Cryptanalysis and Improvement of an ECC-Based Authentication Protocol for Wireless Sensor Networks

Taeui Song[1], Dongwoo Kang[2], Jihyeon Ryu[1], Hyoungshick Kim[3], and Dongho Won[4(✉)]

[1] Department of Platform Software, Sungkyunkwan University, 2066 Seobu-ro, Jangan-gu, Suwon-si, Gyeonggi-do 16419, Korea
{tusong,jhryu}@security.re.kr
[2] Department of Electrical and Computer Engineering, Sungkyunkwan University, 2066 Seobu-ro, Jangan-gu, Suwon-si, Gyeonggi-do 16419, Korea
dwkang@security.re.kr
[3] Department of Software, Sungkyunkwan University, 2066 Seobu-ro, Jangan-gu, Suwon-si, Gyeonggi-do 16419, Korea
hyoung@skku.edu
[4] Department of Computer Engineering, Sungkyunkwan University, 2066 Seobu-ro, Jangan-gu, Suwon-si, Gyeonggi-do 16419, Korea
dhwon@security.re.kr

Abstract. The Internet of Things is the interconnection of devices that exchange collected data with each other through the internet using electronics, software, and sensors. Wireless sensor network is used extensively in implementation of the Internet of Things system. With the increasing use of them, many researchers have focused on the security in wireless sensor network environment. In 2016, Wu et al. proposed a user authentication protocol for wireless sensor network, claiming it was secure from various types of attacks. However, we found out that their scheme has some vulnerabilities to the user impersonation attack, and the denial of service attack. In order to overcome these problems, we review Wu et al.'s protocol and propose an improved protocol based on their protocol. Then, we show that our proposed protocol is more secure than other authentication protocols for wireless sensor network.

Keywords: Authentication · Internet of Things
Elliptic curve cryptography · Wireless sensor network

1 Introduction

The Internet of Things(IoT) means the technology that connects objects to other objects by embedding sensors and communication units in objects. As information and communication technology develops, the IoT is expanding everywhere fast. Nowadays, it is widely used in most fields including home appliances, traffic, construction, and healthcare system. Wireless sensor network(WSN) plays an

O. Gervasi et al. (Eds.): ICCSA 2018, LNCS 10961, pp. 50–61, 2018.
https://doi.org/10.1007/978-3-319-95165-2_4

important role in the IoT by facilitating remote data collection. In general, there are three different kinds of participants in WSN: the sensors, the gateway and users. The sensors which are deployed in some objects and areas collect information from the environment. They have limited power and resources. The gateway acts as a communication bridge between the sensors and users. All sensors are registered in the gateway and the gateway manages the sensors for the security of the network. For this reason, a user who wants to get information from a particular sensor has to register in the gateway where the sensor is registered. After registering in the gateway, the user connects to the gateway and establishes a session key with the sensor through the gateway. In this process, the user must be authenticated to access to the sensor. If the authentication process is successful, the user can obtain information collected by the sensor. Initially, WSN was composed of homogeneous sensors, so there were some difficulties in collecting different kinds of information. However, recently, heterogeneous sensors are used in WSN instead of homogeneous sensors. Because of these sensors, it became possible to gather a variety of information and as a result WSN can be used in various fields. With the increasing use of WSN, security threats to WSN are also growing exponentially year after year. Since there are confidential and sensitive information among the data collected by diverse kinds of sensors, including private and military information, the security of WSN is considered the most important issue. If malicious people steal and misuse critical information, it leads to huge losses. Therefore, in order to keep information safe, access to sensors must be restricted to authorized personnel only. That is to say, all entities must achieve mutual authentication in WSN. For this reason, many researchers have presented several kinds of user authentication protocols for WSN such as RSA-based, smart card-based, and elliptic curve cryptography(ECC)-based protocols.

In 2004, Watro et al. [13] suggested an authentication protocol for wireless sensor network using RSA. Also, Wong et al. [14] proposed a password-based protocol for WSN using hash function in 2006. In 2009, Das [4] found out that an attacker could impersonate sensors in Watro et al.'s protocol and Wong et al.'s protocol was susceptible to the stolen verifier attack and many logged in users with the same log-in id threat. After analyzing Wong et al.'s protocol, Das presented a smart card-based authentication protocol improving Wong et al.'s. Unfortunately, Das's protocol was also shown to be susceptible to the forgery attack and the insider attack later on. In order to fix these problems, Chen and Shih [2], He et al. [5] and Khan and Alghathbar [7] proposed improvements of Das's protocol. However, some security problems were founded in their protocols. For example, Chen and Shih's protocol could not block the replay attack and the forgery attack and He et al.'s protocol could not provide user anonymity as well as mutual authentication. Furthermore, Vaidya et al. [12] found out that Khan and Alghathbar's protocol suffered from the stolen smart card attack, the forgery attack and the node capture attack. Then, they suggested an improved two factor user authentication protocol.

In 2011, Yeh et al. [17] presented the first ECC-based user authentication protocol for WSN but it had some drawbacks including lack of mutual authentication

and forward security. To overcome the vulnerabilities of Yeh et al.'s protocol, Shi and Gong [10] proposed a new user authentication protocol for WSN using ECC in 2013. Later on, Choi et al. [3] pointed out that Shi and Gong's protocol was susceptible to the sensor energy exhausting attack, the session key attack, and the stolen smart card attack. Then they proposed improvements of Shi and Gong's protocol. In 2014, Turkanović et al. [11] presented a user authentication protocol for heterogeneous ad hoc WSN in which a user can access to a sensor directly. Afterward, Amin and Biswas [1] found out that Turkanović et al.'s protocol could not block the stolen smart card attack, the off-line password guessing attack and the user impersonation attack. Moreover, they claimed that Turkanović et al.'s protocol was not appropriate for WSN because the power consumption of the sensor was high in Turkanović et al.'s protocol. In order to solve these vulnerabilities, they presented a new protocol for WSN but it was pointed out that their protocol was also vulnerable to the user, a gateway, and sensor forgery attacks by Wu et al. [16].

In 2014, Hsieh and Leu [6] found out that Vaidya et al.'s protocol was susceptible to the insider attack and the password guessing attack. Also, they proposed an improved protocol based on Vaidya et al.'s. Nevertheless, Hsieh and Leu's protocol still had some problems defending against the off-line guessing attack, the insider attack, the sensor capture attack and the user forgery attack. Hence, Wu et al. [15] suggested a new authentication protocol for WSN and argued that their protocol could overcome the common security problems. However, recently, we found out that Wu et al.'s protocol is not secure against the user forgery attack and the denial of service attack.

In this paper, we review Wu et al.'s protocol and point out that their protocol is vulnerable to the user impersonation attack and the denial of service attack. After illustrating its vulnerabilities, we propose a secure ECC-based authentication protocol for WSN.

The remainder of the paper is organized as follows. First, in Sect. 2, we introduce elliptic curve cryptography which is applied to Wu et al.'s protocol and our protocol. Then, we review Wu et al.'s protocol in Sect. 3 and analyze their protocol in Sect. 4. Our protocol and the security analysis of it are presented in Sects. 5 and 6. Finally, we conclude the paper in Sect. 7.

2 Preliminaries

Before reviewing Wu et al.'s protocol, we explain elliptic curve cryptography which is used in Wu et al.'s and our protocols.

2.1 Elliptic Curve Cryptography

In 1985, Koblitz [8] and Miller [9] suggested the cryptography system using the elliptic curve independently. Although ECC uses a small key size compared to other public key cryptography such as RSA and ElGamal, it provides a similar level of security as them.

The elliptic curve is expressed by the equation $y^2 = x_3 + ax + b \ mod \ p$ over a prime finite field F_p, where $a, b \in F_p$ satisfying $4a^3 + 27b^2 \neq 0 \ mod \ p$. There are three problems related to ECC: Elliptic Curve Discrete Logarithm Problem(ECDLP), Elliptic Curve Computational Diffie-Hellman Problem(ECCDHP), and Elliptic Curve Decisional Diffie-Hellman Problem(ECDDHP).

- ECDLP: Given two points P and Q in G, it is difficult to find $x \in Z_q^*$ such that $Q = xP$, where xP is P added to itself x times using the elliptic curves operation.
- ECCDHP: Given two points xP and yP in G, where $x, y \in Z_q^*$, it is difficult to compute xyP in G.
- ECDDHP: For $x, y, z \in Z_q^*$, given three points xP, yP and zP in G, it is hard to decide whether $zP = xyP$.

3 Review of Wu et al.'s Protocol

There are four phases in Wu et al.'s protocol: initialization, registration, login and authentication and password change. The notations used in this paper are summarized in Table 1.

Table 1. Notations and their meanings

Notation	Meaning
p, q	Large prime numbers
$E(F_p)$	A finite field F_p on the elliptic curve E
G	A subgroup of $E(F_p)$ with order q
P	The generator of G
U_i, ID_i, PW_i	The $i-th$ user with his identity and password
S_j, SID_j	The $j-th$ sensor with its identity
GW, gs	The gateway and its secret key
SK_u, SK_s	The session keys formed by the user and the sensor
$\mathcal{A}$	The attacker
$h(\cdot)$, $h_1(\cdot)$	The hash function
$\oplus$	The exclusive-or operation
$\|$	The concatenation operation

3.1 Initialization

First, GW generates a group G of elliptic curve points on the elliptic curve E. Then, GW chooses a secret key gs and two hash functions.

3.2 Registration

User Registration

1. U_i picks his or her identity ID_i and password PW_i, and generates a random nonce N_1. Next, U_i computes $TP_i = h(N_1 \parallel PW_i)$ and $TI_i = h(N_1 \parallel ID_i)$ and sends $\{TP_i, TI_i, ID_i\}$ to GW through a secure channel.
2. After getting the registration message from U_i, GW computes $PV_i = h(ID_{GW} \parallel gs \parallel TI_i) \oplus TP_i$ and $IV_i = h(TI_i \parallel gs) \oplus TI_i$. Then GW stores ID_i in its database, stores (PV_i, IV_i, P, p, q) into the smart card and sends it to U_i.
3. Finally, U_i stores $NV_i = h(ID_i \parallel PW_i) \oplus N_1$ into the smart card received from GW.

Sensor Registration

1. S_j sends its identity SID_j to GW through a secure channel.
2. GW computes $ss_j = h(SID_j \parallel gs)$ and transmits it to S_j. Then, SID_j and ss_j are stored in S_j.

3.3 Login and Authentication

1. U_i puts his or her smart card in a device and inputs ID_i and PW_i. The smart card calculates $N_1 = NV_i \oplus h(ID_i \parallel PW_i)$, $TI_i = h(N_1 \parallel ID_i)$ and $TP_i = h(N_1 \parallel PW_i)$ using the values stored in it.
2. U_i selects random nonces $\alpha \in [1, q-1]$, N_2 and N_3, and chooses the sensor S_j. Then, the smart card calculates $TI_i^{new} = h(N_2 \parallel ID_i), UC_1 = PV_i \oplus TP_i \oplus N_3$, $UC_2 = \alpha P$, $UC_3 = IV_i \oplus TI_i \oplus TI_i^{new} \oplus h(N_3 \parallel TI_i)$, $UC_4 = h(N_3 \parallel TI_i^{new} \parallel UC_2) \oplus ID_i$ and $UC_5 = h(ID_i \parallel TI_i \parallel TI_i^{new} \parallel SID_j)$. Next, it sends the login request message $LM_1 = \{TI_i, SID_j, UC_1, UC_2, UC_3, UC_4, UC_5\}$ to GW.
3. After getting the login request message from U_i, GW calculates $N_3 = UC_1 \oplus h(ID_{GW} \parallel gs \parallel TI_i)$, $TI_i^{new} = UC_3 \oplus h(TI_i \parallel gs) \oplus h(N_3 \parallel TI_i)$ and $ID_i = UC_4 \oplus h(N_3 \parallel TI_i^{new} \parallel UC_2)$. If ID_i is not in its database or $UC_5 \neq h(ID_i \parallel TI_i \parallel TI_i^{new} \parallel SID_i)$, the process is terminated. If not, GW calculates $ss_j = h(SID_j \parallel gs)$ and $GC_1 = h(TI_i \parallel SID_j \parallel ss_j \parallel UC_2)$. Then it transmits $LM_2 = \{TI_i, SID_j, UC_2, GC_1\}$ to S_j.
4. S_j verifies SID_j and $GC_1 \stackrel{?}{=} h(TI_i \parallel SID_j \parallel ss_j \parallel UC_2)$. If the verification is successful, S_j selects random nonce $\beta \in [1, q-1]$ and calculates $SC_1 = \beta P$, $SC_2 = \beta UC_2$, $SK_s = h_1(UC_2 \parallel SC_1 \parallel SC_2)$, $SC_3 = h(TI_1 \parallel SID_j \parallel SK_s)$ and $SC_4 = h(ss_j \parallel TI_i \parallel SID_j)$. After that, $LM_3 = \{SC_1, SC_3, SC_4\}$ is sent to GW.
5. GW verifies $SC_4 \stackrel{?}{=} h(ss_j \parallel TI_i \parallel SID_j)$. If it is correct, GW calculates $GC_2 = h(ID_{GW} \parallel gs \parallel TI_i^{new}) \oplus h(TI_i^{new} \parallel N_3)$, $GC_3 = h(TI_i^{new} \parallel gs) \oplus h(TI_i \parallel N_3)$ and $GC_4 = h(ID_i \parallel TI_i \parallel TI_i^{new} \parallel SID_j \parallel GC_2 \parallel GC_3 \parallel N_3)$. Finally, $LM_4 = \{SC_1, SC_3, GC_2, GC_3, GC_4\}$ is sent to U_i.

6. After verifying $GC_4 \stackrel{?}{=} h(ID_i \parallel TI_i \parallel TI_i^{new} \parallel SID_j \parallel GC_2 \parallel GC_3 \parallel N_3)$ received from GW, U_i calculates $UC_6 = \alpha SC_1$ and $SK_u = h_1(UC_2 \parallel SC_1 \parallel UC_6)$. Then U_i verifies $SC_4 \stackrel{?}{=} h(TI_i \parallel SID_j \parallel SK_u)$. If it holds, the smart card calculates $NV_i^{new} = N_2 \oplus h(ID_i \parallel PW_i)$, $PV_i^{new} = GC_2 \oplus h(N_2 \parallel PW_i) \oplus h(TI_i^{new} \parallel N_3)$ and $IV_i^{new} = GC_3 \oplus TI_i^{new} \oplus h(TI_i \parallel N_3)$. Lastly, it changes (NV_i, PV_i, IV_i) into $(NV_i^{new}, PV_i^{new}, IV_i^{new})$.

3.4 Password Change

1. U_i puts his or her smart card in a device and enters ID_i and PW_i. Then, the smart card calculates $N_1 = NV_i \oplus h(ID_i \parallel PW_i)$, $TI_i = h(N_1 \parallel ID_i)$ and $TP_i = h(N_1 \parallel PW_i)$.
2. U_i chooses random nonces N_4 and N_5, and computes $TI_i^{new} = h(N_4 \parallel ID_i)$, $UC_7 = PV_i \oplus TP_i \oplus N_5$, $UC_8 = IV_i \oplus TI_i \oplus TI_i^{new} \oplus h(N_5 \parallel TI_i)$, $UC_9 = ID_i \oplus h(N_5 \parallel TI_i^{new})$ and $UC_{10} = h(ID_i \parallel TI_i \parallel TI_i^{new} \parallel N_5)$. After the calculation, U_i sends the message $CM_1 = \{TI_i, UC_7, UC_8, UC_9, UC_{10}\}$ to GW.
3. GW calculates $N_5 = UC_7 \oplus h(ID_{GW} \parallel gs \parallel TI_i)$, $TI_i^{new} = UC_8 \oplus h(TI_i \parallel gs) \oplus h(N_5 \parallel TI_i)$ and $ID_i = UC_9 \oplus h(N_5 \parallel TI_i^{new})$ first. Next, it verifies whether ID_i is in its database and checks $UC_{10} \stackrel{?}{=} h(ID_i \parallel TI_i \parallel TI_i^{new} \parallel N_5)$. If the verification is successful, GW computes $GC_5 = h(ID_{GW} \parallel gs \parallel TI_i^{new}) \oplus h(TI_i^{new} \parallel N_5)$, $GC_6 = h(TI_i^{new} \parallel gs) \oplus h(TI_i \parallel N_5)$ and $GC_7 = h(ID_i \parallel N_5 \parallel TI_i \parallel TI_i^{new} \parallel GC_5 \parallel GC_6)$. Then, $CM_2 = \{GC_5, GC_6, GC_7\}$ is sent to U_i.
4. U_i verifies $GC_7 \stackrel{?}{=} h(ID_i \parallel N_5 \parallel TI_i \parallel TI_i^{new} \parallel GC_5 \parallel GC_6)$. If it holds, U_i can input a new password PW_i^{new}. Next, the smart card calculates $TP_i^{new} = h(N_4 \parallel PW_i^{new})$, $PV_i^{new2} = GC_5 \oplus h(TI_i^{new} \parallel N_5) \oplus TP_i^{new}$, $IV_i^{new2} = GC_6 \oplus h(TI_i \parallel N_5) \oplus TI_i^{new}$ and $NV_i^{new2} = h(ID_i \parallel PW_i^{new}) \oplus N_4$. Finally, (NV_i, PV_i, IV_i) are replaced with $(NV_i^{new2}, PV_i^{new2}, IV_i^{new2})$.

4 Cryptanalysis of Wu et al.'s Protocol

4.1 User Impersonation Attack

In Wu et al.'s protocol, when an attacker $\mathcal{A}$ registers his account, he or she can get the smart card which contains the values of PV_A, IV_A, NV_A, P, p and q. With his or her identity, password and the smart card, $\mathcal{A}$ can impersonate other legal users. We illustrate the process below.

1. An attacker $\mathcal{A}$ gets the values of PV_A, IV_A, NV_A, P, p, and q from his smart card, and computes $N_{A1} = NV_A \oplus h(ID_A \parallel PW_A)$, $TI_A = h(N_{A1} \parallel ID_A)$ and $TP_A = h(N_{A1} \parallel PW_A)$.
2. $\mathcal{A}$ guesses arbitrary identity ID^*.

3. $\mathcal{A}$ selects random nonces $\alpha \in [1, q-1]$, N_{A2}, N_{A3}, and the sensor SID_j which he or she wants to connect, computes $TI_A^{new} = h(N_{A2} \parallel ID_A)$, $UC_{A1} = PV_A \oplus TP_A \oplus N_{A3}, UC_{A2} = \alpha P, UC_{A3} = IV_A \oplus TI_A \oplus TI_A^{new} \oplus h(N_{A3} \parallel TI_A)$, $UC_{A4} = h(N_{A3} \parallel TI_A^{new} \parallel UC_{A2}) \oplus ID^*$ and $UC_{A5} = h(ID_A \parallel TI_A \parallel TI_A^{new} \parallel SID_j)$ and sends $LM_{A1} = \{TI_A, SID_j, UC_{A1}, UC_{A2}, UC_{A3}, UC_{A4}, UC_{A5}\}$.
4. GW computes $N_{A3} = UC_{A1} \oplus h(ID_{GW} \parallel gs \parallel TI_A)$, $TI_A^{new} = UC_{A3} \oplus h(TI_A \parallel gs) \oplus h(N_{A3} \parallel TI_A)$, $ID^* = UC_{A4} \oplus h(N_{A3} \parallel TI_A^{new} \parallel UC_{A2})$ and checks if ID^* is in its database. If there is a match, $\mathcal{A}$ can impersonate the legal user whose identity is ID^*. Although ID^* is different from ID_A which is used to compute TI_A, GW cannot find out it.

4.2 Denial of Service Attack

In Wu et al.'s protocol, a smart card does not check the validity of password entered. That means that even if a user inputs incorrect password, the process continues until GW checks its validity. It leads to the denial of service attack as well as unnecessary waste of resources. The process is illustrated below.

1. An attacker $\mathcal{A}$ puts his or her smart card in a device, enters his identity ID_A and incorrect password PW_A^*, and calculates $N_{A1}^* = NV_A \oplus h(ID_A \parallel PW_A^*)$, $TI_A^* = h(N_{A1}^* \parallel ID_A)$ and $TP_A^* = h(N_{A1}^* \parallel PW_A^*)$.
2. $\mathcal{A}$ selects random nonce $\alpha[1, q-1]$, N_{A2}, and N_3, picks the sensor S_j, computes $TI_A^{new} = h(N_{A2} \parallel ID_A)$, $UC_{A1}^* = PV_i \oplus TP_i^* \oplus N_3$, $UC_{A2} = \alpha P$, $UC_{A3}^* = IV_A \oplus TI_A^* \oplus TI_A^{new} \oplus h(N_{A3} \parallel TI_A^*), UC_{A4}^* = h(N_{A3} \parallel TI_A^{new} \parallel UC_{A2}) \oplus ID^*$ and $UC_{A5}^* = h(ID_A \parallel TI_A^* \parallel TI_A^{new} \parallel SID_j)$, and sends incorrect message $LM_{A1} = \{TI_A^*, SID_j, UC_{A1}^*, UC_{A2}, UC_{A3}^*, UC_{A4}^*, UC_{A5}^*\}$.
3. GW computes $N_{A3}^* = UC_{A1}^* \oplus h(ID_{GW} \parallel gs \parallel TI_A^*)$, $TI_A^{new*} = UC_{A3}^* \oplus h(TI_i^* \parallel gs) \oplus h(N_{A3}^* \parallel TI_A^*)$, $ID_A^* = UC_{A4}^* \oplus h(N_{A3}^* \parallel TI_A^{new*} \parallel UC_{A2})$ and checks if ID_A^* is in its database and $UC_{A5} \stackrel{?}{=} h(ID_A^* \parallel TI_A^* \parallel TI_i^{new*} \parallel SID_j)$. Since UC_{A5}^* does not match with UC_{A5}, GW terminates the process in this phase.

If an attacker $\mathcal{A}$ sends a large of incorrect messages as discussed above, the gateway GW will process the messages over and over. Eventually, it will cause GW to be paralyzed by draining GW's resources.

5 The Proposed Authentication Protocol

To overcome the security drawbacks of Wu et al.'s protocol, we propose an improved protocol based on Wu et al.'s protocol. Our protocol consists of four phases like Wu et al.'s.

5.1 Initialization

This phase is the same as the initialization phase in Wu et al.'s protocol.

5.2 Registration

User Registration

1. U_i picks his or her identity ID_i and password PW_i. After that, U_i selects a random nonce N_1 and calculates $TP_i = h(N_1 \parallel PW_i)$ and $TI_i = h(N_1 \parallel ID_i)$. Then, $\{TP_i, TI_i, ID_i\}$ is sent to GW.
2. GW computes $PV_i = h(ID_{GW} \parallel gs \parallel TI_i) \oplus TP_i$ and $IV_i = h(TI_i \parallel ID_i \parallel gs) \oplus TI_i$, and stores ID_i in its database. Also, GW issues a smart card containing (PV_i, IV_i, P, p, q) and sends it to U_i.
3. After getting the smart card from GW, U_i computes $NV_i = h(ID_i \parallel PW_i) \oplus N_1$ and $V_i = TP_i \oplus TI_i \oplus N_1$, and stores result values into the smart card.

Sensor Registration. There is no difference between this phase and the sensor registration phase in Wu et al.'s protocol.

5.3 Login and Authentication

1. U_i puts his or her smart card in a device and inputs ID_i and PW_i. Then, the smart card computes $N_1 = NV_i \oplus h(ID_i \parallel PW_i)$, $TI_i = h(N_1 \parallel ID_i)$ and $TP_i = h(N_1 \parallel PW_i)$.
2. The smart card verifies $V_i \stackrel{?}{=} TP_i \oplus TI_i \oplus N_1$. If the verification is successful, U_i selects random nonces $\alpha \in [1, q-1]$, N_2, N_3 and the sensor S_j.
3. The smart card computes $TI_i^{new} = h(N_2 \parallel ID_i)$, $UC_1 = PV_i \oplus TP_i \oplus N_3$, $UC_2 = \alpha P$, $UC_3 = IV_i \oplus TI_i \oplus TI_i^{new} \oplus h(N_3 \parallel TI_i)$, $UC_4 = h(N_3 \parallel TI_i^{new} \parallel UC_2) \oplus ID_i$ and $UC_5 = h(ID_i \parallel TI_i \parallel TI_i^{new} \parallel SID_j)$, and sends the login request message $LM_1 = \{TI_i, SID_j, UC_1, UC_2, UC_3, UC_4, UC_5\}$ to GW.
4. GW computes $N_3 = UC_1 \oplus h(ID_{GW} \parallel gs \parallel TI_i)$, $TI_i^{new} = UC_3 \oplus h(TI_i \parallel ID_i \parallel gs) \oplus h(N_3 \parallel TI_i)$ and $ID_i = UC_4 \oplus h(N_3 \parallel TI_i^{new} \parallel UC_2)$. Next, GW checks the validity of ID_i and $UC_5 \stackrel{?}{=} h(ID_i \parallel TI_i \parallel TI_i^{new} \parallel SID_i)$. If it holds, GW calculates $ss_j = h(SID_j \parallel gs)$ and $D_1 = h(TI_i \parallel SID_j \parallel ss_j \parallel UC_2)$ and sends $LM_2 = \{TI_i, SID_j, UC_2, GC_1\}$ to S_j.
5. S_j verifies SID_j and $GC_1 \stackrel{?}{=} h(TI_i \parallel SID_j \parallel ss_j \parallel UC_2)$. If it fails, the process is terminated. Otherwise, S_j picks random nonce $\beta \in [1, q-1]$ and computes $SC_1 = \beta P$, $SC_2 = \beta UC_2$, $SK_s = h_1(UC_2 \parallel SC_1 \parallel SC_2)$, $SC_3 = h(TI_1 \parallel SID_j \parallel SK_s)$ and $SC_4 = h(ss_j \parallel TI_i \parallel SID_j)$. Then, it transmits $LM_3 = \{SC_1, SC_3, SC_4\}$ to GW.
6. GW checks $SC_4 \stackrel{?}{=} h(ss_j \parallel TI_i \parallel SID_j)$. If the verification is successful, GW calculates $GC_2 = h(ID_{GW} \parallel gs \parallel TI_i^{new}) \oplus h(TI_i^{new} \parallel N_3)$, $GC_3 = h(TI_i^{new} \parallel gs) \oplus h(TI_i \parallel N_3)$ and $GC_4 = h(ID_i \parallel TI_i \parallel TI_i^{new} \parallel SID_j \parallel GC_2 \parallel GC_3 \parallel N_3)$. Finally, it sends $LM_4 = \{SC_1, SC_3, GC_2, GC_3, GC_4\}$ to U_i.
7. After getting the message from GW, U_i verifies $GC_4 \stackrel{?}{=} h(ID_i \parallel TI_i \parallel TI_i^{new} \parallel SID_j \parallel GC_2 \parallel GC_3 \parallel N_3)$ first. If it is wrong, the smart card stops the process. If not, the smart card calculates $UC_6 = \alpha SC_1$ and $SK_u = h_1(UC_2 \parallel$

$SC_1 \parallel UC_6$), and verifies $SC_4 \stackrel{?}{=} h(TI_i \parallel SID_j \parallel SK_u)$. If it is successful, the smart card computes $NV_i^{new} = N_2 \oplus h(ID_i \parallel PW_i)$, $PV_i^{new} = GC_2 \oplus h(N_2 \parallel PW_i) \oplus h(TI_i^{new} \parallel N_3)$ and $IV_i^{new} = GC_3 \oplus TI_i^{new} \oplus h(TI_i \parallel N_3)$. Lastly, (NV_i, PV_i, IV_i) are changed into $(NV_i^{new}, PV_i^{new}, IV_i^{new})$.

5.4 Password Change

1. U_i puts his or her smart card in a device and enters ID_i and PW_i. After that, the smart card calculates $N_1 = NV_i \oplus h(ID_i \parallel PW_i)$, $TI_i = h(N_1 \parallel ID_i)$ and $TP_i = h(N_1 \parallel PW_i)$.
2. The smart card computes $TP_i \oplus TI_i \oplus N_1$ and checks if the result value is equal to V_i stored in the smart card. If it is correct, the smart card ask U_i to input a new password PW_i^{new}.
3. After U_i inputs PW_i^{new}, the smart card calculates $TP_i^{new} = h(N_1 \parallel PW_i^{new})$, $PV_i^{new} = PV_i \oplus TP_i \oplus TP_i^{new}$, $NV_i^{new} = h(ID_i \parallel PW_i^{new}) \oplus N_1$ and $V_i^{new} = TP_i^{new} \oplus TI_i \oplus N_1$. Lastly, the smart card changes (PV_i, NV_i, V_i) into $(PV_i^{new}, NV_i^{new}, V_i^{new})$.

6 Cryptanalysis of the Proposed Protocol

In this section, we explain our proposed protocol is secure against various types of attacks. Table 2 shows the comparison of security properties between our protocol and other ECC-based protocols.

Insider attack. In user registration phase, a user submits $TI_i = h(N_1 \parallel PW_i)$ to GW. There is no way that an insider attacker guesses PW_i without knowing the value of N_1. Therefore, our proposed protocol can block the insider attack.

Table 2. The comparison of security properties

Attack and security property	Wu et al.	Shi and Gong	Choi et al.	Ours
Resistant to the insider attack	√	√	√	√
Resistant to the off-line password guessing attack	√	×	×	√
Resistant to the user impersonation attack	×	×	×	√
Resistant to the gateway forgery attack	√	√	√	√
Resistant to the denial of service attack	×	√	√	√
Resistant to the replay attack	√	√	√	√
Resistant to the sensor capture attack	√	√	√	√
Provide user anonymity	√	×	×	√
Provide mutual authentication	√	√	√	√
Resistant to session key leakage	√	√	√	√

Off-line password guessing attack. An attacker $\mathcal{A}$ can get the values of (PV_i, IV_i, NV_i) from U_i's smart card and eavesdrop the messages $\{LM_1^{old}, LM_2^{old}, LM_3^{old}, LM_4^{old}\}$ from the last session. $\mathcal{A}$ guesses ID_i and PW_i and calculates $TI_i^* = h(NV_i \oplus h(ID_i^* \parallel PW_i^*) \parallel ID_i^*)$ and $TP_i^* = h(NV_i \oplus h(ID_i^* \parallel PW_i^*) \parallel PW_i)$ by using the equation $N_1 = NV_i \oplus h(ID_i \parallel PW_i)$. $\mathcal{A}$ can also get the equations $UC_1^{old} = PV_i \oplus h(NV_i \oplus h(ID_i^* \parallel PW_i^*) \parallel PW_i) \oplus N_3$ and $UC_4^{old} = h(N_3 \parallel h(NV_i \oplus h(ID_i^* \parallel PW_i^*) \parallel ID_i^*) \parallel UC_2^{old})$. N_3 is absolutely necessary to get PW_i from the equations that $\mathcal{A}$ obtained. However, $\mathcal{A}$ can get N_3 only if he has the value of gs which is the secret key of the gateway. It is impossible for $\mathcal{A}$ to obtain gs so he or she cannot conduct the off-line password guessing attack.

User impersonation attack. Suppose that $\mathcal{A}$ tries to impersonate legal user using his or her own identity, password and smart card. $\mathcal{A}$ guesses other user's identity ID_i and uses it to calculate $UC_4 = h(N_3 \parallel TI_A^{new} \parallel UC_2) \oplus ID_i$. Also, he or she computes UC_1, UC_2, UC_3 and UC_5 and transmits the login request message to GW. After getting the login request message from $\mathcal{A}$, GW computes $ID_i = UC_4 \oplus h(N_3 \parallel TI_A^{new} \parallel UC_2)$, $TI_A^{new*} = UC_3 \oplus h(TI_A \parallel ID_i \parallel gs) \oplus h(N_3 \parallel TI_A)$. Then, GW checks $UC_5 \stackrel{?}{=} h(ID_i \parallel TI_A \parallel TI_A^{new*} \parallel SID_j)$. However, the verification check fails because $TI_A^{new*} = UC_3 \oplus h(TI_A \parallel ID_i \parallel gs) \oplus h(N_3 \parallel TI_A)$ which is calculated by GW is different from the original $TI_A^{new} = h(TI_A \parallel ID_A \parallel gs)$. It means that the user impersonation attack cannot succeed in our protocol.

Gateway forgery attack. To forge the gateway, $\mathcal{A}$ needs gs because gs is used to compute the values in messages to be sent to SID_j and U_i. However, $\mathcal{A}$ cannot obtain gs so our proposed protocol can block the gateway forgery attack.

Denial of service attack. $\mathcal{A}$ might conduct the denial of service attack by inputting the wrong identity or password and sending the wrong message to the gateway repeatedly. However, in the proposed protocol, the smart card verifies the identity and password entered before transmitting the login request message to the gateway. Therefore, even if $\mathcal{A}$ inputs the wrong identity or password continuously to paralyze the gateway, it never affects the gateway.

Replay attack. Suppose that $\mathcal{A}$ eavesdrops the previous login request message $\{TI_i, SID_j, UC_1, UC_2, UC_3, UC_4, UC_5\}$ and transmits the same login message to the gateway. After that, the gateway computes GC_1 and sends the message M_2 which is the same as the previous M_2. However, the sensor choose a new random nonce β and computes new SC_1 and SC_2 using β. Therefore, although $\mathcal{A}$ conducts replay attack using the previous login message, he or she cannot get the session key unless he or she knows the α which is used to calculate UC_2.

Sensor capture attack. Even if $\mathcal{A}$ gets SID_j and its secret number ss_j, $\mathcal{A}$ cannot obtain secret numbers of other sensors because there is no direct correlation between ss_j and ss_k of other sensor k. It means our protocol can prevent the sensor capture attack.

User anonymity. In our protocol, TI_i is used in the login and authentication phase instead of ID_i. Moreover, it is changed after every authentication phase. Therefore, even if $\mathcal{A}$ gets TI_i, $\mathcal{A}$ cannot get ID_i from TI_i and cannot trace the user's activities.

Mutual authentication. In our proposed protocol, U_i, GW and SID_j can authenticate each other by checking the messages from other party. First, GW verifies the login request message from U_i by checking whether UC_5 is correct. Next, SID_j also verifies the message from GW by checking whether GC_1 is correct. Then, GW checks SC_4 which is sent by SID_j is correct to authenticate SID_j. Finally, U_i authenticates GW by checking GC_4. Through these verification processes, our protocol can provide the mutual authentication.

Session key leakage. Although $\mathcal{A}$ can get the values of UC_2 and SC_1 by eavesdropping the messages between legal entities, $\mathcal{A}$ cannot calculate the session key because it is impossible to obtain SC_2 from UC_2. It means our protocol is secure against session key leakage.

7 Conclusion

In this paper, we reviewed Wu et al.'s ECC-based authentication protocol for WSN and showed that their protocol is vulnerable to the user impersonation attack and the denial of service attack. In order to overcome the security weaknesses of it, we suggested an improved ECC-based authentication protocol. Also, we verified that our proposed protocol can block various types of attacks and it is more secure than other ECC-based authentication protocols by analyzing protocols.

Acknowledgments. This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (NRF-2010-0020210).

References

1. Amin, R., Biswas, G.: A secure light weight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks. Ad Hoc Netw. **36**, 58–80 (2016)
2. Chen, T.H., Shih, W.K.: A robust mutual authentication protocol for wireless sensor networks. ETRI J. **32**(5), 704–712 (2010)
3. Choi, Y., Lee, D., Kim, J., Jung, J., Nam, J., Won, D.: Security enhanced user authentication protocol for wireless sensor networks using elliptic curves cryptography. Sensors **14**(6), 10081–10106 (2014)
4. Das, M.L.: Two-factor user authentication in wireless sensor networks. IEEE Trans. Wirel. Commun. **8**(3), 1086–1090 (2009)
5. He, D., Gao, Y., Chan, S., Chen, C., Bu, J.: An enhanced two-factor user authentication scheme in wireless sensor networks. Ad hoc Sens. Wirel. Netw. **10**(4), 361–371 (2010)

6. Hsieh, W.B., Leu, J.S.: A robust user authentication scheme sing dynamic identity in wireless sensor networks. Wirel. Pers. Commun. **77**(2), 979–989 (2014)
7. Khan, M.K., Alghathbar, K.: Cryptanalysis and security improvements of two-factor user authentication in wireless sensor networks. Sensors **10**(3), 2450–2459 (2010)
8. Koblitz, N.: Elliptic curve cryptosystems. Math. Comput. **48**(177), 203–209 (1987)
9. Miller, V.S.: Use of elliptic curves in cryptography. In: Williams, H.C. (ed.) CRYPTO 1985. LNCS, vol. 218, pp. 417–426. Springer, Heidelberg (1986). https://doi.org/10.1007/3-540-39799-X_31
10. Shi, W., Gong, P.: A new user authentication protocol for wireless sensor networks using elliptic curves cryptography. Int. J. Distrib. Sens. Netw. **9**(4), 730831 (2013)
11. Turkanović, M., Brumen, B., Hölbl, M.: A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion. Ad Hoc Netw. **20**, 96–112 (2014)
12. Vaidya, B., Makrakis, D., Mouftah, H.T.: Improved two-factor user authentication in wireless sensor networks. In: 2010 IEEE 6th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), pp. 600–606. IEEE (2010)
13. Watro, R., Kong, D., Cuti, S.F., Gardiner, C., Lynn, C., Kruus, P.: TinyPK: securing sensor networks with public key technology. In: Proceedings of the 2nd ACM workshop on Security of Ad Hoc and Sensor Networks, pp. 59–64. ACM (2004)
14. 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, 2006, vol. 1, pp. 244–251. IEEE (2006)
15. Wu, F., Xu, L., Kumari, S., Li, X.: A privacy-preserving and provable user authentication scheme for wireless sensor networks based on Internet of Things security. J. Ambient Intell. Humaniz. Comput. **8**(1), 101–116 (2017)
16. Wu, F., Xu, L., Kumari, S., Li, X., Shen, J., Choo, K.K.R., Wazid, M., Das, A.K.: An efficient authentication and key agreement scheme for multi-gateway wireless sensor networks in IoT deployment. J. Netw. Comput. Appl. **89**, 72–85 (2017)
17. Yeh, H.L., Chen, T.H., Liu, P.C., Kim, T.H., Wei, H.W.: A secured authentication protocol for wireless sensor networks using elliptic curves cryptography. Sensors **11**(5), 4767–4779 (2011)