Cryptanalysis on Privacy-Aware Two-Factor Authentication Protocol for Wireless Sensor Networks

Younsung Choi*
Department of Cyber Security, Howon University, 64, 3-gil, Gunsan, Jeollabuk-do, 54058, Republic of Korea
*Corresponding author, e-mail: yschoi@howon.ac.kr

Abstract
Das first proposed two-factor authentication combining the smart card and password to resolve the security problems of wireless sensor networks (WSNs). After that, various researchers studied two-factor authentication suitable for WSNs. In user authentication protocols based on the symmetric key approach, a number of elliptic curve cryptography (ECC)-based authentication protocols have been proposed. To resolve the security and efficiency problems of ECC-based two-factor authentication protocols, Jiang et al. proposed a privacy-aware two-factor authentication protocol based on ECC for WSNs. However, this paper performs a vulnerability analysis on Jiang et al.’s authentication protocol and shows that it has security problems, such as a lack of mutual authentication, a risk of S/D modification and DoS attacks, a lack of sensor anonymity, and weak ID anonymity.

Keywords: Cryptanalysis, Vulnerability analysis, Wireless sensor networks

Copyright © 2017 Institute of Advanced Engineering and Science. All rights reserved.

1. Introduction
Wireless sensor networks (WSNs) can be used to perform real-time monitoring in various environments. Networked sensors can easily be stationed in various environments (e.g., for forest detection and harmful gas monitoring) [1]. Generally, the gateway node has sufficient power and capacity, while the wireless sensors lack sufficient CPU power, memory, computational capability, and storage capacity. Therefore, generally, a user needs to connect with sensors directly to acquire the sensed data [2]. Considering the resources of sensors, the user authentication protocol for WSNs should be efficient in terms of computation cost. Therefore, the power consumption of the cryptographic algorithms used should be reduced while addressing the security requirements. To resolve the difficulty of designing a secure two-factor authentication protocol, a privacy-aware two-factor protocol that addressed various security problems with the resource sensors and sensed data was designed in [3].

In 2009, Das first applied two-factor authentication combining the password and smart card to solve the security problems of WSNs. It presented a new direction for user authentication for WSNs [4]. However, the authentication protocol Das proposed does not provide user anonymity, session key negotiation, or mutual authentication. In addition, it is vulnerable to several attacks, such as gateway node bypassing, offline password guessing, sensor node capture, and denial-of-service attacks. Thus, various improved authentication protocols for WSNs were proposed to resolve the various security problems [5-7]. In addition, in user authentication protocols based on the symmetric key approach, a number of elliptic curve cryptography (ECC)-based authentication protocols have been proposed. Yeh et al. found that the protocol of Chen et al. does not provide a user password updating mechanism and is vulnerable to insider attacks. Thus, Yeh et al. proposed an ECC-based two-factor authentication protocol. However, in Yeh et al.’s scheme, the user and sensor cannot mutually authenticate each other [8]. To solve the problems of Yeh et al.’s scheme, Shi et al. proposed an improved ECC-based authentication protocol. Compared with the protocol of Yeh et al., the protocol of Shi et al. provides more diverse security features and performs better in terms of computation and communication [9]. However, in 2014, Choi et al. revealed that the authentication protocol of Shi et al. is vulnerable to unknown key share, stolen smart card, and sensor energy exhausting attacks. To eliminate these security weaknesses, they also proposed an enhanced authentication protocol [1]. Unfortunately, the protocol of Choi et al. still cannot achieve
anonymity and untraceability. To solve the various security weaknesses of ECC-based two-factor authentication protocols, Jiang et al. proposed a privacy-aware two-factor authentication protocol based on ECC for WSNs. Jiang et al. claim their protocol achieves various security and usability features necessary for real-life application environments [2]. However, this paper analyzes Jiang et al.’s protocol and shows that it has security vulnerabilities, such as a lack of mutual authentication, a risk of SID modification and DoS attacks, a lack of sensor anonymity, and weak ID anonymity. The remainder of this paper is organized as follows. Section 2 explains Jiang et al.’s privacy-aware two-factor authentication protocol based on ECC for WSNs. Section 3 shows that Jiang et al.’s authentication protocol has the security vulnerabilities noted above. Section 4 concludes this paper.

2. Review of Jiang et al.’s two-factor Authentication Protocol

Jiang et al.’s protocol is based on ECC for WSNs. It consists of four phases: registration, login, authentication, and password change. Table 1 shows the notations used in this paper [2]. The ECC provides better efficiency than Rivest Shamir and Adleman (RSA), because it can achieve the same security strength with a smaller key size. Specifically, the 160-bit ECC and the 1024-bit RSA have the same security strength [10, 11]. The elliptic curve equation is defined in the form: \( E_p(a, b) : y^2 = x^3 + ax + b \pmod{p} \) over a prime finite field \( F_p \), where \( a, b \in F_p \), and \( 4a^3 + 27b^2 \neq 0 \pmod{p} \).

| Notation | Description | Notation | Description |
|----------|-------------|----------|-------------|
| \( U \)  | A user      | \( GWN \) | A gateway node |
| \( S_i \) | Sensor node | \( SID_i \) | Sensor node identity |
| \( H(\cdot) \) | Hash function | \( ID_i \) | The identity of \( U_i \) |
| \( PW_i \) | The password of \( U_i \) | \( TS \) | The current timestamp |
| \( SK_i \) | Shared session key | \( PTC_i \) | Protected temporal credential of \( U_i \) |
| \( DID_i, DID_{GWN} \) | A dynamic identity of \( U_i \) and \( S_i \) | \( TC_i, TC_j \) | Temporal credential of \( U_i \) and \( S_j \) |
| \( TE_i \) | The expiration time of a user’s temporal credential | \( K_{GWN-U} \) | Master keys only known to \( GWN \) |
| \( \oplus \) | The bitwise concatenation | \( \oplus \) | The bitwise exclusive OR |

2.1. Registration Phase

Prior to starting Jiang et al.’s authentication protocol, \( GWN \) selects the finite cyclic additional group \( G \) generated by a point \( P \) with a large prime order \( n \) over a finite field \( F_p \) on an elliptic curve. Then, \( GWN \) randomly chooses a number \( x \) as its private key, computes the corresponding public key \( y = xP \), and generates two master secret keys \( K_{GWN-U} \) and \( K_{GWN-S} \). Then, \( GWN \) stores \( x \) and produces the system parameters \( \{E(F_p), G, P, y\} \). Figure 1 shows the user registration process. It is assumed that the communication channel between the participants is secure.

(R1-U) When a user \( U_i \) registers to \( GWN, U_i \) selects his/her own identity \( ID_i \) and password \( PW_i \) and randomly chooses a number \( r_i \). Then, \( U_i \) calculates \( HPW_i = H(PW_i || ID_i || r_i) \) and sends \( \{ID_i, HPW_i\} \) to \( GWN \).

(R2-U) After receiving the request, \( GWN \) checks the legitimacy of \( ID_i \) and refuses the request if \( ID_i \) does not adapt to the requirement of user identity or is the same as an already registered identity in the verification table. Then, \( GWN \) computes \( TC_i = H(K_{GWN-U} || ID_i || TE_i) \) and \( PTC_i = TC_i \oplus HPW_i \). \( GWN \) stores \( \{ID_i, TE_i\} \) in the verification table. Finally, \( GWN \) publishes the card, which embraces \( \{H(\cdot), y, TE_i, PTC_i\} \) to \( U_i \).

(R3-U) \( U_i \) computes \( HPW_i' = H(h(ID_i || PW_i || r_i) \pmod{m}) \), where \( m = 2^{16} \leq m \leq 2^{16} \) integer, which determines the capacity of the pool of \( \langle ID_i, PW_i > \) pairs against offline password guessing attacks [12]. Then, \( U_i \) hoards \( r_i \) and \( HPW_i' \) into the card.

The sensor registration process is described as follows:

(R1-S) \( S_j \) presents its identity \( SID_j \) to \( GWN \) using a secure channel.

(R2-S) \( GWN \) computes \( TC_j = H(K_{GWN-S} || SID_j) \) as the credential for \( S_j \). Then, \( GWN \) replies to \( S_j \) with \( \{TC_j\} \).

(R3-S) After receiving the response, \( S_j \) keeps \( TC_j \).
2.2. Login Phase

The following steps are performed in the system login phase.

(L1) When $U_i$ wants to access $S_i$, $U_i$ slots the smart card into a terminal and inputs $ID_i$, $PW_i$.

(L2) The smart card calculates $HPW_i' = H(h(ID_i || PW_i || r_i) \mod m)$. If the comparison $HPW_i' ? = HPW_i$ is not the same, the card rejects the request. Otherwise, it continues to compute $TC_i = PTC_i \oplus H(PW_i || ID_i || r_i)$.

2.3. Authentication Phase

Subsequent to the login phase, the communicating agents ($U_i$, $S_i$, and $GWN$) mutually authenticate each other and establish a session key as follows. Figure 2 depicts these phases.

(A1) $U_i$ selects a random number $a \in Z_{p-1}$ and calculates $A_i = aP$, $D_i = ay = axP$, $DID_i = ID_i \oplus H(A_i || D_i)$, and $C_i = H(ID_i || TS_i || D_i || A_i || TC_i)$, where $TS_i$ is the timestamp of the current computing platform. Finally, $U_i$ forwards $\{DID_i, A_i, TS_i, C_i\}$ to $GWN$.

(A2) On receiving $\{DID_i, A_i, TS_i, C_i\}$, $GWN$ verifies the freshness of $TS_i$. If $TS_i$ is not fresh, $GWN$ refuses the request; otherwise, $GWN$ calculates $D_i = xA = axP$, $ID_i = DID_i \oplus H(A_i || D_i)$, and $TC_i = H(DID_i || ID_i || TE_i)$ and checks whether $H(ID_i || TS_i || D_i || A_i || TC_i)$ is the same as $C_i$. If these two values are not the same, $GWN$ refuses the request; otherwise, $GWN$ chooses a sensor $S_i$ and calculates $TC_i = H(ID_{GWN-S} || SID_i)$, $DID_{GWN} = DID_i \oplus H(DID_i || TS_i || TS_{S_i})$, and $C_{GWN} = H(ID_i || TC_i || A_i || TS_{S_i})$, where $TS_{S_i}$ is the timestamp of the current computing platform. Finally, $GWN$ sends $\{TS_{S_i}, DID_i, DID_{GWN}, C_{GWN}, A_i\}$ to the $S_i$.

(A3) On receiving $\{TS_{S_i}, DID_i, DID_{GWN}, C_{GWN}, A_i\}$, $S_i$ checks the freshness of $TS_{S_i}$. If $TS_{S_i}$ is invalid, $S_i$ rejects the request; otherwise, $S_i$ computes $ID_i = DID_{GWN} \oplus H(DID_i || TC_i || TS_{S_i})$ and checks whether $H(ID_i || TC_i || A_i || TS_{S_i})$ and $C_{GWN}$ are equal. If these two values are unequal, $S_i$ terminates the current session; otherwise, $S_i$ generates a random key $b \in Z_{p-1}$ and computes $B_i = bP$, $SK_g = H(bA_i) = H(abP)$, and $C_i = H(TC_i || ID_i || SID_i || B_i || TS_i)$, where $TS_i$ is the current timestamp. $S_i$ then sends $\{SID_i, TS_i, C_i, B_i\}$ to $GWN$.

(A4) After checking the legitimacy of $TS_{S_i}$, $GWN$ checks whether $H(TC_i || ID_i || SID_i || B_i || TS_i)$ and $C_i$ are the same. If these two values are not equal, $GWN$ stops the current session; otherwise, $GWN$ confirms that $S_i$ is authenticated. Finally, $GWN$ calculates $E_{GWN} = H(ID_i || TC_i || D_i || B_i || TS_s)$, where $TS_s$ is the timestamp of the current computing platform, and sends $\{SID_i, TS_s, B_i, E_{GWN}\}$ to $U_i$.

(A5) After checking the freshness of $TS_s$, $U_i$ computes and checks whether $H(ID_i || TC_i || D_i || B_i || TS_s)$ and $E_{GWN}$ are equal. If these two values are not the same, $U_i$ stops the current session; otherwise, $U_i$ confirms that $S_i$ and $GWN$ are authenticated. Finally, $U_i$ computes the shared session key $SK_k = H(abB_i) = H(abP)$. 

Figure 1. Registration phase of Jiang et al. ’s protocol
2.4. Password Change Phase

(PC1) If $U_i$ wants to update his/her own password, he or she inputs his/her own card into a terminal and enters $ID_i$ and $PW_i$.

(PC2) The smart card calculates $H(h(ID_i) \parallel PW_i \parallel r_i) \mod m)$. If the equations $HPW'_i \neq HPW_i$ are not the same, the card refuses the request. Otherwise, $U_i$ inputs the old $PW_i$, selects a new $PW'_i$, calculates $PTC'_i = TC_i \oplus RPW_i \oplus H(r \parallel PW'_i)$, and replaces $PTC_i$ with $PTC'_i$. 

\[ \text{User } U_i \]
\[ (ID_i, PW_i, PW'_i) \]
\[ \text{Insert smart card; } \]
\[ \text{Input previous } ID_i \text{ and } PW_i \text{; } ID_i, PW_i \text{; } \]
\[ \text{Compute } HPW'_i = H(h(ID_i) \parallel PW_i \parallel r_i) \mod m) ; \]
\[ \text{Check } HPW'_i \neq HPW_i ; \]
\[ \text{Input new } PW_i \text{; } PW'_i ; \]
\[ \text{Compute } PTC'_i = TC_i \oplus RPW_i \oplus H(r \parallel PW'_i) ; \]
\[ \text{Replace } PTC_i \text{ with } PTC'_i ; \]

Figure 2. Login and authentication phase of Jiang et al.’s protocol

Figure 3. Password change phase of Jiang et al.’s protocol
3. Cryptanalysis on Jiang et al.’s Two-Factor Authentication Protocol

This paper analyzes Jiang et al.’s authentication protocol and determines various security vulnerabilities, including a lack of mutual authentication, a risk of SID modulation and DoS attacks, a lack of sensor anonymity, and weak ID anonymity.

3.1. Lack of Mutual Authentication

Mutual authentication means that two or three parties authenticate each other. All of the parties (e.g., client/user, gateway, and sensors) are assured of the others’ identity. The user and gateway authenticate each other using IDi and TCi, while the gateway and sensors authenticate each other using TCj and CGWN. However, mutual authentication between the user and sensors is not provided. The sensors can authenticate the user with the gateway’s help. However, the user cannot authenticate the sensors. Thus, the user cannot verify whether the sensor SIDj is normal.

3.2. Risk of SID Modification Attacks

The user receives \{SIDi, TS2, Bi, EGWN\} from GWN and checks the message’s accuracy and freshness. However, there is no information indicating that SIDi in \{SIDi, TS2, Bi, EGWN\} is now authenticated by GWN, so an attacker can perform a SID modification attack. When the attacker modifies the SIDi in \{SIDi, TS2, Bi, EGWN\} to SIDattacker, the user is unaware of the change. Therefore, the user mistakenly believes that SIDattacker is a normal sensor node and thus computes the session key SKij for secure communication with SIDattacker even though the attacker cannot know the SKij. Moreover, when SIDi requests communication, the user cannot know whether SIDi is an authenticated sensor node, so they cannot communicate with each other.

3.3. Lack of Sensor Anonymity

Anonymity is a desirable security feature, and it provides identification and key agreement of the user and sensors during the login and authentication phases. Thus, Jiang et al.’s authentication protocol provides the user’s dynamic identification DIDi to protect the user’s anonymity. Moreover, this protocol uses DIDGWN to protect the gateway node’s identification. However, Jiang et al.’s authentication protocol does not provide anonymity of the sensor node. Therefore, an attacker can know which sensor node is communicating with users. In addition, the attacker can abuse the sensor node’s identification, because SIDj can be easily known by the attacker. Therefore, the anonymity of sensor nodes needs to be provided. First, SI checks the freshness of TS2. Then, if TS2 is valid, SI computes IDi = DIDGWN ⊕ H(IDi || TCi || TS2) and checks whether H(IDi || TCi || TS2) and the received CGWN are equal.

3.4. DoS Attack

A DoS attack is an attempt to make a machine or network resource unavailable so regular users cannot use the system’s resources. Although the methods, motives, and targets of DoS attacks may vary, they generally involve efforts to temporarily or indefinitely interrupt or suspend the services of a host connected to the Internet. In Jiang et al.’s authentication protocol, sensor nodes can verify the freshness of a message using TS2. Therefore, when an attacker sends a previous message to the sensor node, the sensor node knows whether this message is a current message or a previous message. However, after an attacker gets the previous message \{ TS2, DIDi, DIDGWN, CGWN, Aji \}, the attacker sends the message changing only TS2 to the current timestamp. To check the legitimacy of the message, the sensor node needs to execute various computations, such as hash function (twice), verification function (twice), and timestamp checking (once). The sensor node has limited battery power and computational ability, so it is possible that a sensor node cannot perform its normal functions when an attacker executes a DoS attack on the sensor node.

3.5. Weak ID Anonymity

In Jiang et al.’s authentication protocol, the user can maintain the ID anonymity using DIDi. An attacker cannot compute IDi from DIDi, because the attacker does not know H(Ai || DI). However, IDi can be exposed in the sensor nodes gained by the attacker. The sensor nodes are scattered in various places, so the attacker can find the sensor nodes and obtain their authority. Therefore, the attacker can compute the user’s identity using...
$ID_i = DID_{GW} \oplus H(DID_i \parallel TC_j \parallel TS_2)$, because the sensor nodes know $TC_j$, which is shared in the sensor registration phase. Hence, the attacker can get $ID_i$ after gaining the sensor nodes, and the anonymity of this protocol is not strong.

4. Conclusion
Jiang et al. proposed a privacy-aware two-factor authentication protocol using ECC for WSNs. They insist that their protocol achieves various security and usability features necessary for real-life application environments while maintaining acceptable efficiency. However, this paper analyzed Jiang et al.’s protocol and showed that this protocol has security vulnerabilities, such as a lack of mutual authentication, a risk of SID modification and DoS attacks, a lack of sensor anonymity, and weak ID anonymity. To solve these vulnerabilities, a security-enhanced privacy-aware two-factor authentication protocol using ECC for WSNs needs to be proposed.

Acknowledgements
This work was supported by the National Research Foundation of Korea grant funded by Korea government (Ministry of Science, ICT & Future Planning) (NRF-2017R1C1B5017492) and this research was supported by financial support of Howon University in 2017.

References
[1] Y Choi et al. Security enhanced user authentication protocol for wireless sensor networks using elliptic curves cryptography. Sensors. 2014; 14(6): 10081-10106.
[2] Q. Jiang et al. A privacy-aware two-factor authentication protocol based on elliptic curve cryptography for wireless sensor networks. International Journal of Network Management. 2017; 27(3).
[3] S. Kumari et al. User authentication schemes for wireless sensor networks: A review. Ad Hoc Networks. 2015; 27: 159-194.
[4] M. L. Das. Two-factor user authentication in wireless sensor networks. IEEE Transactions on Wireless Communications. 2009; 8(3):1086-1090.
[5] M. K. Khan, and K. Alghathbar. Cryptanalysis and security improvements of two-factor user authentication in wireless sensor networks. Sensors. 2010; 10(3): 2450-2459.
[6] T. H. Chen, and W. K. Shih. A robust mutual authentication protocol for wireless sensor networks. ETRI journal. 2010; 32(5): 704-712.
[7] D. He et al. An Enhanced Two-factor User Authentication Scheme in Wireless Sensor Networks. Ad hoc & sensor wireless networks. 2010; 10(4), pp361-371.
[8] H. L. Yeh, et al. A secured authentication protocol for wireless sensor networks using elliptic curves cryptography. Sensors. vol 11(5), 2011: 4767-4779.
[9] W. Shi and P. Gong. A new user authentication protocol for wireless sensor networks using elliptic curves cryptography. International Journal of Distributed Sensor Networks. 2013.
[10] D. Hankerson et al. Guide to elliptic curve cryptography. Springer Science & Business Media, 2006.
[11] S. A. Chaudhry et al. An improved and provably secure privacy preserving authentication protocol for SIP. Peer-to-Peer Networking and Applications. 2017; 10(1): 1-15.
[12] D. Wang. Anonymous two-factor authentication in distributed systems: certain goals are beyond attainment. IEEE Transactions on Dependable and Secure computing. 2015; 12(4): 428-442.