Cryptanalysis of the Lightweight and Anonymous Authentication and Access Control for Real-time Applications in Wireless Sensor Networks*

Sooyeon Shin, Jongshin Kim, and Taekyoung Kwon†
Graduate School of Information, Yonsei University, Seoul, 03722, South Korea
{shinsy80, jongshin123, taekyoung}@yonsei.ac.kr

Abstract

In wireless sensor networks, sensor nodes collect different types of data from the environment and not all collected data has the same security importance. Basically, for real-time applications, user authentication to ensure that only authorized users can access sensor nodes is critical, but access control that allows users with different privileges to access data according to their privileges is also important. Recently, Adavoudi-Jolfaei et al. proposed an improved three-factor authentication scheme by providing more desired security properties such as three-factor authentication and access control. In this paper, however, we show that the Adavoudi-Jolfaei et al.'s scheme has security flaws; a user collusion attack, de-synchronization attack, and no providing sensor node anonymity. We present simple countermeasures against the security flaws we have found.

Keywords: wireless sensor networks, three-factor authentication, access control

1 Introduction

Wireless sensor networks (WSNs) are composed of many low-cost and low-power sensor nodes for monitoring environmental events including movement, temperature and humidity. WSNs have become an important network infrastructure in various Internet of Things (IoT) applications such as wildlife monitoring, industrial monitoring, health-care, and so on. Unlike the previous WSNs where the sensed data may be accessed only at the base stations, users in WSNs for IoT applications can directly access data at the sensor node from anywhere [5]. In this case, unauthorized users should be unable to access the sensor node, and only authorized users should be able to access and acquire data from the sensor node in a secure way. For the purpose, many user authentication and key agreement schemes have been proposed [4, 13, 2, 3, 9, 10, 11, 12]. Meanwhile, not all data in a sensor node is always equally important or has the same security level and some data may need to hide from users. In other words, all authenticated users do not have the privilege to access all kinds of data from the sensor node and the data of the sensor nodes that can be accessed should be different according to the users' access privileges. Therefore, it is important to provide access control as much as user authentication.

In recent, Adavoudi-Jolfaei et al. [1] demonstrated a security vulnerability on Gope et al.'s [7] two-factor authentication protocol in WSNs. To remedy the vulnerability on the Gope et al.'s scheme, they devised an enhanced scheme over the Gope et al.'s scheme by employing biometrics information with a fuzzy extractor and by providing access control as an additional desired security property for WSNs.

Research Briefs on Information & Communication Technology Evolution (ReBICTE), Vol. 5, Article No. 09 (November 30, 2019)

*This work was supported as part of Military Crypto Research Center(UD170109ED) funded 544 by Defense Acquisition Program Administration(DAPA) and Agency for Defense Development(ADD).

†Corresponding author: Graduate School of Information, Yonsei University, 50 Yonsei-ro Seodaemun-gu, Seoul, 03722, South Korea, Tel: +82-2-2123-4523

They proved their scheme is secure against various attacks using the Burrows-Abadu-Needham (BAN) logic. However, the Adavoudi-Jolfaei et al.'s scheme still has several security flaws.

In this paper, we aim to explain the security flaws of the Adavoudi-Jolfaei et al.'s scheme. We show that their scheme fails to provide sensor anonymity and suffers from a user collusion attack and desynchronization attack. The remainder of the paper is organized as follows: Section 2 presents a brief review of the Adafvoudi-Jolfaei et al.'s scheme. Section 3 reveals the security flaws of the Adavoudi-Jolfaei et al.'s scheme. Section 4 illustrates the simple countermeasures to remedy its security flaws. Section 5 finally concludes the paper.

2 Review of Adavoudi-Jolfaei et al.'s Scheme

In this section, we review the Adavoudi-Jolfaei et al.'s scheme [1], a lightweight and anonymous three-factor authentication and access control scheme. The scheme consists of four phases: registration, anonymous authentication and key exchange, password and biometric update, and dynamic node addition. We briefly present the user registration and anonymous authentication and key exchange phases related to security flaws. Table 1 shows the notations used in the Adavoudi-Jolfaei et al.'s scheme.

Table 1: List of notations used in Adavoudi-Jolfaei et al.'s scheme.

Notation	Description	Notation	Description
U	User	N_u	Random number generated by U
GW	Gateway node	SK	Session key between U and SN
SN	Sensor node	APM	A set of users' access privilege masks
SC	Smart card	G	A set of users' group IDs
ID_u	Identity of the user	K_{ug}	Shared key between U and GW
AID_u	One-time-alias identity of U	KEM_{ug}	Shared emergency key between U and GW
SID	Shadow identity of the user	K_{gs}	Secret key between GW and SN
ID_G	Identity of the gateway	Ts_{ug}	Transaction sequence number
w	Secret key of the gateway	$h(\cdot)$	One-way hash function
SN_{id}	Identity of the sensor node	$\oplus$	XOR operation
PSW_u	Password of the user	B_u	Biometric of the user
$GEN(B_u)$	One part of fuzzy extraction function, output a biometric key RS_u, and a helper string A_u		
$REP(B_u, A_u)$	One part of fuzzy extraction function, output the biometric key RS_u in $GEN(B_u)$		

In both Gope et al.'s scheme and Adavoudi-Jolfaei et al.'s scheme, the sensor registration phase was missed, thus we add it according to the papers [7, 1]. Before the WSN deployment, GW preloads SN_{id} and K_{gs} into the memory of each SN and saves SN_{id} and $K_{gs}^{\#}$ into the database, where $K_{gs}^{\#} = K_{gs} \oplus h(ID_G||w||SN_{id})$. In Adavoudi-Jolfaei et al.'s scheme, for providing access control, GW generates a set of access group-IDs $G = \{G_1, G_2, \cdots\}$ and a set of access privilege masks $APM = \{APM_1, APM_2 \cdots\}$, where $G_j \in G$ is a 128-bit unique random number used to identity a particular access group and $APM_j \in APM$ is a 128-bit random number except first 16-bits (high order) in which each bit defines different task or service. A user can belong to one or more access groups and multiple users who have similar access privileges can be organized into the same group.

2.1 Registration phase

In this phase, GW issues a smart card to an intended user via secure channel. During this phase, depending on the probable user query, GW prepares an access list which defines the user's privilege and consists of ID_u, G_j, and user access privilege mask APM_j.

1. $U \Rightarrow GW$: $\langle ID_u$, Personal credential$\rangle$

2. $GW \Rightarrow U$: a smart card containing $\{K_{ug}, (SID, KEM_{ug}), Ts_{ug}, G_u, h(\cdot)\}$, where $K_{ug} = h(ID_u||n_g) \oplus ID_G, sid_j = h(ID_u||r_j||K_{ug}), SID = \{sid_1, sid_2, \cdots\}, KEM_{ug_j} = h(ID_u||sid_j||r'_j)$, n_g, r_j, r'_j are random numbers generated by GW, and Ts_{ug} a 64-bit random sequence number generated by GW. For U, GW finally saves $\langle Ts_{ug}, (SID, KEM^{\#}_{ug}), K^{\#}_{ug}, K^{\#}_{gs}, ID^{\#}_u, G^{\#}, APM^{\#}\rangle$ into the database, where $KEM^{\#}_{ug} = KEM_{ug} \oplus h(ID_G||ID_u||w), K^{\#}_{ug} = K_{ug} \oplus h(ID_G||ID_u||w), ID^{\#}_u = ID_u \oplus h(ID_G||ID_u||w), G^{\#}_j = G_j \oplus h(ID_G||ID_u||w), G^{\#} = \{G^{\#}_1, G^{\#}_2, \cdots\}, APM^{\#}_j = APM_j \oplus h(ID_G||ID_u||w)$, and $APM^{\#} = \{APM^{\#}_1, APM^{\#}_2, \cdots\}$.

3. U inputs PSW_u and B_u; then SC stores $\langle K^*_{ug}, f^*_{ug}, (SID^*, KEM^*_{ug}), Ts_{ug}, G^*, A_u, GEN(\cdot), REP(\cdot), h(\cdot)\rangle$ in its memory, where $GEN(B_u) = (RS_u, A_u), K^*_{ug} = h(h(ID_u) \oplus h(PSW_u) \oplus h(RS_u)), KEM^*_{ug} = KEM_{ug} \oplus h(h(ID_u) \oplus h(PSW_u) \oplus h(RS_u)), SID^* = SID \oplus h(h(ID_u) \oplus h(PSW_u) \oplus h(RS_u)), G^* = G \oplus h(h(ID_u) \oplus h(PSW_u) \oplus h(RS_u)), f^*_u = h(h(K_{ug}) \oplus h(ID_u) \oplus h(PSW_u) \oplus h(RS_u))$.

2.2 Anonymous authentication and key exchange phase

In both Gope et al.'s scheme and Adavoudi-Jolfaei et al.'s scheme, to speed up the authentication processes and to prevent any replay attack, a 64-bit random sequence number, Ts_{ug}, is used as an one-time pseudonym. In addition, to provide user anonymity and untraceability, they also employed a set of unlinkable shadow-IDs SID and a corresponding set of emergency keys KEM. These values are used in the case of loss of synchronization of Ts_{ug} between U and GW.

1. $U \Rightarrow GW$: $\langle AID_u, G'_j, N_x, Ts_{ug}$ (if req), $SN_{id}, V_1\rangle$. U inputs ID_u, PSW_u and biometrics B_u, then SC computes $RS_u = REP(B_u, A_u), K_{ug} = K^*_{ug} \oplus h(h(ID_u) \oplus h(PSW_u) \oplus h(RS_u))$, and $f_u = h(h(K_{ug}) \oplus h(ID_u) \oplus h(PSW_u) \oplus h(RS_u))$. SC checks $f_u \stackrel{?}{=} f^*_u$. If so, SC computes $N_x = K_{ug} \oplus N_u$, where N_u is a random number generated by U, $G = G^* \oplus h(h(ID_u) \oplus h(PSW_u) \oplus h(RS_u))$ and $AID_u = h(ID_u||K_{ug}||N_u||Ts_{ug})$; then U chooses an access group-ID G_j from G. Finally, SC computes $G'_j = G_j \oplus N_u$ and $V_1 = h(AID_u||G'_j||K_{ug}||N_x||SN_{id})$ and sends a request message to GW. In case of loss of synchronization, U chooses one of the unused pair of (sid_j, KEM_{ug_j}) from (SID^*, KEM^*_{ug}) and assigns sid_j as AID_u and KEM_{ug_j} as K_{ug}.

2. $GW \Rightarrow SN$: $\langle AID_u, APM'_j, SK', T, V_2\rangle$. GW first checks the validity of Ts_{ug}. If GW cannot find it provided by U in its database, it terminates the connection. Otherwise, GW selects the related tuple to U using Ts_{ug}. GW decodes ID_u and K_{ug} and checks the validity of V_1. If so, GW computes $N_u = N_x \oplus K_{ug}$ and $G_j = G_j \oplus N_u$, then checks $AID'_u \stackrel{?}{=} AID_u$, where $AID'_u = h(ID_u||K_{ug}||N_u||Ts_{ug})$. If so, GW computes $APM'_j = h(K_{gs}) \oplus APM_j$ by finding APM_j related to G_j and generates SK and a timestamp T and finally sends the message by computing $SK' = h(K_{gs}) \oplus SK$ and $V_2 = h(AID_u||APM'_j||SK'||T||K_{gs})$. In case of loss of synchronization, U will re-send the request message using $AID_u = sid_j$ and $K_{ug} = KEM_j$ instead of using Ts_{ug}. In this case, GW will check the validity of AID_u by comparing sid_j with the entries in its database. If GW can find it, then GW derives the tuple associated to sid_j and retrieves KEM_j. GW checks the validity of V_1 with these values and proceeds further processes.

3. $SN \Rightarrow GW$: $\langle T', SN_{id}, V_3\rangle$. SN first checks the freshness of T and verifies V_2. If so, SN computes $APM_j = APM'_j \oplus h(K_{gs})$ and generates a timestamp T'. SN then derives $SK = SK' \oplus h(K_{ug})$ and computes $V_3 = h(SK||K_{gs}||SN_{id}||T')$. Finally, SN sends the response message and updates $K_{gs} = K_{gs_{new}}$, where $K_{gs_{new}} = h(K_{gs}||SN_{id})$.

4. $GW \Rightarrow U$: $\langle SK'', V_4, Ts, x \text{ (if req)} \rangle$. GW first checks the freshness of T' and generates a random number m and computes $Ts_{ug_{new}} = m, Ts = h(K_{ug}||ID_u||N_u) \oplus Ts_{ug_{new}}, SK'' = h(K_{ug}||ID_u||N_u) \oplus SK$, and $V_4 = h(SK''||N_u||Ts||K_{ug})$. Finally, GW sends the response message and updates $K_{ug} = K_{ug_{new}}$ and $K_{gs} = K_{gs_{new}}$, where $K_{ug_{new}} = h(K_{ug}||ID_u||Ts_{ug_{new}})$ and $K_{gs_{new}} = h(K_{gs}||SN_{id})$. In the case of loss of synchronization, instead of the above update method, GW randomly generates $K_{ug_{new}}$ and sends $x = K_{ug_{new}} \oplus h(ID_u||KEM_j)$ with other parameters.

5. U first checks V_4. If so, U derives $SK = SK'' \oplus h(K_{ug}||ID_u||N_u)$ and updates $Ts_{ug} = Ts_{ug_{new}}$ and $K_{ug} = K_{ug_{new}}$, where $Ts_{ug_{new}} = h(K_{ug}||ID_u||N_u) \oplus Ts$ and $K_{ug_{new}} = h(K_{ug}||ID_u||Ts_{ug_{new}})$. In the case of loss of synchronization, U differently updates $K_{ug} = K_{ug_{new}}$, where $K_{ug_{new}} = h(ID_u||KEM_j) \oplus x$.

3 Security Flaws in Adavoudi-Jolfaei et al.'s Scheme

In this section, we show that the Adavoudi-Jolfaei et al.'s scheme has several security flaws.

3.1 User collusion attack

Since users' access group-IDs are given to users as they are in the registration phase, the users can exploit other users' group-IDs through user colluding to obtain sensor data required higher privileges. GW stores the group ID that a user has in the database, but does not verify that the group ID presented by the user in the anonymous authentication and key exchange phase is the group to which the user belongs. Therefore, the Adavoudi-Jolfaei et al.'s scheme is vulnerable to a user collusion attack.

3.2 De-synchronization attack

Adavoudi-Jolfaei et al. showed that the Gope et al's scheme [7] is vulnerable to a session key disclosure attack. To solve this problem, they used the vulnerable update method that Gope et al. pointed out [6, 7]. In other words, the updated $Ts_{ug_{new}}$ is transmitted to a user, thus if the last response message sent from GW is disrupted by an adversary, it will cause loss of synchronization between the user and GW.

Both schemes utilized a set of shadow IDs SID and the corresponding set of emergency keys KEM_{ug} for each user to solve the problem of loss of synchronization. However, it also causes another de-synchronization attack or DoS attack. In the registration phase, if GW cannot find Ts_{ug} of the request message sent from U in its database, then GW will terminate the connection. Upon receiving this termination message, U will re-send the request message using one of the shadow ID and emergency key. In that case, an adversary can exploit this method by arbitrarily changing Ts_{ug} of the request message to break the synchronization between GW and U and to exhaust SID and KEM_{ug} shared between them.

3.3 No sensor node anonymity

In the Adavoudi-Jolfaei et al.'s scheme, U and SN send the request message $\langle AID_u, G'_j, N_x, Ts_{ug} \text{ (if req)}, SN_{id}, V_1 \rangle$ and response message $\langle T', SN_{id}, V_3 \rangle$ to GW via insecure channel, respectively. Clearly, if an adversary intercepts either the request message of U or the response message of SN, he/she can obtain SN's identity SN_{id}. Thus, the Adavoudi-Jolfaei et al.'s scheme does not ensure sensor node anonymity.

4 Countermeasure

In this section, we present a simple countermeasure against the above security flaws. The first problem of the Adavoudi-Jolfaei et al.'s scheme is that the access group-IDs used to prove the user's privileges

are exposed to the user and *GW* does not verify that the given user's access group-ID during the login is what was granted to that user. To solve the first problem, for each user, access group-IDs are transformed into a value associated with the user and *GW* verifies whether the access group ID presented by the user is correct or not. The second problem is that a random sequential number Ts_{ug} used for speeding up the authentication process and preventing replay attack and a shadow-ID sid_j and an emergency key KEM_j used for user anonymity and untraceability are rather a target of attacks. To solve the problem, we employ efficient elliptic curve cryptography (ECC) operations (i.e., only twice point multiplications at the user side and once point multiplication at the gateway side) instead of random serial numbers, shadow IDs, and emergency keys for user anonymity and untraceability, and utilize a timestamp to prevent replay attacks. Due to space limitations, we don't introduce the basic knowledge about ECC and the reader can refer [8] for ECC. The last problem is that the identity of the sensor node is exposed in the messages. To solve the problem, we make sure that the sensor node IDs are not exposed to messages. We describe only the modifications made during the registration phase and the anonymous authentication and key exchange phase.

4.1 Modified Registration Phase

Before deployment, *GW* chooses an elliptic curve E over prime finite field F_q and an additional subgroup G of E, which generated by P with a large prime order p. *GW* then generates its private and public key pair $\{y, Q_g\}$, where $y \in \mathbb{Z}_p^*$ and $Q_g = yP$. *GW* publishes the system parameters $\{E, G, p, P, Q_g\}$. For providing access control, *GW* generates a set of access groups $GID = \{(GID_1 : G_1, APM_1), (GID_2 : G_2, APM_2), \cdots\}$ where GID_j identifies and specifies a particular access group, G_j is a 128-bit unique random number for GID_j, and $APM_j \in APM$ is a 128-bit random number except first 16-bits (high order) in which each bit defines different task or service. *GW* stores the set of access groups in the database regardless of users. The details of the modified registration phase are as follows.

2. $GW \Rightarrow U$: a smart card containing $\{K_{ug}, G_u, P, Q_g, h(\cdot)\}$, where $K_{ug} = h(ID_u||n_g) \oplus ID_G, M_u = h(ID_u||w||b)$, n_g and b are random numbers generated by *GW*. According to U's privileges, *GW* prepares $G_u = \{(GID_1, G_1^u), (GID_2, G_2^u), \cdots\}$, where $G_j^u = M_u \oplus G_j$. *GW* issues the smart card to the user and finally saves $ID_u, b, K_{ug}^{\#}$, $GID_u = \{GID_1, GID_2, \cdots\}$ into the database, where $K_{ug}^{\#} = K_{ug} \oplus h(ID_G||ID_u||w)$ for each U.

3. U inputs PSW_u and B_u; then *SC* stores $\langle K_{ug}^*, f_u^*, G_u^*, A_u, P, Q_g, GEN(\cdot), REP(\cdot), h(\cdot)\rangle$ in its memory, where $GEN(B_u) = (RS_u, A_u), K_{ug}^* = h(h(ID_u) \oplus h(PSW_u) \oplus h(RS_u)), G_u^* = G_u \oplus h(ID_u||PSW_u||RS_u)$, $f_u^* = h(h(K_{ug}) \oplus h(ID_u||PSW_u||RS_u))$.

4.2 Modified Anonymous Authentication and Key Exchange Phase

Although we employ ECC to erase the security flaws found in the Adavoudi-Jolfaei et al.'s scheme, ECC operations are used only by users and gateway with less resource constraints than sensor nodes. Moreover, sensor nodes utilize only efficient symmetric operations, thus it is as lightweight as the Adavoudi-Jolfaei et al.'s scheme in the side of sensor nodes. The details of the modified authentication and key exchange phase are as follows.

1. $U \Rightarrow GW$: $\langle TID_u, MSN_u, MG_j^u, X_u, V_1, T''\rangle$. U inputs ID_u, PSW_u and biometrics B_u, then *SC* computes $RS_u = REP(B_u, A_u), K_{ug} = K_{ug}^* \oplus h(h(ID_u) \oplus h(PSW_u) \oplus h(RS_u))$, and $f_u = h(h(K_{ug}) \oplus h(ID_u) \oplus h(PSW_u) \oplus h(RS_u))$. *SC* checks $f_u \stackrel{?}{=} f_u^*$. If so, *SC* computes $G_u = G_u^* \oplus h(ID_u||PSW_u||RS_u)$ and U selects proper GID_j and G_j^u. Then *SC* generates a random number x and timestamp T and computes $X_u = xP, Y_u = xQ_g, TID_u = ID_u \oplus h(X_u||Y_u), MSN_u = SN_{id} \oplus h(Y_u||T''), AC_j^u = G_j^u \oplus h(K_{ug}||T'')$, and $V_1 = h(ID_u||SN_{id}||G_j^u||K_{ug}||X_u||T'')$ and sends a request message to *GW*.

2. $GW \Rightarrow SN$: $\langle TID_u, APM'_j, SK', V_2, T \rangle$. GW first checks the freshness of T''. If so, GW computes $Y'_u = yX_u$ and $ID'_u = TID_u \oplus h(X_u||Y_u)$ and find the related tuple to U in its database using ID'_u. GW then computes $SN'_{id} = MSN_u \oplus h(Y_u||T)$, $G'^u_j = MG^u_j \oplus h(K_{ug}||T'')$, and $V'_1 = h(ID'_u||SN'_{id}||G'^u_j||K_{ug}||X_u||T'')$ and checks $V'_1 \stackrel{?}{=} V_1$. If so, GW computes $M_u = h(ID_u||w||b)$ and $G'_j = G'^u_j \oplus M_u$ and verifies that the GID_j of G'_j is contained in G_u. If so, GW computes $APM'_j = h(K_{gs}) \oplus APM_j$ by finding APM_j related to G'_j, generates SK and a timestamp T, and finally sends the message by computing $SK' = h(K_{gs}) \oplus SK$ and $V_2 = h(TID_u||SN_{id}||APM_j||SK||T||K_{gs})$.

3. $SN \Rightarrow GW$: $\langle TID_u, V_3, T' \rangle$. SN first checks the freshness of T and verifies V_2. If so, SN computes $APM_j = APM'_j \oplus h(K_{gs})$ and generates a timestamp T'. SN then derives $SK = SK' \oplus h(K_{ug})$ and computes $V_3 = h(TID_u||SN_{id}||SK||T'||K_{gs})$. Finally, SN sends the response message and updates $K_{gs} = K_{gs_{new}}$, where $K_{gs_{new}} = h(K_{gs}||SN_{id})$.

4. $GW \Rightarrow U$: $\langle SK'', V_4, Ts, x \text{ (if req)} \rangle$. GW first checks the freshness of T' and verifies V_3. If so, GW generates a timestamp T''' and computes $SK'' = SK'' = h(K_{ug}||ID_u||Y_u) \oplus SK$, and $V_4 = h(ID_u||SN_{id}||SK||T'''||K_{ug})$. Finally, GW sends the response message and updates $K_{ug} = K_{ug_{new}}$ and $K_{gs} = K_{gs_{new}}$, where $K_{ug_{new}} = h(K_{ug}||TID_u)$ and $K_{gs_{new}} = h(K_{gs}||SN_{id})$.

5. U first checks the freshness of T''' and verifies V_4. If so, U derives $SK = SK'' \oplus h(K_{ug}||ID_u||Y_u)$ and updates $K_{ug} = K_{ug_{new}}$, where $K_{ug_{new}} = h(K_{ug}||TID_u)$. From now on, U can communicate with SN using TID_u and SK in a secure way including anonymity and untraceability.

5 Conclusion

In this paper, we have reviewed the recently proposed the Adavoudi-Jolfaei et al.'s authentication and access control scheme for real-time applications in WSNs. We have analyzed the security flaws of the Adavoudi-Jolfaei et al.'s scheme. We have pointed out that the Adavoudi-Jolfaei et al.'s scheme failed to provide sensor node anonymity and it is vulnerable to user collusion attack and de-synchronization attack. We have briefly presented the countermeasures against those security flaws of the Adavoudi-Jolfaei et al.'s scheme. In the future work, we will propose an enhanced anonymous authentication and access control scheme for WSNs. We will also analysis security and performance of the enhanced scheme.

References

[1] A. Adavoudi-Jolfaei, M. Ashouri-Talouki, and S. F. Aghili. 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, January 2019.

[2] I.-P. Chang, T.-F. Lee, T.-H. Lin, and C.-M. Liu. Enhanced two-factor authentication and key agreement using dynamic identities in wireless sensor networks. *Sensors*, 15(12):29841–29854, December 2015.

[3] A. K. Das and A. Goswami. A robust anonymous biometric-based remote user authentication scheme using smart cards. *Journal of King Saud University - Computer and Information Sciences*, 27(2):193–210, April 2015.

[4] M. L. Das. Two-factor user authentication in wireless sensor networks. *IEEE Transactions on Wireless Communications*, 8(3):1086–1090, March 2009.

[5] Y. Faye, L. Niang, and T. Noël. A survey of access control schemes in wireless sensor networks. *International Journal of Computer and Information Engineering*, 5(11):1254–1263, November 2011.

[6] P. Gope and T. Hwang. A realistic lightweight authentication protocol preserving strong anonymity for securing RFID system. *Computers & Security*, 55:271–280, November 2015.

[7] P. Gope and T. Hwang. 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, November 2016.

[8] D. Hankerson, A. Menezes, and S. Vanstone. *Guide to Elliptic Curve Cryptography*. Springer-Verlag, Berlin, Heidelberg, December 2003.

[9] Q. Jiang, J. Ma, F. Wei, Y. Tian, J. Shen, and Y. Yang. An untraceable temporal-credential-based two-factor authentication scheme using ECC for wireless sensor networks. *Journal of Network and Computer Applications*, 76:37–48, December 2016.

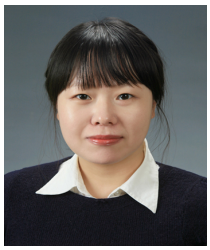
[10] Q. Jiang, S. Zeadally, J. Ma, and D. He. Lightweight three-factor authentication and key agreement protocol for internet-integrated wireless sensor networks. *IEEE Access*, 5:3376–3392, March 2017.

[11] S. Shin and T. Kwon. Two-factor authenticated key agreement supporting unlinkability in 5G-integrated wireless sensor networks. *IEEE Access*, 6:11229–11241, January 2018.

[12] S. Shin and T. Kwon. A lightweight three-factor authentication and key agreement scheme in wireless sensor networks for smart homes. *Sensors*, 19(9):2012:1–2012:24, April 2019.

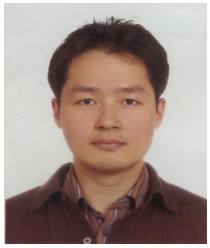
[13] M. Turkanović, B. Brumen, and M. Hölbl. A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the internet of things notion. *Ad Hoc Networks*, 20:96–112, September 2014.

Author Biography



Sooyeon Shin received her B.S., M.S., and Ph.D. degrees in computer science and engineering from Sejong University, Seoul, Korea, in 2004, 2006, and 2012, respectively. From 2012 to 2013, she was a post-doctoral researcher at Sejong University. In 2013, she joined Yonsei University, Seoul, Korea, to continue her post-doctoral research. Her current research interests include cryptographic protocol, privacy preservation, user authentication, computer network security, wireless sensor network security, and usable security.

Jongshin Kim rreceived his B.S. degrees in embedded software engineering from Busan University of Foreign Studies, Busan, Korea, in 2018. He is currently working toward the M.S. degree at the Graduate School of Information, Yonsei University, Seoul Korea. His current research interests include wireless network security and information security.



Taekyoung Kwon received his B.S., M.S., and Ph.D. degrees in computer science from Yonsei University, Seoul, Korea, in 1992, 1995, and 1999, respectively. He is currently an Associate Professor of information at Yonsei University, Seoul, Korea. From 1999 to 2000, he was a Post-Doc Researcher at the University of California, Berkeley, CA, USA. From 2001 to 2013, he was a professor of computer engineering at Sejong University, Seoul, Korea. In 2013, he returned to Yonsei University, Seoul, Korea. His current research interests include applied cryptography, cryptographic protocol, network protocol, usable security, and human-computer interactions.