al. [8] introduce a lightweight remote user authentication scheme. It supports the dynamic addition of users, password, and biometric updates.However, those schemes cannot provide the revocation of removing malicious or faulty smart meters from the network.

978-0-7381-0508-6/21/$31.00 ©2021 IEEE

Mahmood et al. [9] suggested a key exchange protocol to provide the user's anonymity for smart grid with edge computing infrastructure. However, Liang et al. [10] showed that the attacker can impersonate the user by ephemeral secret leakage attack in Mahmood et al.'s scheme [9].

Recently, Wang et al. [3] introduced a blockchain-based mutual authentication for smart grid systems with edge computing. To prevent the leakage of the secret key and enhance the efficiency of key management, they apply the blockchain to the authentication protocol. However, we found that their scheme is still insecure.

Therefore, we review the scheme of Wang et al. [3] and then perform the security analysis to discuss its weaknesses in this paper.

III. Background

A. Network Model

The smart grid network model introduced in this paper is composed of resister authority (RA), EU, ES, and blockchain.

1) RA: It is an electricity supplier and it is a trusted party to all smart grid system participants. RA has the authority to manage participants using smart contracts for user identification, key update, and revocation using blockchain.
2) EU: As a smart meter, EU measures the user's energy consumption and reports the information to the ES. In general, EU is geographically linked to the closest ES.
3) ES: It not only acts as an aggregator to collect information but also acts as a controller that controls the ES. In addition, it communicates with the central cloud server to store and process data, and it is connected to the blockchain network to perform additional tasks for authentication of end nodes.
4) Blockchain: It is responsible for recording the public key material in the smart contract in this paper. The information recorded in the blockchain is used to issue, update, and destroy keys to users, and these tasks are performed through smart contracts.

B. Smart Contract

In Wang et al.'s scheme [3], smart contracts are used to manage the key materials table. It is used to provide anonymity to communication participants and to support efficient termination without asynchronous problems. To do this, Algorithms 1-4 are used in smart contracts.

IV. Review of the Target Scheme

A. System Setup

In this step, RA performs the following process to create its own secret value and public parameters used throughout the system.

RA selects a cyclic additive group $\mathbb{G}$ with generator P and prime order q on an elliptic curve $E(\mathbb{F}_q)$ over the finite field $\mathbb{F}_q$. RA sets the two secure one-way hash function $h_1 : \{0,1\}^* \rightarrow \mathbb{Z}_q$ and $h_2 : \{0,1\}^* \rightarrow \{0,1\}^{2\kappa}$, where $\kappa = log_2 q$ is the security parameter. RA picks the private key $s \leftarrow \mathbb{Z}_q^*$, and computes the public key $P_{pub} = s \cdot P$. Finally, RA securely keeps s and releases the public parameters $(\mathbb{G}, P, q, h_1, h_2, P_{pub})$.

Algorithm 1: KMT_Initialization

```
contract KMT {
  address owner
  struct KMTS {
    byte32 PKH;
    uint256[2] RP;
    uint256[4] EID;
    DataTime ET;}
  KMTS[] public KMT;
  constructor KMT() {
    owner = msg.sender;
    len = 0;
    return 1;
}
```

Fig. 1. Algorithm to initialize key material in smart contracts.

Algorithm 2: KMT_Update

```
function updateKMT (oldPKH, PKH, RP, EID, ET){
  if owner ≠ msg.sender then
    return 0;
  else {
  Exist(KMTS[i].PKH == oldPKH) then {
      KMTS[i].PKH = PKH;
      KMTS[i].RP = RP;
      KMTS[i].EID = EID;
      KMTS[i].ET = ET;
      return 1; }
    else{
      len++;
      KMTS[len].PKH = PKH;
      KMTS[i].RP = RP;
      KMTS[len].EID = EID;
      KMTS[len].ET = ET;
    return 1; }
}
```

Fig. 2. Algorithm to update Key material in smart contracts.

B. Registration Phase

In the registration stage, EU_i and ES_j register with RA through the secure channel and record public and secret parameters used for authentication in the blockchain. In this paper, only the EU_i registration step is described as an example, and

Algorithm 3: KMT_Query

```
function queryKMT (PKH) {
  if Exist(KMTS[i].PKH == PKH) then
    return KMTS;
  else;
    return 0;
}
```

Fig. 3. Algorithm to query key material in smart contract.

Algorithm 4: KMT_Revoke

```
function revoke KMT (PKH) {
  if owner ≠ msg.sender then
    return 0;
  else {
    if Exist(KMTS[i].PKH == PKH) then {
      Release(KMTS[i]);
      for; i < len; i++
        KMTS[i] = KMTS[i+1];
      len−−;
      return 1; }
    else
      return 0; }
}
```

Fig. 4. Algorithm to revoke key in smart contracts.

ES_j registration step is omitted because it is performed in the same way.

1) First, EU_i transmits its ID_i with the registration request to RA, and RA checks whether EU_i is already registered. If so, it stops the registration phase. Otherwise, RA selects a random number $r_i \in \mathbb{Z}_{||}^*$and calculates $RP_i = r_i \cdot P$, $sck = r_i + s \cdot h_1(ID_i||RP_i)$ and $PK_i = sck_i \cdot P$.
2) RA computes $PKH_i = h_1(PK_i)$ and $EID_i = Enc_{P_{pub}}(ID_i)$, and sets an expiration time ET_i. RA posts $(PKH_i, EID_i, RP_i, ET_i)$ on the blockchain using Algorithm 2, i.e., updateKMT$(Null, PKH_i, EID_i, RP_i, ET_i)$. Finally, RA sends the secret parameters sck_i and PK_i to EU_i.
3) EU_i computes $PKH_i = h_1(PK_i)$ and obtains $(PKH_i, EID_i, RP_i, ET_i)$ by invoking Algorithm 3, i.e., KMTQuery(PKH_i). Then, EU_i checks whether $sck_i \cdot P = RP_i + h_1(ID_i||RP_i) \cdot P_{pub} = PK_i$ is valid. If it holds, EU_i stores the secret key sck_i. Otherwise, EU_i restarts or terminates the registration phase.

C. Authentication Phase

In the authentication step, registered EU_i and ES_j verify each other's identities using the blockchain system in order to construct a session key for communication with each other. The detailed procedure is as follows.

1) EU_i selects a random number $a \in \mathbb{Z}_q^*$, and computes $AP_i = a \cdot P$, $pid_i = PK_i \cdot h_2(AP_i||a \cdot PK_j)$ and $k = a + sck_i \cdot h_1(PK_i||pid_i||AP_i||TS_1)$, where TS_1 is the current timestamp. Then, EU_i sends (AP_i, pid_i, k, TS_1) to ES_j.
2) After receiving the message from EU_i, ES_j checks the freshness of TS_1. If it is fresh, ES_j computes $PK_i'a = pid_i \oplus h_2(AP_i||sck_k \cdot AP_i)$ and PKH_i‘ $= h(PK_i')$. ES_j invokes queryKMT(PKH_i') to check the validation of EU_i's identity. If it is valid, ES_j verifies whether $kP = AP_i + h_1(PK_i'||pid_i||AP_i||TS_1)$. If it holds, ES_j selects a random number $b \in \mathbb{Z}_{||}^*$ and computes $BP_i = b \cdot P$, $K_1 = sck_j \cdot AP_i + b \cdot PK_i$, $K_2 = b \cdot AP_i$, $SK_{ji} = h_1(PK_i'||ID_j||K_1||K_2)$, and $SV_j = h_1(SK_{ji}||K_1||K_2||TS_2)$. Finally, ES_j sends (BP_i, SV_j, TS_2) to EU_i.
3) EU_i check the freshness of TS_2. If it is fresh, EU_i computes $K_3 = a \cdot PK_j + sck_i \cdot BP_i$, $K_4 = a \cdot BP_i$, and $SK_{ij} = h_1(PK_i||ID_j||K_3||K_4)$. Finally, EU_i check $h_1(SK_{ij}||K_3||K_4||TS_2)$ is equal to SV_j. If they are same, EU_i and ES_j share same session key.

D. Update Phase

At this phase, EU_i calls updateKMT$(oldPKH_i, PKH_i, RP_i, EID_i, ET_i)$ to generate new key materials when the expiration time is reached or the secret key is comprised. Through this, PKH_i, RP_i and ET_i are newly set.

E. Revocation Phase

At this stage, revokeKMT(PKH_i) is invoked when RA detects an abnormal behavior of EU_i or when EU_i needs to be removed or left from the smart grid system.

V. Security Analysis of Wang et al.'s Scheme

In this section, we perform the security analysis of Wang et al.'s scheme [3]. We have identified three vulnerabilities. The details are as follows:

A. Side-channel attack

Among the various IoT security considerations, side-channel attacks are one major issue that must be considered. The attacker recovers the secret key stored in the device by analyzing sub-channel information such as the amount of computation time, power consumption, electromagnetic field radiation, and calculation result value injected with errors.

In the registration stage of Wang et al.'s scheme, ES_i and EU_i stores the secret value sck_i sent to the RA, in the memory. However, they do not encrypt or mask sck_i. That is, sck_i can be easily extracted by side-channel attack and the attacker can perform an impersonation attack.

B. Impersonation attack

After the side-channel attack, the attacker can begin the following process to impersonate a legitimate EU_i by using the extracted secret value sck_i in the authentication:

- The attacker generates a random number $a_{\mathcal{A}} \in \mathbb{Z}_q^*$, and computes $AP_{\mathcal{A}} = a \cdot P$, $pid_{\mathcal{A}} = PK_{\mathcal{A}} \cdot h_2(AP_{\mathcal{A}}||a_{\mathcal{A}} \cdot PK_j)$ and $k_{\mathcal{A}} = a + sck_i \cdot h_1(PK_{\mathcal{A}}||pid_{\mathcal{A}}||AP_{\mathcal{A}}||TS_{\mathcal{A}})$, Then, the attacker sends $(AP_{\mathcal{A}}, pid_{\mathcal{A}}, k_{\mathcal{A}}, TS_{\mathcal{A}})$ to ES_j.

$AP_{\mathcal{A}}, pid_{\mathcal{A}}, k_{\mathcal{A}}$ are parameters created from sck_i that is extracted from the user's smart meter by the attacker, so they are also valid in the subsequent authentication step. In other words, the attacker can establish the session key with ES_j and finally succeeds in disguised as EU_i.

C. Key material change attack

In the authentication phase, EU_i uses ES_j's identity ID_j to calculate the session key $SK_{ij} = h_1(PK_i||ID_j||K_3||K_4)$. However, before calculating this, it can be assumed that ID_j is public information because EU_i does not have any process to derive ID_j from the transmitted parameters. In other words, the attacker can easily obtain the victim's real identity ID_j, use it to pass the owner check procedure of updateKMT algorithm. maliciously manipulate the victim's key materials.

D. Anonymity

In communication exchanging sensitive information, it is important to hide the identity of the communication participants. As mentioned in the previous subsection, ES_j's identity is public. Therefore, Wang et al.'s scheme [3] does not guarantee anonymity.

VI. Conclusion

With the advent of cloud edge computing technology, Wang et al. recently introduced the blockchain-based authentication protocol to protect user privacy in a smart grid combined with edge computing. However, we have found that this protocol is vulnerable to the side-channel attack, the impersonation attack, the key material manipulation attack, and it does not guarantee anonymity. We have described these security flaws. In future work, we need to design an improved and anonymous authentication scheme.

Acknowledgment

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2019R1A2C1010159)

References

[1] G. Dileep, "A survey on smart grid technologies and applications," *Renewable Energy*, vol. 146, pp. 2589–2625, 2020.

[2] P. Tedeschi, S. Sciancalepore, A. Eliyan, and R. Di Pietro, "Like: Lightweight certificateless key agreement for secure iot communications," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 621–638, 2019.

[3] J. Wang, L. Wu, K.-K. R. Choo, and D. He, "Blockchain-based anonymous authentication with key management for smart grid edge computing infrastructure," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 1984–1992, 2019.

[4] M. Kim, J. Moon, D. Won, and N. Park, "Revisit of password-authenticated key exchange protocol for healthcare support wireless communication," *Electronics*, vol. 9, no. 5, p. 733, 2020.

[5] J. Ryu, H. Lee, H. Kim, and D. Won, "Secure and efficient three-factor protocol for wireless sensor networks," *Sensors*, vol. 18, no. 12, p. 4481, 2018.

[6] S. Roy, S. Chatterjee, A. K. Das, S. Chattopadhyay, N. Kumar, and A. V. Vasilakos, "On the design of provably secure lightweight remote user authentication scheme for mobile cloud computing services," *IEEE Access*, vol. 5, pp. 25 808–25 825, 2017.

[7] J.-L. Tsai and N.-W. Lo, "Secure anonymous key distribution scheme for smart grid," *IEEE transactions on smart grid*, vol. 7, no. 2, pp. 906–914, 2015.

[8] M. Wazid, A. K. Das, N. Kumar, and J. J. Rodrigues, "Secure three-factor user authentication scheme for renewable-energy-based smart grid environment," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 3144–3153, 2017.

[9] K. Mahmood, X. Li, S. A. Chaudhry, H. Naqvi, S. Kumari, A. K. Sangaiah, and J. J. Rodrigues, "Pairing based anonymous and secure key agreement protocol for smart grid edge computing infrastructure," *Future Generation Computer Systems*, vol. 88, pp. 491–500, 2018.

[10] X.-C. Liang, T.-Y. Wu, Y.-Q. Lee, C.-M. Chen, and J.-H. Yeh, "Cryptanalysis of a pairing-based anonymous key agreement scheme for smart grid," in *Advances in Intelligent Information Hiding and Multimedia Signal Processing*. Springer, 2020, pp. 125–131.