Cryptoanalysis of Lightweight and anonymous three-factor authentication and access control protocol for real-time applications in wireless sensor networks

Jihyeon Ryu[1], Youngsook Lee[2], and Dongho Won[3],⋆

[1]Department of Platform Software, Sungkyunkwan University
jhryu@security.re.kr
[2]Cyber Security, Howon University,
ysooklee@howon.ac.kr
[3]Department of Computer Engineering, Sungkyunkwan University
dhwon@security.re.kr

Abstract. Wireless sensor networks are used to monitor various environmental conditions such as temperature, sound, pollution level, humidity, wind and so on. Therefore, fast and secure authentication is important and requires a lightweight protocol that is secure. AmirHosein et al. proposed a 3-factor protocol in wireless sensor networks, but we found that the protocol in the wireless sensor network had some weaknesses. Firstly, it is a weakness in smart card deodorization. And it is also vulnerable to user impersonation. Moreover, it is possible to attack the session key. In this paper, we describe this weakness and prove AmirHosein et al.'s scheme is insecure.

Keywords: Remote user authentication · Wireless sensor network · Biometric

1 Introduction

Wireless sensor networks (WSNs) represent the communication between wireless sensors with various sensors and are one of the core technologies used in modern IoT. WSNs use many industrial and consumer applications such as temperature monitoring, environmental conditions like temperature, sound, pollution level, humidity, wind and so on.

The WSN consists of the following three elements: (1) User interface (2) Gateway node (GW) (3) Sensor node (SN). The user interface provides the user with an environment to access the GW and SN. The GW enables communication between the user U and the sensor node SN, and the SN measures the

⋆ Corresponding author.

R. Alfred et al. (eds.), *Computational Science and Technology*,
Lecture Notes in Electrical Engineering 603,
https://doi.org/10.1007/978-981-15-0058-9_33

physical environmental condition. WSN should provide users with fast speeds and simple protocols, and of course safety must also be satisfied. Accordingly, various types of user authentication paper have recently been proposed in the WSN [2–4].

Wong et al [6]. proposed a 2-factor user authentication scheme in a lightweight, dynamic hash-based WSN for the first time in 2006. However, this technique has problems such as Replay attack and forgery attack, so in 2007, Tseng et al. proposed a new authentication scheme that complements it [7]. However, this also has problems such as Replay attack and MITM attack, and Vaidya et al proposed a robust dynamic user authentication scheme [8]. This has the problem of DoS and forgery attack [9]. There have also been many user authentication papers on wireless sensor networks [10–14]. The Das model [15] was also a frequently cited 2-factor user authentication in WSN. However, the scheme of Das proposed a scheme of He et al [16]. and Khan et al [17]. due to the lack of mutual authentication and key exchange and vulnerability to impersonation attack. However, Kumar et al [18]. found that [16] is vulnerable to information leaking attacks, does not guarantee user anonymity, and [17] does not provide mutual authentication and does not guarantee the confidentiality of their messages.

In 2016, Gope et al [5]. propose a novel two-factor lightweight anonymous authentication protocol in WSN that uses a database to overcome the previous vulnerabilities. However, AmirHosein et al [1]. argue that their protocol is vulnerable to side-channel attacks because they are 2-factors, and session keys are also vulnerable. To overcome these drawbacks, in 2019, AmirHosein et al. proposed a new 3-factor authentication protocol in WSN. We confirmed that the scheme of AmirHosein et al. is still vulnerable.

The rest of the paper is summarized as follows. We provide some preliminary knowledge such as hash function, fuzzy extractor in section 2. In Section 3, we review AmirHosein et al.'s protocol of [1]. Moreover, we specify some vulnerabilities in AmirHosein et al.'s protocol [1] in section 4. At last, the conclusion is shown in Section 5.

2 Preliminary Knowledge

This section describes the basic knowledge of the hash function and contents of the fuzzy extractor used in AmirHosein et al.'s scheme [1].

2.1 Hash function

A hash function maps data of arbitrary length to fixed length data, and is useful for fast data retrieval and fast encryption. The hash function has the following three properties [19].

preimage-resistance When there is an output, it is computationally infeasible to find the input that hashes it. i.e. to find any preimage $x^{'}$ such that $h\,(x^{'}) = y$ when given any y for which a corresponding input is not known.

2nd-preimage resistance It is computationally infeasible to find another input with the same output for a particular input. i.e. to find a 2nd-preimage $x^{'} \neq x$ such that $h\ (x) = h\ (x^{'})$.

collision resistance It is computationally impossible to find two inputs with the same hashing result. i.e. any two distinct inputs x, $x^{'}$ which hash to the same output. such that $h\ (x) = h\ (x^{'})$

2.2 Fuzzy Extractor

Handling the user's biometric information should be very careful and accurate. However, the biometric information may not be recognized exactly the same. The fuzzy extractor uses error tolerances to solve this. Based on [20], the fuzzy extractor works in the following:

$$Gen(B) \rightarrow \langle x, y \rangle \tag{1}$$

$$Rep(B^*, y) = x \text{ if } B^* \text{ is information similar to B} \tag{2}$$

B represents biometric information of the user, and B^* represents information similar to biometric information of the user. *Gen* is a probabilistic algorithm using biometric input B , and extracts string $x \in \{0,1\}^k$ and assistance string $y \in \{0,1\}^*$. *Rep* is a deterministic algorithm that recovers α from y and any vector B^* that is similar to B.

3 Review of the target protocol

This section describes AmirHosein et als's protocol [1]. The scheme consists of three phases as follows: registration, login, authentication, and password change. The notation for the target paper [1] is written in Table 1.

3.1 Registration Phase

In the registration phase, the user and gateway nodes in the private channel exchange secret information about the smart card. This allows confidential information to be stored in the database used by the smart card and gateway nodes when the user authenticates.

1. User U chooses his/her identity U_{id} and sends the registration request U_{id} and Personal credential to the gateway node GW in the secure channel.

Table 1. Notations used in AmirHosein et al. protocol.

Notations	Description
U	The user
GW	Gateway node
SN	Sensor node
$\mathcal{A}$	The malicious attacker
U_{id}	Identity of user
U_{psw}	Password of user
U_b	Biometric information of user
AU_{id}	Disposable identity of user
SU_{id}	Shadow identity of user
GW_{id}	Identity of gateway node
SN_{id}	Identity of the sensor node
SC	Smart cad
DB	Database
w	Private key of gateway node
APM	A set of user U's access privilege masks
G	A set of user U's group ids
KEM_{ug}	Secret emergency key between user and gateway
Sk_{ug}	Secret key between user and gateway
Sk_{gs}	Secret key between gateway and sensor node
SK	Session key between user and sensor node
Ts_{ug}	Transaction sequence values
$h\ (\cdot)$	One-way hash function
$X \parallel Y$	Concatenate operation
$\oplus$	Bitwise XOR operation

2. The gateway node GW generates random number n_g, unique random number used to identify a particular access group G_j, random number user access privilege mask APM_j and random sequence number Ts_{ug}. After that group the created variables $G = \{G_1, G_2, ...\}$, $APM = \{APM_1, APM_2, ...\}$. After obtaining the registration request from user U, GW calculates $Sk_{ug} = h\ (U_{id} \parallel n_g) \oplus GW_{id}$, $sid_j = h(U_{id} \parallel r_j \parallel Sk_{ug})$, $SU_{id} = \{sid_1, sid_2, ...\}$, $KEM_{ug_j} = h\ (U_{id} \parallel sid_j \parallel r'_j)$, $G = \{G_1, G_2, ...\}$ and $APM = \{APM_1, APM_2, ...\}$. Also GW computes $U^{\#}_{id} = U_{id} \oplus h\ (GW_{id} \parallel w \parallel Ts_{ug})$, $Sk^{\#}_{ug} = Sk_{ug} \oplus h\ (GW_{id} \parallel U_{id} \parallel w)$, $G^{\#}_j = G_j \oplus h\ (GW_{id} \parallel U_{id} \parallel w)$, $APM^{\#}_j = APM_j \oplus h\ (GW_{id} \parallel U_{id} \parallel w)$, $Sk^{\#}_{gs} = Sk_{gs} \oplus h\ (GW_{id} \parallel w \parallel SN_{id})$ and $KEM^{\#}_{ug} = KEM_{ug} \oplus h\ (GW_{id} \parallel U_{id} \parallel w)$ using its secret key w. And save the data $\langle Ts_{ug}, (SU_{id}, KEM^{\#}_{ug}), Sk^{\#}_{ug}, Sk^{\#}_{gs}, U^{\#}_{id}, G^{\#}, APM^{\#} \rangle$ in $DB.GW$ sends $\langle Sk_{ug}, (SU_{id}, KEM_{ug}), Ts_{ug}, G_U, h\ (\cdot) \rangle$ to user U in SC.
3. After user U takes SC from the GW, chooses his/her U_{id}, password U_{psw}, imprints the biometric U_b and then computes $Gen(U_b) = (RS_U, P_U)$, $Sk^{*}_{ug} = Sk_{ug} \oplus h\ (h\ (U_{id}) \oplus h\ (U_{psw}) \oplus h\ (RS_U))$, $KEM^{*}_{ug} = KEM_{ug} \oplus h\ (h\ (U_{id}) \oplus h\ (U_{psw}) \oplus h\ (RS_U))$, $SU^{*}_{id} = SU_{id} \oplus h\ (h\ (U_{id}) \oplus h\ (U_{psw}) \oplus h\ (RS_U))$,

$G^* = G \oplus h\ (h\ (U_{id}) \oplus h\ (U_{psw}) \oplus h\ (RS_U))$, $f_U^* = h\ (h\ (Sk_{ug}) \oplus h\ (U_{id}) \oplus h\ (U_{psw}) \oplus h\ (RS_U))$. And save the data $\langle Sk_{ug}^*, f_U^*, (SU_{id}^*, KEM_{ug}^*), Tsug, G^*, P_U, Gen\ (\cdot), Rep\ (\cdot), h\ (\cdot)\rangle$ in SC.

3.2 Login Phase

The user enters his/her confidential information into the smart card and requests login in the login phase.

1. U inserts the smart card and enters U_{id}, U_{psw} and U_b. The smart card computes $RS_U = Rep\ (U_b, P_U)$, $Sk_{ug} = Sk_{ug}^* \oplus h\ (h\ (U_{id}) \oplus h\ (U_{psw}) \oplus h\ (RS_U))$ and checks the condition $f_U = h\ (h\ (Sk_{ug}) \oplus h\ (U_{psw}) \oplus h\ (U_{id}) \oplus h\ (RS_U)) \stackrel{?}{=} f_U^*$. If it holds, the smart card ensures that the user successfully passes the verification process. Otherwise, this phase terminates immediately.
2. After verification successfully, user U generates random number N_u and computes $N_x = Sk_{ug} \oplus N_u$, $G = G^* \oplus h\ (h\ (U_{id}) \oplus h\ (U_{psw}) \oplus h\ (RS_U))$, $AU_{id} = h\ (U_{id} \parallel Sk_{ug} \parallel N_u \parallel Ts_{ug})$, $G_j^{'} = G_j \oplus N_u$, $V_1 = h\ (AU_{id} \parallel G_j^{'} \parallel Sk_{ug} \parallel N_x \parallel SN_{id})$. In case of loss of synchronization, user U selects one of the unused pair of $(sid_j^*, KEM_{ug_j}^*)$ from (SU_{id}^*, KEM_{ug}^*) and surrender his/her U_{id}, U_{psw}, RS_U and computes $sid_j = sid_j^* \oplus h\ (h\ (U_{id}) \oplus h\ (U_{psw}) \oplus h\ (RS_U))$, $KEM_{ug} = KEM_{ug}^* \oplus h\ (h\ (U_{id}) \oplus h\ (U_{psw}) \oplus h\ (RS_U))$, $AU_{id} = sid_j$ and $Sk_{ug} = KEM_{ug_j}$.
3. U sends the login request messages $M_{A_1} = \{AU_{id}, G_j^{'}, N_x, Ts_{ug}(ifreq), SN_{id}, V_1\}$ to GW.

3.3 Authentication Phase

In the authentication phase, the gateway node verifies the user with the login message received from the user, and sends a new message containing the secret information to the sensor node. The sensor node and user share their keys and exchange secret information.

1. After receiving the login request messages M_{A_1} from user U, the GW first checks the validity of the transaction sequence number Ts_{ug}. GW computes $N_u = Sk_{ug} \oplus N_x$ and $G_j = G_j^{'} \oplus N_u$, and also computes $h\ (GW_{id} \parallel U_{id} \parallel w) = G_j^{\#} \oplus G_j$, $APM_j = APM_j^{\#} \oplus h\ (GW_{id} \parallel U_{id} \parallel w)$ that $G_j^{\#}$ and $APM_j^{\#}$ are in DB. After that, GW calculates $AU_{id} = h\ (U_{id} \parallel Sk_{ug} \parallel N_U \parallel Ts_{ug})$, $V_1 = h\ (AU_{id} \parallel G_j^{'} \parallel Sk_{ug} \parallel N_x \parallel SN_{id})$ and checks if AU_{id} and V_1 is valid. If successfully verification of AU_{id} then continue calculates. Otherwise, GW terminates the session. GW generates a session key SK and time stamp T, calculates $SK^{'} = h\ (Sk_{gs}) \oplus SK$, $APM_j^{'} = h\ Sk_{gs} \oplus APM_j$ and $V_2 = h\ (AU_{id} \parallel APM_j^{'} \parallel SK^{'} \parallel T \parallel Sk_{gs})$. Finally, GW sends the messages $M_{A_2} = \{AU_{id}, APM_j^{'}, SK^{'}, T, V_2\}$ to the sensor node SN.

2. Upon getting the message M_{A_2}, SN checks T whether it is valid or not. If this does not hold, SN disconnects the session. Then SN also verifies $V_2 \stackrel{?}{=} h\ (AU_{id} \parallel APM_j^{'} \parallel SK^{'} \parallel T \parallel Sk_{gs})$. If it does not satisfy, SN disconnects also. SN computes $APM_j = APM_j^{'} \oplus h\ (Sk_{gs})$ and generates new time stamp $T^{'}$. SN continues to calculate $SK = h\ (Sk_{gs}) \oplus SK^{'}$, $V_3 = h\ (SK \parallel Sk_{gs} \parallel SN_{id} \parallel T^{'})$, $K_{gs_{new}} = h\ (Sk_{gs} \parallel SN_{id})$ and $Sk_{gs} = K_{gs_{new}}$. At last, SN transmits $M_{A_3} = \{T^{'}, SN_{id}, V_3\}$ to GW.
3. The gateway node GW checks the time stamp T' and $V_3 \stackrel{?}{=} h\ (SK \parallel Sk_{gs} \parallel SN_{id} \parallel T^{'})$. If not, it terminates the connection. Gw generates a random number $Ts_{ug_{new}}$ and calculates $Ts = h\ (Sk_{ug} \parallel U_{id} \parallel N_U), SK^{"} = h\ (Sk_{ug} \parallel U_{id} \parallel N_U) \oplus SK$, $V_4 = h\ (SK^{"} \parallel N_U \parallel Ts \parallel Sk_{ug})$, $K_{ug_{new}} = h\ (Sk_{ug} \parallel U_{id} \parallel Ts_{ug_{new}})$, $Sk_{ug} = K_{ug_{new}}$, $K_{gs_{new}} = h(\ Sk_{gs} \parallel SN_{id})$ and updates $Sk_{ug} = K_{ug_{new}}$ and $Sk_{gs} = K_{gs_{new}}$. If GW cannot get Ts_{ug} in M_{A_1}, GW generates a random number $K_{ug_{new}}$ and calculates $x = h\ (U_{id} \parallel KEM_{ug_j}) \oplus K_{ug_{new}}$. And then, GW updates $Sk_{ug} = K_{ug_{new}}$ after that sends the messages $M_{A_4} = \{SK^{"}, Ts, V_4, x\}$ to the user U.
4. After user U obtains the message $V_4 = h\ (SK^{"} \parallel N_U \parallel Ts \parallel Sk_{ug})$ checks its validity. If there is no abnormality, proceed to the next step or end it. And U computes $SK = h\ Sk_{ug} \parallel U_{id} \parallel N_U) \oplus SK^{"}$, $Ts_{ug_{new}} = h\ (Sk_{ug} \parallel U_{id} \parallel N_U) \oplus Ts$, $K_{ug_{new}} = h\ (Sk_{ug} \parallel U_{id} \parallel Ts_{ug_{new}}$ and then updates $Sk_{ug} = K_{ug_{new}}$ and $Ts_{ug} = Ts_{ug_{new}}$.
5. U and SN successfully shared SK. The SN responds user U's query according to APM_j stored for user U using session key SK. Finally, at the end of this phase, the SN removes APM_j from storage due to security reasons.

3.4 Password and Biometrics Change Phase

Follow the steps below to change the user's password:

1. U puts his/her smart card into the terminal and inserts U_{id}, previous password U_{psw} and previous biometric U_b. And inputs the new password U_{psw}^*, new biometric U_b^*.
2. Smart card computes $RS_U = Rep\ (U_b, P_U)$ and retrieve Sk_{ug},KEM_{ug}, SU_{id}, G and f_U as follows. $Sk_{ug} = Sk_{ug}^* \oplus h\ (h\ (U_{id}) \oplus h\ (U_{psw}) \oplus h\ (RS_U))$, $KM_{ug} = KEM_{ug}^* \oplus h\ (h\ (U_{id}) \oplus h\ (U_{psw}) \oplus h\ (RS_U))$, $SU_{id} = SU_{id}^* \oplus h\ (h\ (U_{id}) \oplus h\ (U_{psw}) \oplus h\ (RS_U))$, $G = G \oplus h\ (h\ (U_{id}) \oplus h\ (U_{psw}) \oplus h\ (RS_U))$ and $f_U = f_U^* \oplus h\ (h\ (Sk_{ug}) \oplus h\ (U_{psw}) \oplus h\ (U_{id}) \oplus h\ (RS_U))$.
3. Smart card computes $Gen\ (U_b^*)$, Sk_{ug}^{**}, SU_{id}^{**}, KEM_{ug}^{**}, G^{**} and f_U^{**} as bellow. $Gen\ (U_b^*) = (RS_U^*, P_U^*)$, $Sk_{ug}^{**} = Sk_{ug} \oplus h\ (h\ (U_{id}) \oplus h\ (U_{psw}^*) \oplus h\ (RS_U^*))$, $SU_{id}^{**} = SU_{id} \oplus h\ (h\ (U_{id}) \oplus h\ (U_{psw}^*) \oplus h\ (RS_U^*))$, $KEM_{ug}^{**} = KEM_{ug} \oplus h\ (h\ (U_{id}) \oplus h\ (U_{psw}^*) \oplus h\ (RS_U^*))$, $G^{**} = G \oplus h\ (h\ (U_{id}) \oplus h\ (U_{psw}^*) \oplus h\ (RS_U^*))$, $f_U^{**} = h\ (h\ (Sk_{ug}) \oplus h\ (U_{psw}^*) \oplus h\ (U_{id}) \oplus h\ (RS_U^*))$.
4. Finally, smart card will replace Sk_{ug}^* with Sk_{ug}^{**}, SU_{id}^* with SU_{id}^{**}, KEM_{ug}^* with KEM_{ug}^{**}, G^* with G^{**}, f_U^* with f_U^{**} and P_U with P_U^* .

4 Analysis of the target protocol

We prove that AmirHosein et al.'s protocol [1] has some security exposure in this section. The details are as follows.

4.1 Loss of Smart card information

Attacker $\mathcal{A}$ can decrypt the information on the SC equally in the following two cases. First case is insider attack in the registration phase and the second case is loss of synchronize in the login phase. Insider attack is a stronger attack, but it can be considered when there is no case of loss of synchronize.

Insider attack In registration phase, Attacker $\mathcal{A}$ extract the smart card SC when GW sends to U. He/she can get the SC information $\{Sk_{ug}, SU_{id}, KEM_{ug}, Ts_{ug}, G, h(\cdot)\}$ that are not encrypted.

Loss of synchronize

1. An attacker $\mathcal{A}$ steals the U's smart card SC, the inside information is $\langle Sk^*_{ug}, f^*_u, (SU^*_{id}, KEM^*_{ug}), Tsug, G^*, P_U, Gen(\cdot), Rep(\cdot), h(\cdot)\rangle$.
2. And in loss of synchronize case, $\mathcal{A}$ can thus get user's login message $M_{A_1} = \{AU_{id}, G^{'}_j, N_x, Ts_{ug}(ifreq), SN_{id}, V_1\}$. $\mathcal{A}$ computes $h(h(U_{id}) \oplus h(U_{psw}) \oplus h(RS_U)) = AU_{id} \oplus SU^*_{id}$. The obtained information $h(h(U_{id}) \oplus h(U_{psw}) \oplus h(RS_U))$ can be calculated $Sk_{ug} = Sk^*_{ug} \oplus h(h(U_{id}) \oplus h(U_{psw}) \oplus h(RS_U))$, $KEM_{ug} = KEM^*_{ug} \oplus h(h(U_{id}) \oplus h(U_{psw}) \oplus h(RS_U))$, $G = G^* \oplus h(h(U_{id}) \oplus h(U_{psw}) \oplus h(RS_U))$.

4.2 User Impersonation Attack

An attacker $\mathcal{A}$ can make a user impersonation attack, and the victim is assumed to be U. The details are as follows.

1. $\mathcal{A}$ generates random numbers $N_{\mathcal{A}}$ and computes $N_{x\mathcal{A}} = Sk_{ug} \oplus N_{\mathcal{A}}$, $G^{'}_{j\mathcal{A}} = G_j \oplus N_{\mathcal{A}}$, $AID_{\mathcal{A}} = h(U_{id} \parallel Sk_{ug} \parallel N_{\mathcal{A}} \parallel Ts_{ug})$ and $V_{1\mathcal{A}} = h(AU_{id} \parallel G_{j\mathcal{A}} \parallel Sk_{ug} \parallel N_{\mathcal{A}} \parallel SN_{id})$ that Sk_{ug} and G_j obtained from stolen smart card attack.
2. $\mathcal{A}$ transmits the login request $M_{\mathcal{A}1} = \{AID_{\mathcal{A}}, G^{'}_{j\mathcal{A}}, N_{x\mathcal{A}}, Ts_{ug}, SN_{id}, V_{1\mathcal{A}}\}$ to the gateway node GW.
3. After GW obtains the login request from the $\mathcal{A}$, first, verifies Ts_{ug} and calculates $N_{\mathcal{A}} = Sk_{ug} \oplus N_{x\mathcal{A}}$, $G_j = G^{'}_{j\mathcal{A}} \oplus N_{\mathcal{A}}$ and $h(GW_{id} \parallel U_{id} \parallel w) = G^{\#}_j \oplus G_j$, $APM_j = APM^{\#}_j \oplus h(GW_{id} \parallel U_{id} \parallel w)$ that $G^{\#}_j$ and $APM^{\#}_j$ are in DB. And GW computes $AID_{\mathcal{A}} = h(U_{id} \parallel Sk_{ug} \parallel N_{\mathcal{A}} \parallel Ts_{ug})$, $V_{1\mathcal{A}} = h(AU_{id} \parallel G^{'}_{j\mathcal{A}} \parallel Sk_{ug} \parallel N_{\mathcal{A}} \parallel SN_{id})$ and checks if $AID_{\mathcal{A}}$ and V_1 is valid. GW does not detect attackers. Unfortunately, GW still misunderstand to communicate with U.

As a result, the attacker $\mathcal{A}$ will be verified as GW by user U. Therefore, the user impersonation attack is succeed.

4.3 Session Key Attack

Assume that Attacker $\mathcal{A}$ has access to the DB. At this time, Attacker $\mathcal{A}$ can extract the session key SK of user U and sensor node SN as follows.

1. Assume that the attacker $\mathcal{A}$ can access to the database $DB = \langle Ts_{ug}, (SU_{id}, KEM_{ug}^{\#}), Sk_{ug}^{\#}, Sk_{gs}^{\#}, U_{id}^{\#}, G^{\#}, APM^{\#} \rangle$. He/she will use the data $Sk_{ug}^{\#}$.
2. Attacker $\mathcal{A}$ calculates $h\ (GW_{id} \parallel U_{id} \parallel w) = Sk_{ug} \oplus Sk_{ug}^{\#}$, $APM_j = APM_j^{\#} \oplus h\ (GW_{id} \parallel U_{id} \parallel w)$ and extract the message $M_{\mathcal{A}_2} = \{AU_{id}, APM_j^{'}, SK^{'}, T, V_2\}$. And then, $\mathcal{A}$ computes $h\ (Sk_{gs}) = APM_j^{'} \oplus APM_j$ and $SK = h(Sk_{gs}) \oplus SK^{'}$. Now, attacker $\mathcal{A}$ successfully seized the session key SK.

As a result, this result shows that AmirHosein et al.'s protocol does not satisfy session key.

5 Conclusions

In this paper, we revisited AmirHosein et al.'s three-factor user authentication protocol for wireless sensor networks, and pointed out that insider attack and loss of synchronize attack are possible in AmirHosein et al.'s protocol. The stolen smart card attack could be used to extract critical user's information. Consequently, It enables attacks on escape of session key and user impersonation attack. For these reasons, their protocol cannot assure the security of authentication. Finally, our further research would propose an improved user authentication protocol which can handle with these problems.

Acknowledgements

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

References

1. Adavoudi-Jolfaei, A., Ashouri-Talouki, M., Aghili, S. F.: Lightweight and anonymous three-factor authentication and access control scheme for real-time applications in wireless sensor networks. Peer-to-Peer Networking and Applications, **12**(1), 43–59.(2019)
2. Gope, P., Das, A. K., Kumar, N., Cheng, Y.: Lightweight and Physically Secure Anonymous Mutual Authentication Protocol for Real-Time Data Access in Industrial Wireless Sensor Networks. IEEE Transactions on Industrial Informatics. (2019)
3. Li, X., Peng, J., Obaidat, M. S., Wu, F., Khan, M. K., Chen, C.: A Secure Three-Factor User Authentication Protocol With Forward Secrecy for Wireless Medical Sensor Network Systems. IEEE Systems Journal. (2019)

4. Rani, R., Kakkar, D., Kakkar, P., Raman, A.: Distance based enhanced threshold sensitive stable election routing protocol for heterogeneous wireless sensor network. In Computational Intelligence in Sensor Networks, Springer, Berlin, Heidelberg. 101–122. (2019)
5. Gope, P., Hwang, T.: A realistic lightweight anonymous authentication protocol for securing real-time application data access in wireless sensor networks. IEEE Transactions on industrial electronics, **63**(11), 7124–7132. (2016)
6. Wong, K. H., Zheng, Y., Cao, J., Wang, S.: A dynamic user authentication scheme for wireless sensor networks. In IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'06) **1**(8). (2006)
7. Tseng, H. R., Jan, R. H., Yang, W.:. An improved dynamic user authentication scheme for wireless sensor networks. In IEEE GLOBECOM 2007-IEEE Global Telecommunications Conference. 986–990. (2007)
8. Vaidya, B., S Silva, J., Rodrigues, J. J.: Robust dynamic user authentication scheme for wireless sensor networks. In Proceedings of the 5th ACM symposium on QoS and security for wireless and mobile networks. 88–91. (2009)
9. Faye, Y., Niang, I., Guyennet, H.: A user authentication-based probabilistic risk approach for Wireless Sensor Networks. In 2012 International Conference on Selected Topics in Mobile and Wireless Networking. 124–129. (2012)
10. Ryu, J., Lee, H., Kim, H., Won, D.: Secure and Efficient Three-Factor Protocol for Wireless Sensor Networks. Sensors, **18**(12), 4481. (2018)
11. Moon, J., Lee, D., Lee, Y., Won, D.: Improving biometric-based authentication schemes with smart card revocation/reissue for wireless sensor networks. Sensors, **17**(5), 940. (2017)
12. Jung, J., Moon, J., Lee, D., Won, D.: Efficient and security enhanced anonymous authentication with key agreement scheme in wireless sensor networks. Sensors, **17**(3), 644. (2017)
13. Chung, Y., Choi, S., Lee, Y., Park, N., Won, D.: An enhanced lightweight anonymous authentication scheme for a scalable localization roaming service in wireless sensor networks. Sensors, **16**(10), 1653. (2016)
14. Nam, J., Choo, K. K. R., Han, S., Kim, M., Paik, J., Won, D.: Efficient and anonymous two-factor user authentication in wireless sensor networks: achieving user anonymity with lightweight sensor computation. Plos one, **10**(4), e0116709. (2015)
15. Das, M. L.: Two-factor user authentication in wireless sensor networks. IEEE transactions on wireless communications, **8**(3), 1086–1090. (2009)
16. He, D., Gao, Y., Chan, S., Chen, C., Bu, J.: An Enhanced Two-factor User Authentication Scheme in Wireless Sensor Networks. Ad hoc and sensor wireless networks. **10**(4), 361–371. (2010)
17. Khan, M. K., Alghathbar, K.: Cryptanalysis and security improvements of 'two-factor user authentication in wireless sensor networks'. Sensors. **10**(3), 2450–2459. (2010)
18. Kumar, P., Lee, H. J.: Cryptanalysis on two user authentication protocols using smart card for wireless sensor networks. In 2011 Wireless Advanced. 241–245. IEEE. (2011)
19. Katz, J., Menezes, A. J., Van Oorschot, P. C., Vanstone, S. A.: Handbook of applied cryptography. CRC press. (1996)
20. Dodis, Y., Reyzin, L., Smith, A.: Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In International conference on the theory and applications of cryptographic techniques. 523–540. Springer, Berlin, Heidelberg. (2004)