Cryptanalysis and Improvement of a User Authentication Scheme for Internet of Things Using Elliptic Curve Cryptography

Majid Bayat[1], Mohammad Beheshti Atashgah[2], Morteza Barari[2], and Mohammad Reza Aref[3]

(Corresponding author: Majid Bayat)

Department of Computer Engineering, Shahed University, Tehran, Iran[1]

ICT Complex, Malek-Ashtar University of Technology Tehran, Iran[2]

Department of Electrical Engineering, Sharif University of Technology[3]

(Email: mbayat@shahed.ac.ir)

(Received July 10, 2018; Revised and Accepted Dec. 22, 2018; First Online June 16, 2019)

Abstract

The concept of Internet of Things (IoT) is that objects and things via the Internet infrastructure can interconnect into a global dynamic extended network. In order to catch the final goal, IoT takes advantages of other useful technologies like RFIDs, WSNs, M2M communications, big data and cloud computing. Wireless Sensor Networks (WSNs) is one of the main parts of IoT's building blocks which can be used in almost all scopes of the IoT's applications. Because of the importance of the WSN's security, researchers are already working on new and efficient techniques on its different security schemes and protocols such as user authentication schemes. Recently, Wu *et al.* proposed a new user authentication scheme for Internet of Things-based wireless sensor networks. The scheme suggests a new method in which a user of IoT can be authenticated with a sensor node of the WSN through a communication with a gateway. Unfortunately, we have found that Wu *et al.*'s scheme has some security vulnerabilities and is not immune to some security attacks. This paper focuses on eliminating the security vulnerabilities of Wu *et al.*'s scheme by suggesting an enhanced scheme. We introduce a provable security for our scheme and present its formal security analysis by ProVerif. Moreover, we compare the proposed scheme with some other related schemes for WSNs in aspects of efficiency and security.

Keywords: Authentication; Internet of Things; ProVerif; Security; WSN

1 Introduction

The Internet of Things (IoT) is defined as a network of highly connected things and devices. In current perspective, the IoT includes various kinds of things, *e.g.*, sensors, actuators, RFID tags, smart phones or backend servers, which are very different in terms of size, capability and functionality. In other words, Internet of Things uses some technologies such as: Wireless Sensor Networks (WSN), Radio Frequency Identification (RFID), Machine-to-Machine communication, cloud computing and *etc.* According to Gartner's forecast [27], the IoT, which excludes PCs, smart phones and tablets, will grow to more than 26 billion units installed in 2020.

WSNs are crucial for the future of Internet of Things because it covers necessary IoT applications. The WSN contains small, wireless, ad-hoc sensor nodes which are used in a wide range of application scenarios such as health care, smart homes, military, environment and *etc.* [1, 8, 11, 15, 16, 19–21, 25].

Wireless sensor networks include three main parts: the users, the sensors and the gateway. The most important part is the gateway which can communicate with all the sensors. The gateway is accountable for the wireless sensor network security. The sensors and users register on relative gateway. Users who want to use from the data collected by the sensors should contact the gateway. Here, a common method is use of an session encryption key. Constructing a secure session key between the sensor and the user is a basic issue. If a user requests a data from a sensor of WSN, first of all, he/she should be identified for the legitimate access. The usual method is utilizing an authentication scheme among the sensors and users. So, authentication protocols are essential for WSN.

2 Related Works

In recent years, WSNs and their different security mechanisms have attracted many researcher's attention. Due to the limited resource of the WSNs, classic security mechanisms are not applicable because of much energy consumption. Therefore, many lightweight security methods are proposed for WSN (*e.g.* intrusion detection, secure data aggregation, secure and efficient routing protocols, *etc.*) [23, 24, 30, 40].

Watero *et al.* proposed a user authentication protocol for WSN based on RSA in 2004 [34]. But, in 2009, Das showed that Watero *et al.*'s scheme is vulnerable against sensor forgery attack [7]. Moreover, he presented an other efficient authentication protocol that using smart card. But in 2010, his proposed scheme was evaluated by Chen *et al.* [5], He *et al.* [13], Khan *et al.* [17] and Vaidya *et al.* [33], respectively and it became clear that his scheme suffers from several security weaknesses like destitute of mutual authentication, the impersonation attack and the insider attack. Furthermore, Vaidya *et al.* showed that the Khan *et al.*'s scheme was also vulnerable against stolen smart card and the sensor nodes capture attacks and finally, they proposed an improved scheme. In 2011, Kumar *et al.* pointed out that He *et al.* [13] was vulnerable against information leakage attack and their scheme could not satisfy the following security properties: user anonymity, mutual authentication and constructing a shared session key by the sensor and the user [18].

Because of acceptable computational complexity, Elliptic Curve Cryptography (ECC) has been recently used for WSNs [2, 12, 22, 26, 29]. In 2011, Yeh *et al.* [38] showed that the Chen *et al.*'s scheme [5] suffers from the insider attack and lack of a password change phase. They also proposed the first ECC-based authentication scheme for WSNs. But, in 2011, Han [39] pointed out that Yeh *et al.*'s scheme does not satisfy forward security and mutual authentication. In 2013, Shai *et al.* [31] showed a two factor ECC-based authentication scheme. But, in 2014, Choi *et al.* presented that the Shai *et al.*'s scheme is not immune against the known session key attack and the off-line password guessing attack [6]. In addition, they presented a novel scheme. In 2015, Wu *et al.* [35] stated that the Choi *et al.*'s scheme still has some vulnerabilities such as user forgery attacks and off-line password guessing. Additionally, the user identity is revealed in the message and therefore, the privacy of user's identity is not met.

In order to pass the popular attacks, Turkanovic suggested a new scheme for heterogeneous wireless sensor networks in 2014 [32]. But, in 2015, Farash *et al.* [10] and Chang *et al.* [4] independently showed that Turkanović is vulnerable against the off-line password guessing and stolen verifier attacks. Moreover, in their scheme, the identity of the user can be traced.

In 2014, Hsieh *et al.* [14] showed that Vaidya *et al.*'s scheme [33] is vulnerable to off-line password guessing attack and the insider attack. Additionally, they presented a new scheme in their paper.

Wu *et al.* [36] presented a new scheme for WSNs which is based on the Fantacci *et al.* [9] and Nguyen [28] recommendations for IoT security. In this scheme, a user sends messages to a gateway at first and after that the gateway communicates with a sensor. Finally, by the Wu *et al.*'s scheme a user, a gateway and a sensor can authenticate each other.

In this paper, we show that the Wu *et al.*'s scheme is vulnerable to some security weaknesses and to overcome those flaws, we suggest an enhanced authentication scheme.

2.1 Our Contribution

In this paper, we show that the Wu *et al.*'s user authentication scheme [36] is not a secure scheme because it is vulnerable against forgery and Denial of Service (DoS) attacks. After that, in order to eliminate the weaknesses we suggest an enhanced user authentication scheme for IoT. In addition, we present a formal security analysis by ProVerif and a provable security in the random oracle model for our scheme. Finally, we compare the proposed scheme with related schemes in case of security and efficiency. The results indicate that our scheme is a suitable and practical design for utilizing in IoT.

2.2 Paper Organization

The rest of this paper is organized as follows: We review Wu *et al.*'s scheme and its security analysis in Section 2. In Section 3, we introduce our improved scheme. The security analysis of the proposed scheme and some comparisons are posed in Section 4. Finally, we conclude the paper in Section 5.

3 Review of the Wu *et al.*'s Scheme

In this section, we review the Wu *et al.*'s scheme [36]. Their scheme includes four phases: Initialization, Registration, Login and Authentication. Table 1 presents utilized notations of the Wu *et al.*'s scheme.

3.1 Initialization

GW obtains an addition group G with a large prime order q on $E(F_q)$. P is a generator of group G. ID_{GW} is the identity of GW. GW also picks a secret key x and two hash functions $h(\cdot)$ and $h_1(\cdot)$.

3.2 Registration

This phase includes registration procedures for user U_i and sensor S_j.
For U_i:

1) U_i picks a random number r_0, his/her own identity ID_i and a password PW_i. After that, he/she computes $MP_i = h(r_0 \parallel PW_i)$ and $MI_i = h(r_0 \parallel ID_i)$, and sends $\{MP_i, MI_i, ID_i\}$ to GW through a secure channel.

2) GW computes $e_i = h(ID_{GW} \parallel x \parallel MI_i) \oplus MP_i$ and $f_i = h(MI_i \parallel x) \oplus MI_i$. GW injects (e_i, f_i, P, p, q) into the smart card, saves ID_i in the database for auditing, and gives the smart card to U_i by a secure channel.

Table 1: Symbols were used in the Wu *et al.*'s and proposed schemes

Symbols	Description
p, q	Large prime numbers
$E(F_q)$	An elliptic curve E over the finite field F_q
G	An additive subgroup of points of E with order q
P	A generator of G
GW, x	The gateway and its corresponding secret key
U_i, ID_i, PW_i	The i-th user, his/her identity and password
S_j, SID_j	The j-th sensor and its identity
sk_u, sk_s	The session keys computed by the user and the sensor
A	The adversary (malicious)
$h(.), h_1(.)$	One-way hash functions
T_i	Timestamp of user U_i
l	Security parameter of system
$E_k(.)/D_k(.)$	The symmetric encryption/decryption function with key k
$a \oplus b, a\|b$	The XOR operation and the conjuction with string a and b
$a = ?b$	Check whether a equal b

3) U_i saves $d_i = h(ID_i \parallel PW_i) \oplus r_0$ into the smart card.

For S_j:

1) S_j submits SID_j to GW through a secure channel.

2) GW calculates $c_j = h(SID_j \parallel x)$ and sends it to S_j through a secure channel. S_j stores SID_j and c_j.

In addition, if a sensor be substituted by the other sensor one or a new sensor connects the WSN, the new sensor should register to GW similar to the upper steps.

3.3 Login and Authentication

1) U_i inserts his/her card and enters ID_i and PW_i. $r_1 = d_i \oplus h(ID_i \parallel PW_i)$, $MI_i = h(r_1 \parallel ID_i)$ and $\mathrm{MP}_i = h(r_1 \parallel PW_i)$ are computed by the smart card.

2) U_i picks a random number $\alpha \in [1, q-1]$, r_2 and r_3. U_i obtains the sensor S_j as the partner and calculates $MI_i^{new} = h(r_2 \parallel ID_i)$, $B_1 = e_i \oplus MP_i \oplus r_3$, $B_2 = \alpha P$, $B_3 = f_i \oplus MI_i \oplus MI_i^{new} \oplus h(r_3 \parallel MI_i)$, $B_4 = h(r_3 \parallel MI_i^{new} \parallel B_2) \oplus ID_i$ and $B_5 = h(ID_i \parallel MI_i \parallel MI_i^{new} \parallel SID_j)$. Then, he/she sends $M_1 = \{MI_i, SID_j, B_1, B_2, B_3, B_4, B_5\}$ to S_j.

3) GW computes $r_3 = B_1 \oplus h(ID_{GW} \parallel x \parallel MI_i)$, $MI_i^{new} = B_3 \oplus h(MI_i \parallel x) \oplus h(r_3 \parallel MI_i)$ and $ID_i = B_4 \oplus h(r_3 \parallel MI_i^{new} \parallel B_2)$. Then, GW checks if ID_i is in database and $B_5 = ?h(ID_i \parallel MI_i \parallel MI_i^{new} \parallel SID_j)$. If they hold, GW calculates $c_j = h(SID_j \parallel x)$ and $D_1 = h(MI_i \parallel SID_j \parallel c_j \parallel B_2)$. Next, the message $M_2 = \{MI_i, SID_j, B_2, D_1\}$ is sent to sensor S_j.

4) S_j checks SID_j and $D_1 \stackrel{?}{=} h(MI_i \parallel SID_j \parallel c_j \parallel B_2)$. If they are incorrect, S_j fails the session. Otherwise, S_j picks a random $\beta \in [1, q-1]$ and then computes $C_1 = \beta P$, $C_2 = \beta B_2$, $sk_s = h_1(B_2 \parallel C_1 \parallel C_2)$, $C_3 = h(MI_i \parallel SID_j \parallel sk_s)$ and $C_4 = h(c_j \parallel MI_i \parallel SID_j)$. Next, S_j sends $M_3 = \{C_1, C_3, C_4\}$ to GW.

5) GW checks $C_4 \stackrel{?}{=} h(c_j \parallel MI_i \parallel SID_j)$. If it holds, then GW calculates $D_2 = h(ID_{GW} \parallel x \parallel MI_i^{new}) \oplus h(MI_i^{new} \parallel r_3)$, $D_3 = h(MI_i^{new} \parallel x) \oplus h(MI_i \parallel r_3)$ and $D_4 = h(ID_i \parallel MI_i \parallel MI_i^{new} \parallel SID_j \parallel D_2 \parallel D_3 \parallel r_3)$. Finally, GW sends $M_4 = \{C_1, C_3, D_2, D_3, D_4\}$ to U_i.

6) U_i checks $D_4 \stackrel{?}{=} h(ID_i \parallel MI_i \parallel MI_i^{new} \parallel SID_j \parallel D_2 \parallel D_3 \parallel r_3)$. If it holds, U_i computes $B_6 = \alpha C_1$ and $sk_u = h_1(B_2 \parallel C_1 \parallel B_6)$. After that, U_i checks whether $C_4 \stackrel{?}{=} h(MI_i \parallel SID_j \parallel sk_u)$. If it holds, the smart card calculates a new data $d_i^{new} = r_2 \oplus h(ID_i \parallel PW_i)$, $e_i^{new} = D_2 \oplus h(MI_i^{new} \parallel r_3) \oplus h(r_2 \parallel PW_i)$, and $f_i^{new} = D_3 \oplus MI_i^{new} \oplus h(MI_i \parallel r_3)$. Finally, it replaces (d_i, e_i, f_i) with $(d_i^{new}, e_i^{new}, f_i^{new})$, respectively.

3.4 Password Change

1) This step is identical with the step 1 of login and authentication phase.

2) U_i randomly picks values r_4 and r_5 and then computes $MI_i^{new} = h(r_4 \parallel ID_i)$, $B_7 = e_i \oplus MP_i \oplus r_5$, $B_8 = f_i \oplus MI_i \oplus MI_i^{new} \oplus h(r_5 \parallel MI_i)$, $B_9 = ID_i \oplus h(r_5 \parallel MI_i^{new} \parallel B_2)$ and $B_{10} = h(ID_i \parallel MI_i \parallel MI_i^{new} \parallel r_5)$

3) GW calculates $r_5 = B_7 \oplus h(ID_{GW} \parallel x \parallel MI_i)$, $MI_i^{new} = B_8 \oplus h(MI_i \parallel x) \oplus h(r_3 \parallel MI_i)$ and $ID_i = B_9 \oplus h(r_5 \parallel MI_i^{new} \parallel B_2)$, and checks the validity of ID_i and $B_{10} = ?h(ID_i \parallel MI_i \parallel MI_i^{new} \parallel r_5)$. If either of them is failed, the request is rejected. Otherwise, GW computes $D_5 = h(ID_{GW} \parallel x \parallel MI_i^{new}) \oplus h(MI_i^{new} \parallel r_5)$, $D_6 = h(MI_i^{new} \parallel x) \oplus h(MI_i \parallel r_5)$ and $D_7 = h(ID_i \parallel r_5 \parallel MI_i \parallel MI_i^{new} \parallel D_5 \parallel D_6)$. GW sends $M_6 = \{D_5, D_6, D_7\}$ to the user U_i.

4) U_i checks $D_7 \stackrel{?}{=} h(ID_i \parallel r_5 \parallel MI_i \parallel MI_i^{new} \parallel D_5 \parallel D_6)$. If this equation does not hold, U_i fails the session. Otherwise, U_i is asked to input a new password PW_i^{new}. Then, the smart card calculates $MP_i^{new} = h(r_4 \parallel PW_i^{new})$, $e_i^{new2} = D_5 \oplus h(MI_i^{new} \parallel r_5) \oplus MP_i^{new}$, $f_i^{new2} = D_6 \oplus h(MI_i \parallel r_5) \oplus MI_i^{new}$ and $d_i^{new2} = r_4 \oplus h(ID_i \parallel PW_i^{new})$, and finally updates (d_i, e_i, f_i) with $(d_i^{new2}, e_i^{new2}, f_i^{new2})$.

3.5 Security Analysis of Wu *et al.*'s Scheme

In this section, we show that Wu *et al.*'s scheme is vulnerable against two types of attacks: Denial of Service (DoS) attack and forgery attack.

Table 2: Login and Authentication phases of the Wu *et al.*'s scheme

U_i	GW	S_j
Step One: input ID_i, PW_i compute $r_1 = d_i \oplus h(ID_i \parallel PW_i)$ $\mathrm{MI}_i = h(r_1 \parallel ID_i)$ and $\mathrm{MP}_i = h(r_1 \parallel PW_i)$ choose random numbers $\alpha \in [1, q-1]$, r_2 and r_3 compute the followings: $MI_i^{new} = h(r_2 \parallel ID_i)$ $B_1 = e_i \oplus MP_i \oplus r_3$ $B_2 = \alpha P$ $B_3 = f_i \oplus MI_i \oplus MI_i^{new} \oplus h(r_3 \parallel MI_i)$ $B_4 = h(r_3 \parallel MI_i^{new} \parallel B_2) \oplus ID_i$ $B_5 = h(ID_i \parallel MI_i \parallel MI_i^{new} \parallel SID_j)$ $\xrightarrow{M_1=\{MI_i, SID_j, B_1, B_2, B_3, B_4, B_5\}}$		
	Step Two: compute the followings: $r_3 = B_1 \oplus h(ID_{GW} \parallel x \parallel MI_i)$ $MI_i^{new} = B_3 \oplus h(MI_i \parallel x) \oplus h(r_3 \parallel MI_i)$ $ID_i = B_4 \oplus h(r_3 \parallel MI_i^{new} \parallel B_2)$ check: ID_i, $B_5 \stackrel{?}{=} h(ID_i \parallel MI_i \parallel MI_i^{new} \parallel SID_j)$ compute: $c_j = h(SID_j \parallel x)$ $D_1 = h(MI_i \parallel SID_j \parallel c_j \parallel B_2)$ $\xrightarrow{M_2=\{MI_i, SID_j, B_2, D_1\}}$	
		Step Three: check: SID_j check: ID_i, $D_1 \stackrel{?}{=} h(MI_i \parallel SID_j \parallel c_j \parallel B_2)$ choose random $\beta \in [1, q-1]$ compute the followings: $C_1 = \beta P$ $C_2 = \beta B_2$ $sk_s = h_1(B_2 \parallel C_1 \parallel C_2)$ $C_3 = h(MI_i \parallel SID_j \parallel sk_s)$ $C_4 = h(c_j \parallel MI_i \parallel SID_j)$ $\xleftarrow{M_3=\{C_1, C_3, C_4\}}$
	Step Four: check: $C_4 \stackrel{?}{=} h(c_j \parallel MI_i \parallel SID_j)$ compute the followings: $D_2 = h(ID_{GW} \parallel x \parallel MI_i^{new}) \oplus h(MI_i^{new} \parallel r_3)$ $D_3 = h(MI_i^{new} \parallel x) \oplus h(MI_i \parallel r_3)$ $D_4 = h(ID_i \parallel MI_i \parallel MI_i^{new} \parallel SID_j \parallel D_2 \parallel D_3 \parallel r_3)$ $\xleftarrow{M_4=\{C_1, C_3, D_2, D_3, D_4\}}$	
Step Five: check: $D_4 \stackrel{?}{=} h(ID_i \parallel MI_i \parallel MI_i^{new} \parallel SID_j \parallel D_2 \parallel D_3 \parallel r_3)$ compute the followings: $B_6 = \alpha C_1$ $sk_u = h_1(B_2 \parallel C_1 \parallel B_6)$ check: $C_4 \stackrel{?}{=} h(MI_i \parallel SID_j \parallel sk_u)$ compute: $d_i^{new} = r_2 \oplus h(ID_i \parallel PW_i)$ $e_i^{new} = D_2 \oplus h(MI_i^{new} \parallel r_3) \oplus h(r_2 \parallel PW_i)$ $f_i^{new} = D_3 \oplus MI_i^{new} \oplus h(MI_i \parallel r_3)$ replace (d_i, e_i, f_i) with $(d_i^{new}, e_i^{new}, f_i^{new})$		

- **Denial of service attack:** An attacker can masquerade himself/herself as a real user U_i and apply DoS attack against server GW. Since the term $M_1 = \{MI_i, SID_j, B_1, B_2, B_3, B_4, B_5\}$ is always valid, an attacker can apply DoS attack by sending this message to the GW. Note that $M_1 = \{MI_i, SID_j, B_1, B_2, B_3, B_4, B_5\}$ does not contain any fresh term like a time stamp, the attacker can frequently send M_i to the GW and finally, this action allows the server GW to be unavailable. Moreover, the attacker can provide DoS attack more effectively by using Distributed Denial of Service (DDoS) attack.

- **Forgery attack:** Although Wu *et al.*'s stated that their proposed scheme is immune to user forgery attack, but we show that an adversary can play the role of a user U_i and a sensor S_j and consequently GW is convinced that U_i and S_j established a secure session key.

  The adversary records all messages M_1, M_2, M_3 and M_4 of a successful session between the U_i, S_j and GW. After that, the adversary starts a new session and sends the recorded $M_1 = \{MI_i,\ SID_j,\ B_1,\ B_2,\ B_3, B_4, B_5\}$ to server GW. Upon receiving M_1, GW executes its computations and verifications and sends generated M_2 to sensor S_j.

  The adversary intercepts M_2, chooses a random number β' and computes the following parameters:

$$\begin{aligned} C_1' &= \beta' P \\ C_2' &= \beta' B_2 \end{aligned}$$

  The attacker computes a new valid session key sk_s' and the value C_3' as follows:

$$\begin{aligned} sk_s' &= h_1(B_2 \parallel C_1' \parallel C_2') \\ C_3' &= h(MI_i \parallel SID_j \parallel sk_s') \end{aligned}$$

  The adversary uses the recorded value C_4 of the previous session and sends a new message M_3' to GW instead of sensor S_j.

$$M_3' = \{C_1', C_3', C_4\}$$

  Upon receiving M_3', GW verifies the value C_4 and accepts it as a valid value. GW generates the message M_4 and sends it to U_i. Therefore, the adversary can forge U_i and S_j and convince GW that S_j and U_i established a secure session key with each other.

  The proposed attack is arisen of two weaknesses. First, a valid submitted message M_1 in a session, is a valid message for GW at next sessions and second issue is that GW does not utilize a random number in its computations.

4 The Proposed Scheme

In this section, we propose a new scheme that solves the security problems of Wu *et al.*'s scheme. Like Wu *et al.*'s scheme, our new scheme includes four phases: Initialization, Registration, Login and Authentication, and Password change.

4.1 Initialization

GW firstly generates an addition group G with a large prime order q on $E(F_q)$. P is a generator of group G. ID_{GW} is the identity of GW. GW also picks a secret key x and two hash functions $h(\cdot)$ and $h_1(\cdot)$.

4.2 Registration

This phase includes registration procedures for user U_i and sensor S_j.
For U_i:

1) U_i chooses a number r_0 at random, his/her own identity ID_i and a password PW_i. After that, he/she computes the followings:

$$\begin{aligned} MP_i &= h(r_0 \parallel PW_i) \\ MI_i &= h(r_0 \parallel ID_i) \end{aligned} \quad (1)$$

   and then sends $\{MP_i, MI_i, ID_i\}$ to GW via a secure channel.

2) GW computes

$$e_i = h(ID_{GW} \parallel x \parallel MI_i) \oplus MP_i \quad (2)$$
$$f_i = h(MI_i \parallel x) \oplus MI_i \quad (3)$$

   Then, GW injects (e_i, f_i, P, p, q) into the smart card, saves ID_i in the database for auditing, and gives the card to U_i through a secure channel.

3) U_i saves the following d_i into the relative smart card.

$$d_i = h(ID_i \parallel PW_i) \oplus r_0$$

For S_j:

1) S_j submits SID_j to GW via a secure channel.

2) GW calculates $c_j = h(SID_j \parallel x)$ and sends it to S_j through a secure channel. Moreover, S_j stores the parameters SID_j and c_j.

4.3 Login and Authentication

1) U_i inserts his/her smart card and enters ID_i and PW_i. The card computes

$$\begin{aligned} r_1 &= d_i \oplus h(ID_i \parallel PW_i) \\ MI_i &= h(r_1 \parallel ID_i) \\ MP_i &= h(r_1 \parallel PW_i) \end{aligned}$$

2) U_i chooses random numbers $\alpha \in [1, q-1]$, r_2 and r_3, selects sensor S_j as the partner, obtains a time stamp T_i and calculates

$$\begin{aligned} MI_i^{new} &= h\left(r_2 \parallel ID_i\right) \\ B_1 &= e_i \oplus MP_i \oplus r_3 \\ B_2 &= \alpha P \\ B_3 &= f_i \oplus MI_i \oplus MI_i^{new} \oplus h\left(r_3 \parallel MI_i\right) \\ B_4 &= h\left(r_3 \parallel MI_i^{new} \parallel B_2\right) \oplus ID_i \\ B_5 &= h(ID_i \parallel MI_i \parallel MI_i^{new} \parallel SID_j \parallel T_i) \end{aligned}$$

Then, he/she sends M_1 to GW.

$$M_1 = \{MI_i, SID_j, B_1, B_2, B_3, B_4, B_5, T_i\}$$

3) GW checks whether $|T - T_i| < \Delta$, where T is current time and Δ is a predefined delay. If $|T - T_i| > \Delta$, GW rejects the session. If T_i is accepted, GW computes

$$\begin{aligned} r_3 &= B_1 \oplus h(ID_{GW} \parallel x \parallel MI_i) \\ MI_i^{new} &= B_3 \oplus h(MI_i \parallel x) \oplus h(r_3 \parallel MI_i) \\ ID_i &= B_4 \oplus h\left(r_3 \parallel MI_i^{new} \parallel B_2\right) \end{aligned}$$

Then, GW checks if ID_i is in database and $B_5 \stackrel{?}{=} h(ID_i \parallel MI_i \parallel MI_i^{new} \parallel SID_j \parallel T_i)$. If one of the verifications fails, the session is rejected. GW picks $\lambda \in [1, q-1]$ at random, obtains a time stamp T_G and calculates

$$\begin{aligned} C_0 &= \lambda P \\ c_j &= h\left(SID_j \parallel x\right) \\ D_1 &= h\left(MI_i \parallel SID_j \parallel c_j C_0 \parallel B_2 \parallel T_G\right) \end{aligned}$$

Next, the message M_2 is sent to sensor S_j.

$$M_2 = \{MI_i, SID_j, B_2, D_1, C_0, T_G\}$$

4) S_j checks SID_j, $|T - T_G| > \Delta$ and $D_1 \stackrel{?}{=} h(MI_i \parallel SID_j \parallel c_j C_0 \parallel B_2 \parallel T_G)$. If either checking fails, S_j rejects the session. Otherwise, S_j chooses a random $\beta \in [1, q-1]$ and computes

$$\begin{aligned} C_1 &= \beta P \\ C_2 &= \beta B_2 \\ sk_s &= h_1\left(B_2 \parallel C_1 \parallel C_2\right) \\ C_3 &= h\left(MI_i \parallel SID_j \parallel sk_s\right) \\ C_4 &= h\left(c_j C_0 \parallel MI_i \parallel SID_j\right) \end{aligned}$$

Next, S_j sends M_3 to GW.

$$M_3 = \{C_1, C_3, C_4\}$$

5) After receiving M_3, GW checks $C_4 \stackrel{?}{=} h(c_j C_0 \parallel MI_i \parallel SID_j)$. If it holds, GW computes

$$\begin{aligned} D_2 &= h(ID_{GW} \parallel x \parallel MI_i^{new}) \oplus h\left(MI_i^{new} \parallel r_3\right) \\ D_3 &= h\left(MI_i^{new} \parallel x\right) \oplus h\left(MI_i \parallel r_3\right) \\ D_4 &= h(ID_i \parallel MI_i \parallel MI_i^{new} \parallel SID_j \parallel D_2 \parallel D_3 \parallel r_3) \end{aligned}$$

Finally, GW sends M_4 to U_i.

$$M_4 = \{C_1, C_3, D_2, D_3, D_4\}$$

6) Upon receiving M_4, U_i checks $D_4 \stackrel{?}{=} h(ID_i \parallel MI_i \parallel MI_i^{new} \parallel SID_j \parallel D_2 \parallel D_3 \parallel r_3)$. If it is true, U_i computes

$$\begin{aligned} B_6 &= \alpha C_1 \\ sk_u &= h_1\left(B_2 \parallel C_1 \parallel B_6\right) \end{aligned}$$

After that, U_i checks $C_4 \stackrel{?}{=} h\left(MI_i \parallel SID_j \parallel sk_u\right)$. If it holds, the smart card calculates new data as follows

$$\begin{aligned} d_i^{new} &= r_2 \oplus h\left(ID_i \parallel PW_i\right) \\ e_i^{new} &= D_2 \oplus h\left(MI_i^{new} \parallel r_3\right) \oplus h(r_2 \parallel PW_i) \\ f_i^{new} &= D_3 \oplus MI_i^{new} \oplus h\left(MI_i \parallel r_3\right) \end{aligned}$$

Finally, it replaces (d_i, e_i, f_i) with $(d_i^{new}, e_i^{new}, f_i^{new})$, respectively. Table 3 presents the login and authentication phase.

4.4 Password Change

1) This step is identical with the Step 1 of login and authentication phase.

2) U_i randomly chooses values r_4 and r_5 and calculates the followings

$$\begin{aligned} MI_i^{new} &= h\left(r_4 \parallel ID_i\right) \\ B_7 &= e_i \oplus MP_i \oplus r_5 \\ B_8 &= f_i \oplus MI_i \oplus MI_i^{new} \oplus h\left(r_5 \parallel MI_i\right) \\ B_9 &= ID_i \oplus h\left(r_5 \parallel MI_i^{new} \parallel B_2\right) \\ B_{10} &= h(ID_i \parallel MI_i \parallel MI_i^{new} \parallel r_5) \end{aligned}$$

U_i sends $M_5 = \{M_i, B_7, B_8, B_9, B_{10}\}$ and a password change request to GW.

3) Upon receiving M_5 and the password change request, GW calculates

$$\begin{aligned} r_5 &= B_7 \oplus h(ID_{GW} \parallel x \parallel MI_i) \\ MI_i^{new} &= B_8 \oplus h\left(MI_i \parallel x\right) \oplus h\left(r_5 \parallel MI_i\right) \\ ID_i &= B_9 \oplus h\left(r_5 \parallel MI_i^{new} \parallel B_2\right) \end{aligned}$$

and then checks the validity of ID_i and also checks the following:

$$B_{10} =? h\left(ID_i \parallel MI_i \parallel MI_i^{new} \parallel r_5\right)$$

If either of them fails, the request is rejected. Otherwise, GW computes

$$\begin{aligned} D_5 &= h(ID_{GW} \parallel x \parallel MI_i^{new}) \oplus h(MI_i^{new} \parallel r_5) \\ D_6 &= h\left(MI_i^{new} \parallel x\right) \oplus h\left(MI_i \parallel r_5\right) \\ D_7 &= h(ID_i \parallel r_5 \parallel MI_i \parallel MI_i^{new} \parallel D_5 \parallel D_6) \end{aligned}$$

GW sends $M_6 = \{D_5, D_6, D_7\}$ to the user U_i with grant.

4) After receiving M_6, U_i checks $D_7 = ?h(ID_i \parallel r_5 \parallel MI_i \parallel MI_i^{new} \parallel D_5 \parallel D_6)$. If this equation is rejected, U_i fails the session. Otherwise, U_i is requested to input a new password PW_i^{new}. Then, the following values are computed by the smart card:

$$MP_i^{new} = h\left(r_4 \parallel PW_i^{new}\right)$$
$$e_i^{new2} = D_5 \oplus h\left(MI_i^{new} \parallel r_5\right) \oplus MP_i^{new}$$
$$f_i^{new2} = D_6 \oplus h\left(MI_i \parallel r_5\right) \oplus MI_i^{new}$$
$$d_i^{new2} = r_4 \oplus h\left(ID_i \parallel PW_i^{new}\right)$$

Finally U_i, updates (d_i, e_i, f_i) with (d_i^{new2}, e_i^{new2}, f_i^{new2}), respectively.

5 Security Analysis

In this section, we evaluate the security of our scheme. We discuss the security properties of the proposed scheme and present a provable security of our scheme. In addition, a formal proof of the proposed scheme is introduced. Finally security and efficiency comparisons are posed.

5.1 Analysis of the Security Properties

- **Resistant to insider attack:** Within registration phase, U_i sends $MP_i = h\left(r_0 \parallel PW_i\right)$ to GW. The adversary is incapable to guess the correct password PW_i because the adversary has not the random r_0. Thus a malicious GW cannot obtain the password of users.

- **Resistant to off-line password guessing attack:** Assume an adversary A is eavesdropping the communications between U_i and GW to obtain the password PW_i. The adversary records message M_1 (**??**) and try to find the password. Since the password is not contained at the M_1, the adversary is unable to find PW_i. In addition, let the adversary steels the smart card and obtains e_i, f_i and d_i. Since the adversary has not r_0 and the secret value x, it cannot find the passwords via e_i and d_i. Thus the proposed protocol is immune to off-line password guessing attack.

- **Resistant to user forgery attack:** In order to forge U_i, the adversary A should generate a valid message M_1. Since A does not know x, it is unable to calculate valid values $B_1 = h\left(ID_{GW} \parallel x \parallel MI_i\right) \oplus r_3$ and $B_3 = h\left(MI_i \parallel x\right) \oplus MI_i^{new} \oplus h\left(r_3 \parallel MI_i\right)$. In addition, due to the used time stamp, the adversary cannot utilize an old message M_1 to forge U_i. Thus the proposed protocol is secure against user forgery attack.

- **Resistant to gateway forgery attack:** If the adversary A wants to forge GW, it should compute $D_1(20), D_2(28), D_3(29)$ and $r_3(15)$ correctly. Since A has not the secret value x, it is incapable to generates the needed values. Therefore, A is unable to forge GW in our scheme.

- **Resisitant to sensor capture attack:** Sensor capturing attack leads that using retrieved information from compromise sensor node to execute attacks in IoT environment. Adversary attempts to retrieve information about other sensor nodes, and the users in order to compromise any other secure communication between the users and the non-compromised sensor nodes in the IoT. In our scheme, each sensor has a unique identity SID_j and the corresponding secret value c_j. Thus, compromising a sensor does not affect on the other sensors.

- **Resistant to de-synchronization attack:** It implies that the legitimate user's login and authentication is rejected by the gateway. In the proposed scheme, the gateway checks the password in a session before password changing. This avoids inserting wrong passwords. Moreover, inappropriate data between the user and the gateway causes this attack. The gateway only saves the identity for audit and it does not store any data about the users. Data is changed on the user side. It is infeasible that inappropriate data become visible between the gateway and the user. Thus, the proposed scheme is immune to the de-synchronization attack.

- **Resistant to replay attack:** Due to the utilized random fresh numbers by user, gateway and sensor and usage of time stamp, our protocol is immune against reply attack.

- **Resistant to known-key attack:** In our scheme, the session key is $sk_s = h_1\left(B_2 \parallel C_1 \parallel C_2\right)$, where $C_2 = \beta B_2 = \alpha C_1$. Since β and α are randomly selected at each session, the session keys are completely independent. Thus, if A can obtain a session key, it cannot calculates the next session keys.

- **User anonymity:** The proposed protocol utilizes a pseudonym MI_i as the identity of U_i and it be updated in each authentication and password change phase. Therefore, the adversary cannot trace U_i via MI_i. In addition, MI_i does nor reveal ID_i because it is a hash result of ID_i and r_1. Thus, our scheme satisfies the anonymity property for user U_i.

- **Strong forward secrecy:** Assume the adversary who records the flows of previous sessions, obtains all secret information of U_i, S_j and GW. By assuming the intractability ECCDH problem, it cannot compute the the random values α (10) and β (22) and the session key of previous sessions. Thus, the proposed scheme satisfies strong forward secrecy.

Table 3: Login and Authentication phases of the proposed scheme

U_i	GW	S_j
Step 1: input ID_i, PW_i compute $r_1 = d_i \oplus h(ID_i \parallel PW_i)$ $\mathrm{MI}_i = h(r_1 \parallel ID_i)$ and $\mathrm{MP}_i = h(r_1 \parallel PW_i)$ choose random numbers $\alpha \in [1, q-1]$, r_2 and r_3 compute the followings: $MI_i^{new} = h(r_2 \parallel ID_i)$ $B_1 = e_i \oplus MP_i \oplus r_3$ $B_2 = \alpha P$ $B_3 = f_i \oplus MI_i \oplus MI_i^{new} \oplus h(r_3 \parallel MI_i)$ $B_4 = h(r_3 \parallel MI_i^{new} \parallel B_2) \oplus ID_i$ $B_5 = h(ID_i \parallel MI_i \parallel MI_i^{new} \parallel SID_j \parallel T_i)$ $\xrightarrow{M_1=\{MI_i, SID_j, B_1, B_2, B_3, B_4, B_5, T_i\}}$		
	Step 2: Verify T_i and compute the followings: $r_3 = B_1 \oplus h(ID_{GW} \parallel x \parallel MI_i)$ $MI_i^{new} = B_3 \oplus h(MI_i \parallel x) \oplus$ $h(r_3 \parallel MI_i)$ $ID_i = B_4 \oplus h(r_3 \parallel MI_i^{new} \parallel B_2)$ check: ID_i, $B_5 =? h(ID_i \parallel MI_i \parallel MI_i^{new} \parallel SID_j \parallel T_i)$ choose $\lambda \in [1, q-1]$ compute: $C_0 = \lambda P$ $c_j = h(SID_j \parallel x)$ $D_1 = h(MI_i \parallel SID_j \parallel c_j C_0 \parallel B_2 \parallel T_G)$ $\xrightarrow{M_2=\{MI_i, SID_j, B_2, D_1, C_0, T_G\}}$	
		Step 3: check T_G check SID_j check ID_i, $D_1 =? h(MI_i \parallel SID_j \parallel c_j C_0 \parallel B_2 \parallel T_G)$ choose random $\beta \in [1, q-1]$ compute the followings: $C_1 = \beta P$ $C_2 = \beta B_2$ $sk_s = h_1(B_2 \parallel C_1 \parallel C_2)$ $C_3 = h(MI_i \parallel SID_j \parallel sk_s)$ $C_4 = h(c_j C_0 \parallel MI_i \parallel SID_j)$ $\xleftarrow{M_3=\{C_1, C_3, C_4\}}$
	Step 4: check: $C_4 =? h(c_j C_0 \parallel MI_i \parallel SID_j)$ compute the followings: $D_2 = h(ID_{GW} \parallel x \parallel MI_i^{new}) \oplus h(MI_i^{new} \parallel r_3)$ $D_3 = h(MI_i^{new} \parallel x) \oplus h(MI_i \parallel r_3)$ $D_4 = h(ID_i \parallel MI_i \parallel MI_i^{new} \parallel SID_j \parallel D_2 \parallel D_3 \parallel r_3)$ $\xleftarrow{M_4=\{C_1, C_3, D_2, D_3, D_4\}}$	
Step 5: check: $D_4 =? h(ID_i \parallel MI_i \parallel MI_i^{new} \parallel SID_j \parallel D_2 \parallel D_3 \parallel r_3)$ compute the followings: $B_6 = \alpha C_1$ $sk_u = h_1(B_2 \parallel C_1 \parallel B_6)$ check: $C_4 =? h(MI_i \parallel SID_j \parallel sk_u)$ compute: $d_i^{new} = r_2 \oplus h(ID_i \parallel PW_i)$ $e_i^{new} = D_2 \oplus h(MI_i^{new} \parallel r_3) \oplus h(r_2 \parallel PW_i)$ $f_i^{new} = D_3 \oplus MI_i^{new} \oplus h(MI_i \parallel r_3)$ replace (d_i, e_i, f_i) with $(d_i^{new}, e_i^{new}, f_i^{new})$		

5.2 Provable Security

This section introduces the formal proof of our scheme based on the Bresson *et al.*'s model [3]. In the presented proof, The protocol P includes three entities; one user U, one sensor S and a gateway GW. The notation I is used for denoting different users.

We utilize U^i as the $i-th$ instance of U. GW^t, S^j and I^k can similarly be used. We assume a simulator and an oracle to answer to inquired messages. The oracles outputs three states: Accept, reject and $\perp$. If the oracle U^i or S^j is accepted and computes a session key, the following notations are determined; an identity for session (sid_{U^i} *or* sid_{S^j}), an identity for the partner (pid_{U^i} *or* pid_{S^j}) and the session keys (sk_{U^i} *or* sk_{S^j}).

Initialization is done before the simulation. U has the identity ID, password PW and a smart card containing d, e, f, P, q and p. PW is selected of a set with size N. S has parameters c, P, p, q and an identity SID. GW is assigned with an identity ID_{GW} and values x, P, q and p. Moreover, the adversary A knows $ID, SID, ID_{GW}, P, q, p$. In addition, the following definition is used in the simulation:

- **Partnering:** U^i and S^j are partners if a session key is established between them. Beside constructing the session key, four conditions should be satisfied; U^i and S^j are accepted; $sid_{U^i} = sid_{S^j}, pid_{S^j} = U^i, pid_{U^i} = S^j, \cdots, sk_{U^i} = sk_{S^j}$.
- $sfs-fresh$**:** I^k reaches $sfs-fresh$ if the below events are not occurred:
  1) $Reveal(I^k)$
  2) $Reveal(Pid_{I^k})$
  3) Any $Corrupt(I^m)$ query before the $Test$ query, where m is a legitimate participant, containing k.
- $sfs-ake$ *security***:** if A has the advantage on guessing the coin a on P after $Test(I^k)$ where I^k is $sfs-fresh$ and A guesses a bit a', the advantage is defined as

$$Adv_P^{sfs-ake}(A) = 2Pr[a = a'] - 1$$

  A scheme is "$sfs-ake$"$-$secure if $Adv_P^{sfs-ake}(A)$ be a negligible value.

Now, in the form of following theorem, we give the formal proof of our new scheme.

Theorem 1. *The adversary $\mathcal{A}$ can make at most q_s, q_e and q_h queries from Send, Execute and Hash oracles, respectively. $\mathcal{A}$ has the following advantage:*

$$\begin{aligned} Adv_P^{sfs-ake}(A) \quad \leq \quad & \frac{(q_s+q_e)^2}{q-1} + \frac{q_h^2 + (q_s+q_e)^2}{2^l} \\ & + \frac{12q_h + 7q_s}{2^{l-1}} + \frac{2q_s}{N} + 4q_s((q_s+q_e)^2 \\ & + 1)Adv_A^{ECGDH}(t + (2q_s + 4q_e)T_s) \end{aligned}$$

Which in the above equation, $\mathcal{P}$ denotes the scheme, G is a cyclic addition group in the field of $E(F_q)$ that has a prime order q and the passwords are chosen from a set with N elements. Additionally, l denotes the length of security parameter. We consider T_m as the needed time for a scalar multiplication in group G.

Proof. The proposed proof of theorem includes of a some related games from the game G_0 to the game G_8. In the test session of the game G_i, the adversary $\mathcal{A}$ guesses the coin a that is denoted by $Succ_i$. Since there is only one user in the proof procedure, there is no need for $\mathcal{A}$ to take time in guessing the user's identity.

- Game G_0: This game simulates the real attacks with random oracles. If one of the following items happens, a random bit like a is selected instead of the answer of $Test$.
  - When the game aborts or stops, $\mathcal{A}$ does not guess.
  - $\mathcal{A}$ makes more queries than the predetermined quantities.
  - $\mathcal{A}$ utilizes more time than the predetermined time.

    In accordance with the upper definition, we have:

$$Adv_P^{sfs-ake}(A) = 2Pr[Succ_0] - 1$$

- Game G_1: In this game, all oracles should be simulated. We also define three lists which the answers to relative queries are stored in them. L_h-list stores the answers to hash queries. If $\mathcal{A}$ asks a hash query, the answer will be stored in L_A-list and the transcripts of all messages are stored in the L_P-list. In order to break the privacy of authentication processes and to obtain the session keys, the adversary $\mathcal{A}$ can make queries to oracles. Then $Pr[Succ_1] = Pr[Succ_0]$ and so, G_0 and G_1 are indistinguishable.
- Game G_2: In this stage, we want to avoid the collisions in the messages. Using the birthday paradox, we introduce the three following collisions:
  - In different sessions, it is possible that the random numbers $\alpha, \beta \in [1, q-1]$ to be used for the same. Note that, in this case, the total probability will be bounded by $\frac{(q_s+q_e)^2}{2(q-1)}$.
  - The three random numbers r_1, r_2 and r_3 may have collisions. The total probability will be $\frac{(q_s+q_e)^2}{2^{l+1}}$.
  - The upper bound of the probility of collisions in hash functions is $\frac{q_h^2}{2^{l+1}}$.

    Finally, we can find that $|Pr[Succ_2] - Pr[Succ_1]| \leq \frac{(q_s+q_e)^2}{2(q-1)} + \frac{(q_s+q_e)^2+q_h^2}{2^{l+1}}$.

- Game G_3: During this game, we want to find the probability of forging M_1 without random oracles. Since the simulator $\mathcal{B}$ answers as S, we can add steps to $Send\left(U^i, GW^t, M_1\right)$: the simulator $\mathcal{B}$ needs to check if $M_1 \in L_P - list$ and $(ID \parallel *, *)$, $(* \parallel ID, MI)$, $(* \parallel MI, *)$, $(* \parallel ID, *)$, $(* \parallel B_2, *)$ and $(ID \parallel MI \parallel * \parallel SID, B_5)$ are in L_A-list. If any of these parameters fails, the relative query will be terminated. Since S does not password PW or MI^{new}, $(r_1 \parallel PW, *)$ cannot be exterminated. The probabilities for $(* \parallel ID, MI)$ and $(ID \parallel MI \parallel * \parallel SID, B_5)$ are all bounded by $\frac{q_e}{2^l}$ and other parameters are bounded by $\frac{q_h}{2^l}$. Finally, we can see that $|Pr\left[Succ_3\right] - Pr\left[Succ_2\right]| \leq \frac{(5q_h+2q_s)}{2^l}$.

- Game G_4: In this game, we want to find the probibility of forging M_2 without random oracles. we can add steps to $Send(GW^t, S^j, M_2)$: the simulator $\mathcal{B}$ needs to check if $M_2 \in L_P - list$ and $(SID \parallel *, c)$, $(MI \parallel SID \parallel c \parallel B_2, D_1)$ are in $L_A - list$. The probabilities for $(MI \parallel SID \parallel c \parallel B_2, D_1)$ is bounded by $\frac{q_s}{2^l}$ while for $(SID \parallel *, c)$, this bound is equal to $\frac{q_h}{2^l}$. Therefore, we can see that $|Pr\left[Succ_4\right] - Pr\left[Succ_3\right]| \leq \frac{(q_h+q_s)}{2^l}$.

- Game G_5: During this game, we find the probibility of forging M_3 without random oracles. we can add steps to $Send(GW^t, S^j, M_3)$: the simulator $\mathcal{B}$ needs to check if $M_3 \in L_P - list$ and $(1, B_2 \parallel C_1 \parallel *, *)$, $(MI \parallel SID \parallel * \parallel C_3)$ and $(c \parallel MI \parallel SID \parallel C_4)$ are in $L_A - list$. The probabilities for $(MI \parallel SID \parallel * \parallel C_3)$ and $(c \parallel MI \parallel SID \parallel C_4)$ are bounded by $\frac{q_s}{2^l}$ and for $(1, B_2 \parallel C_1 \parallel *, *)$, this bound is at most equal to $\frac{q_h}{2^l}$. Finally, we can see that $|Pr\left[Succ_5\right] - Pr\left[Succ_4\right]| \leq \frac{(q_h+2q_s)}{2^l}$.

- Game G_6: In this game, we want to find a forge of forging M_4 without random oracles. we can add steps to $Send(GW^t, U^i, M_4)$: the simulator $\mathcal{B}$ requires to verify $M_4 \in L_P - list$ and $(ID_{GW} \parallel * \parallel MI^{new}, *)$, $(MI^{new} \parallel r_3, *)$, $(MI^{new} \parallel *, *)$, $(1, B_2 \parallel C_1 \parallel *, *)$, $(MI \parallel SID \parallel * \parallel C_3)$ and $(ID \parallel MI \parallel MI^{new} \parallel SID \parallel D_2 \parallel D_3 \parallel r_3, *)$ are in $L_A - list$. The last two terms have the upper bound $\frac{q_s}{2^l}$ and the others have at most $\frac{q_h}{2^l}$. So, we can see that $|Pr\left[Succ_6\right] - Pr\left[Succ_5\right]| \leq \frac{(5q_h+2q_s)}{2^l}$.

- Game G_7: In this game, the adversary $\mathcal{A}$ uses random oracles to solve the ECGDH-problem. We modify the h_1 oracle as follows: If $\mathcal{A}$ asks a $(1, \alpha P \parallel \beta P \parallel \lambda)$, the simulator $\mathcal{B}$ checks if $(1, \alpha P \parallel \beta P \parallel *, sk) \in L_A - list$. If there exists such a term, $\mathcal{B}$ returns sk. Otherwise, $\mathcal{B}$ uses the ECDDH oracle to check $\lambda = ?\alpha\beta P$. If this check is failed, $\mathcal{B}$ stops the game and report failure. Otherwise, $\mathcal{B}$ chooses $sk \in \{0, 1\}^l$, answers to the query and finally adds $(1, \alpha P \parallel \beta P \parallel \lambda, sk)$ into L_A-list. Here, we intersects the game into two aspects. Firs of all, the adversary $\mathcal{A}$ asks $Corrupt$ ($smart\ card$)-query and then, gets all information of the card.
  - This aspect simulates active attacks. The adversary $\mathcal{A}$ selects a password PW^* with size N. Then, he/she can forge messages to start the session. Since $\mathcal{A}$ can ask at most q_s $Send$-query, the probability of guessing the correct password is $\frac{q_s}{N}$.
  - This aspect simulates passive attacks. Here, we have two cases:
    (a) In orther to break the ECGDH-problem, the adversary $\mathcal{A}$ asks $Execute$-queries and h_1-queries. $\mathcal{A}$ can retrieve from L_A-list with the probability that bounded by $\frac{1}{q_h}$. In this case, the probability is at most $q_h Adv_A^{ECGDH}(t + 4q_e T_m)$.
    (b) In orther to simulate the $Execute$-queries, the adversary $\mathcal{A}$ asks $Send$-queries. Similar to the last case, we can obtain the probability $q_h Adv_A^{ECGDH}(t + 2q_e T_m)$.

  Finally, we have:

$$
\begin{aligned}
| \quad & Pr\left[Succ_6\right] - Pr\left[Succ_5\right] | \\
\leq \quad & \frac{q_s}{N} + q_h Adv_A^{ECGDH}(t + 4q_e T_m) \\
+ \quad & q_h Adv_A^{ECGDH}(t + 2q_e T_m) \\
\leq \quad & \frac{2q_s}{N} + q_h Adv_A^{ECGDH}(t + (4q_e + 2q_s)T_m)
\end{aligned}
$$

- Game G_8: This game is about strong forward security. The adversary $\mathcal{A}$ can ask all $Corrupt$-oracles. However, in the light of the $sfs - fresh$ notion, $Corrupt(1^m)$-query should occure after $Test$. So, $\mathcal{A}$ can utilizes the old sessions only. Like game G_7, we can find $(1, \alpha P \parallel \beta P \parallel \alpha\beta P, sk)$ from L_A-list. The probability of obtaining αP and βP in the same session is $\frac{1}{(q_s+q_e)^2}$. Therefore, $|Pr\left[Succ_8\right] - Pr[Succ_7]| \leq 2q_h(q_s + q_e)^2 Adv_A^{ECGDH}(t + (4q_e + 2q_s)T_m)$. This implies that the adversary $\mathcal{A}$ has no more advantage and $Pr[Succ_8] = \frac{1}{2}$.

Finally, Theorem 1 is proved by combining all above games. □

5.3 Formal Verification Using ProVerif

This section analyses the security of the proposed protocol via the ProVerif as one of the most well-known formal automated security analysis tools.

5.3.1 Premises in the Verification

As in [36], first of all, we mention some realties containing: constants, shared keys, channels, equations and functions which are required for analysis of the protocol. The realties are described in Figure 1.

In order to test correspondence relevance for the sensor and the user (during the login and authentication phase), we use four different events. In addition, the first two queries check the session keys security and the last two verify the correctness of relevances of events. These events are presented in Figure 2.

```
(*Channels and shared keys are listed below*)
free ch1: channel. (*the public channel between the user
and the sensor*)
free ch2: channel. (*the public channel between the sensor
and GW*)
free sch1: channel [private]. (*the secret channel between
the user and GW*)
free sch2: channel [private]. (*the secret channel between
the sensor and GW*)
free sku: bitstring [private]. (*the user's session key*)
free sks: bitstring [private]. (*the sensor's session key*)
(*Constants are listed below*)
free x:bitstring [private]. (*the private key of GW*)
free ID_i:bitstring [private]. (*Ui's identity*)
free PW_i:bitstring [private]. (*Ui's password*)
const IDGW:bitstring. (*GW's identity*)
const P:bitstring. (*the generator P*)
const SID_j:bitstring. (*S_j's identity*)
table d(bitstring). (*database in GW*)
(*Functions and equations are listed below:*)
fun h(bitstring):bitstring. (*hash function*)
fun h_1(bitstring):bitstring. (*hash function*)
fun mul(bitstring,bitstring):bitstring.
    (*scalar multiplication function*)
fun xor(bitstring,bitstring):bitstring. (*XOR function*)
fun con(bitstring,bitstring):bitstring.
    (*string concatenation*)
equation for all m:bitstring, n:bitstring;
    xor(xor(m, n), n)
    = m. (*XOR computation*)
equation forall m:bitstring,n:bitstring;
    mul(mul(P, m), n)
= mul(mul(P, n), m).(*scalar multiplication*)
```

Figure 1: The ProVerif code definition

```
Events
event UserStart(bitstring)
event UserAuth(bitstring)
event SensorStart(bitstring)
event SensorAuth(bitstring)
Queries
query attacker(sku)
query attacker(sks)
query id:bitstring; inj-event(UserAuth(id))
== > inj-event(UserStart(id)).
query sid:bitstring; inj-event(SensorAuth(sid))
== > inj-event(SensorStart(sid).
```

Figure 2: Events and queries in Proverif code

5.3.2 Scheme Model

We simulate our proposed scheme in parallel execution steps. Moreover, there are three entities in our scheme as participants and each participant has its own process:

$$process!User|!GW|!Sensor.$$

The processes of the user, the sensor and the gateway are mentioned in Figure 3, Figure 4 and Figure 5, respectively. The processes of the user and the sensor can be divided into two separated parts: registration and authentication. The process of the gateway includes three parts: two parts for registration and one part for authentication.

5.3.3 The Verification Results

The final main results are shown in Figure 6. It determines that the session keys are secure via the verification.

5.3.4 Comparison

In this section, we compare our proposed scheme with other schemes from both of the security and performance points of views. We want to compare our proposed scheme with some recent well-known schemes: Wu *et al.*'s scheme ([36]), Hsieh *et al.*'s scheme ([14]), Shi *et al.*'s scheme ([31]) Choi *et al.*'s scheme ([6]), Chang *et al.*'s scheme ([4]) and Farash *et al.*'s scheme ([10]).

Please note that since there are two versions of Chang *et al.*'s scheme ([4]): One is based on the hash functions and the other one is based on the elliptic curve cryptography, we use S1 and S2 to denote the versions.

Security comparison:

Although Wu *et al.* claimed that their proposed scheme is resistant against to replay attack and user forgery attack, however we showed that their scheme is vulnerable against these attacks.

In the security comparison posed in Table 4, we consider these security properties: Insider attack, off-line guessing attack, user forgery attack, gateway forgery attack, sensor capture attack, desyncronization attack, replay attack, known-key attack, user anonymity and strong forward security.

Performance comparison:

In this section, we discuss about performance of our scheme and compare it with some related schemes. Table 5 presents the comparison and uses the following notations and considerations:

- T_s denotes the time cost of a scalar multiplication in G and T_h is the time for a hash computation. In accordance with the Xu *et al.*'s scheme ([37]), we can see that $T_s \gg T_h$.

Table 4: Comparison of the security parameters

	Our scheme	[36]	[14]	[32]	[31]	[6]	[4] (S1)	[4] (S2)	[10]
Immune to the insider attack	✓	✓	×	✓	✓	✓	✓	✓	✓
Immune to the off-line guessing attack	✓	✓	×	×	×	×	×	×	×
Immune to the user forgery attack	✓	×	×	×	×	×	✓	✓	✓
Immune to the gateway forgery attack	✓	✓	✓	✓	✓	✓	✓	✓	✓
Immune to the sensor capture attack	✓	✓	×	×	✓	✓	✓	✓	✓
Immune to the de-syncronization attack	✓	✓	✓	✓	✓	✓	✓	✓	✓
Immune to the replay attack	✓	×	✓	✓	✓	✓	✓	✓	✓
Immune to the known-key attack	✓	✓	×	✓	✓	✓	✓	✓	✓
User anonymity	✓	✓	✓	×	×	×	✓	✓	✓
Strong forward security	✓	✓	×	×	✓	✓	×	✓	✓

- We consider that the points in G has totally 320 bits. The security parameter l is 160-bit and hence, the length of secret parameters such as x in the gateway, random numbers, the hash results and SID_j are 160-bits. Moreover, we use Q_u and Q_s to denote the quantities of the users and the sensors in the WSN. $|P|$, $|p|$ and $|q|$ are lengths for the parameters P, p and q such that $|p| \approx 160$ and $|q| \approx 160$.
- In Table 5, we show $(Q_u + Q_s + 1)$ with the Q_T.

6 Conclusion

In this paper, we firstly discussed on the security evaluation of the Wu *et al.*'s user authentication scheme and showed that their scheme is vulnerable against forgery attack and DoS attack. After that, in order to eliminate the weaknesses, we proposed an improved user authentication scheme. In addition, we presented a formal security analysis of our scheme via ProVerif and we suggested a provable security for the proposed scheme. Finally, we compared security and efficiency of our proposed scheme with some related schemes which indicate that the proposed scheme is a well-performed, secure and more practical scheme for IoT communications.

References

[1] A. Akbarzadeh, M. Bayat, B. Zahednejad, A. Payandeh, and M. R. Aref, "A lightweight hierarchical authentication scheme for internet of things," *Journal of Ambient Intelligence and Humanized Computing*, July 2018. (https://doi.org/10.1007/s12652-018-0937-6)

[2] A. A. Alamr, F. Kausar, J. Kim, and C. Seo, "A secure ECC-based RFID mutual authentication protocol for internet of things," *The Journal of Supercomputing*, vol. 74, no. 9, pp. 4281–4294, 2018.

[3] E. Bresson, O. Chevassut, and D. Pointcheval, "Security proofs for an efficient password-based key exchange," in *Proceedings of the 10th ACM Conference on Computer and Communications Security*, pp. 241–250, 2003.

[4] C. C. Chang and H. D. Le, "A provably secure, efficient, and flexible authentication scheme for ad hoc wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 1, pp. 357–366, 2016.

[5] T. H. Chen and W. K. Shih, "A robust mutual authentication protocol for wireless sensor networks," *ETRI Journal*, vol. 32, no. 5, pp. 704–712, 2010.

[6] Y. Choi, D. Lee, J. Kim, J. Jung, J. Nam, and D. Won, "Security enhanced user authentication protocol for wireless sensor networks using elliptic curves cryptography," *Sensors*, vol. 14, no. 6, pp. 10081-106, 2014.

[7] M. L. Das, "Two-factor user authentication in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 3, pp. 1086–1090, 2009.

[8] P. K. Dhillon and S. Kalra, "Secure multi-factor remote user authentication scheme for internet of things environments," *International Journal of Communication Systems*, vol. 30, no. 16, pp. e3323, 2017.

[9] R. Fantacci, T. Pecorella, R. Viti, and C. Carlini, "A network architecture solution for efficient iot wsn backhauling: challenges and opportunities," *IEEE Wireless Communications*, vol. 21, no. 4, pp. 113–119, 2014.

[10] M. S. Farash, M. Turkanović, S. Kumari, and M. Hölbl, "An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the internet of things environment," *Ad Hoc Networks*, vol. 36, pp. 152–176, 2016.

[11] X. Feng, X. Liu, and H. Yu, "A new internet of things group search optimizer," *International Journal of Communication Systems*, vol. 29, no. 3, pp. 535–552, 2016.

[12] H. Hayouni, M. Hamdi, and T. H. Kim, "A survey on encryption schemes in wireless sensor networks," in *7th International Conference on Advanced Software Engineering and Its Applications (ASEA'14)*, pp. 39–43, 2014.

[13] D. He, Y. Gao, S. Chan, C. Chen, and J. Bu, "An enhanced two-factor user authentication scheme in

h

Table 5: Comparison of performance

	our scheme	[36]	[14]	[32]	[31]	[6]	[4] (S1)	[4] (S2)	[10]
User's complexity	$2T_m + 13T$	$2T_m + 13T_h$	$8T_h$	$7T_h$	$3T_m + 5T_h$	$3T_m + 7T_h$	$7T_h$	$2T_m + 7T_h$	$11T_h$
Sensor's complexity	$2T_m + 4T_h$	$2T_m + 4T_h$	$2T_h$	$5T_h$	$2T_m + 4T_h$	$2T_m + 4T_h$	$5T_h$	$2T_m + 5T_h$	$7T_h$
Gateway's complexity	$1T_m + 13T_h$	$13T_h$	$5T_h$	$7T_h$	$T_m + 4T_h$	$T_m + 4T_h$	$8T_h$	$9T_h$	$14T_h$
Communication Cost (bits)	3680	3680	1280	4000	3840	4220	2720	3040	3520
Private number stored in the Gateway(bits)	160	160	160	$160Q_T$	320	320	$160Q_T$	$160Q_T$	160
Security for IoT	✓	×	×	×	×	×	×	×	×

wireless sensor networks." *Ad Hoc & Sensor Wireless Networks*, vol. 10, no. 4, pp. 361–371, 2010.

[14] W. B. Hsieh and J. S. Leu, "A robust user authentication scheme using dynamic identity in wireless sensor networks," *Wireless Personal Communications*, vol. 77, no. 2, pp. 979–989, July 2014.

[15] M. S. Hwang, Li-Hua Li, "A New Remote User Authentication Scheme Using Smart Cards", *IEEE Transactions on Consumer Electronics*, vol. 46, no. 1, pp. 28–30, Feb. 2000.

[16] R. Kantola, H. Kabir, and P. Loiseau, "Cooperation and end-to-end in the internet," *International Journal of Communication Systems*, vol. 30, no. 12, pp. e3268, 2017.

[17] M. K. Khan and K. Alghathbar, "Cryptanalysis and security improvements of 'two-factor user authentication in wireless sensor networks'," *Sensors*, vol. 10, no. 3, pp. 2450–2459, 2010.

[18] P. Kumar and H. J. Lee, "Cryptanalysis on two user authentication protocols using smart card for wireless sensor networks," in *Wireless Advanced (WiAd'11)*, pp. 241–245, 2011.

[19] S. Kumari, X. Li, F. Wu, A. K. Das, K. K. R. Choo, and J. Shen, "Design of a provably secure biometrics-based multi-cloud-server authentication scheme," *Future Generation Computer Systems*, vol. 68, pp. 320–330, 2017.

[20] S. Kumari, X. Li, F. Wu, A. K. Das, H. Arshad, and M. K. Khan, "A user friendly mutual authentication and key agreement scheme for wireless sensor networks using chaotic maps," *Future Generation Computer Systems*, vol. 63, pp. 56–75, 2016.

[21] S. Kumari, "Design flaws of "an anonymous two-factor authenticated key agreement scheme for session initiation protocol using elliptic curve cryptography," *Multimedia Tools and Applications*, vol. 76, no. 11, pp. 581–583, June 2017.

[22] S. Kumari, M. Karuppiah, A. K. Das, X. Li, F. Wu, and N. Kumar, "A secure authentication scheme based on elliptic curve cryptography for iot and cloud servers," *The Journal of Supercomputing*, pp. 1–26, 2017.

[23] C. T. Li, M. S. Hwang, "A lightweight anonymous routing protocol without public key en/decryptions for wireless ad hoc networks",*Information Sciences*, vol. 181, no. 23, pp. 5333–5347, Dec. 2011.

[24] C. T. Li, M. S. Hwang and Y. P. Chu, "An efficient sensor-to-sensor authenticated path-key establishment scheme for secure communications in wireless sensor networks", *International Journal of Innovative Computing, Information and Control*, vol. 5, no. 8, pp. 2107-2124, Aug. 2009.

[25] W. T. Li, T. H. Feng, and M. S. Hwang, "Distributed detecting node replication attacks in wireless sensor networks: A survey," *International Journal of Network Security*, vol. 16, no. 5, pp. 323–330, 2014.

[26] Z. Liu, E. Wenger, and J. Großschädl, "Mote-ecc: Energy-scalable elliptic curve cryptography for wireless sensor networks," in *International Conference on Applied Cryptography and Network Security*, pp. 361–379, 2014.

[27] P. Middleton, P. Kjeldsen, and J. Tully, "Forecast: The internet of things, worldwide," *Gartner Research*, 2013. (https://www.gartner.com/doc/2625419/forecast-internet-things-worldwide-)

[28] K. T. Nguyena, M. Laurentb, N. Oualha,"Survey on secure communication protocols for the internet of things," *Ad Hoc Networks*, vol. 32, pp. 17–31, 2015.

[29] S. Rostampour, N. Bagheri, M. Hosseinzadeh, and A. Khademzadeh, "A scalable and lightweight grouping proof protocol for internet of things applications," *The Journal of Supercomputing*, vol. 74, no. 1, pp. 71–86, 2018.

[30] O. Said, "Analysis, design and simulation of internet of things routing algorithm based on ant colony optimization," *International Journal of Communication Systems*, vol. 30, no. 8, pp. e3174, 2017.

[31] W. Shi and P. Gong, "A new user authentication protocol for wireless sensor networks using elliptic curves cryptography," *International Journal of Distributed Sensor Networks*, vol. 9, no. 4, pp. 730831, 2013.

[32] 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*, vol. 20, pp. 96–112, 2014.

[33] B. Vaidya, D. Makrakis, and H. T. Mouftah, "Improved two-factor user authentication in wireless sensor networks," in *IEEE 6th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob'10)*, pp. 600–606, 2010.

```
The user's process:
let User=
new r0:bitstring;
let MPi=h(con(r0,PWi)) in
let MIi=h(con(r0,IDi)) in
out(sch1,(MPi,MIi,IDi));
in(sch1,(xei:bitstring,xfi:bitstring));
let ei = xei in
let fi = xfi in
let di = xor(h(con(IDi,PWi)),r0) in
!
(
event UserStart(IDi);
let r1 = xor(di,h(con(IDi,PWi))) in
let MIi' = h(con(r1,IDi)) in
let MPi' = h(con(r1,PWi)) in
new alpha:bitstring;
new r2:bitstring;
new r3:bitstring;
new ti':bitstring;
let MIinew = h(con(r2,IDi)) in
let B1= xor(xor(ei,MPi'),r3) in
let B2 = mul(P,alpha) in
let B3 = xor(xor(xor(fi,MIi'),MIinew),
h(con(r3,MIi'))) in
let B4 = xor(IDi,h(con(con(r3,MIinew),B2))) in
let B5 = h(con(con(con(IDi,MIi'),MIinew),SIDj)) in
let M1 =(MIi',SIDj,B1,B2,B3,B4,B5) in
out(ch1,M1);
in (ch1,(xC1:bitstring,xC3:bitstring,xD2:bitstring,
xD3:bitstring,xD4:bitstring));
if xD4 = h(con(con(con(con(con(con(IDi,MIi'),
MIinew),SIDj),xD2),xD3),r3)) then
let B6 = mul(xC1,alpha) in
let sku = h1(con(con(B2,xC1),B6)) in
if xC3 = h(con(con(MIi',SIDj),sku)) then
let dinew = xor(r2,h(con(IDi,PWi))) in
let einew = xor(xor(xD2,h(con(MIinew,r3))),
h(con(r2,PWi))) in
let finew = xor(xor(xD3,MIinew),h(con(MIi',r3))) in
let di = dinew in
let ei = einew in
let fi = finew in
0).
```

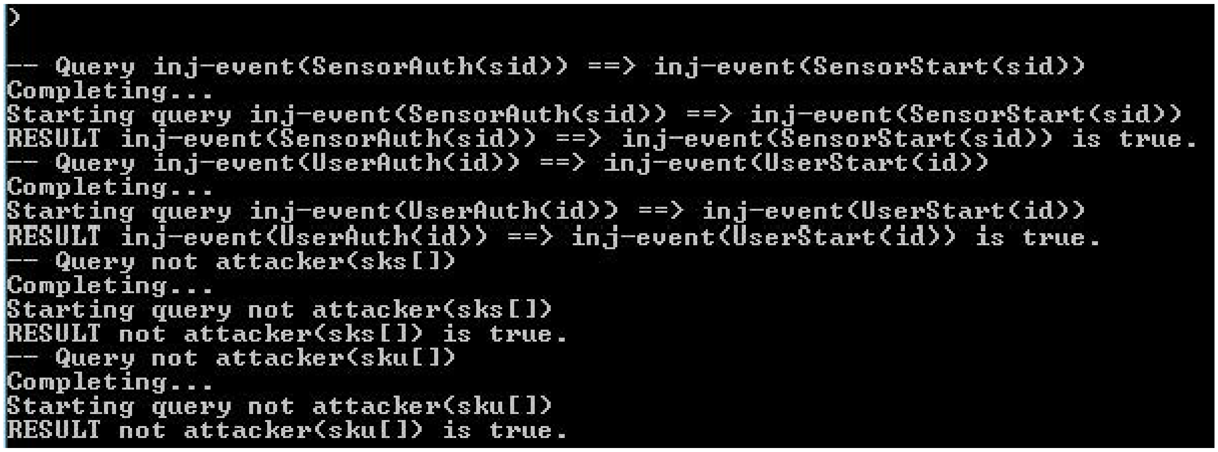
Figure 3: Code for the user's role

```
The sensor's process:
let Sensor =
out(sch2,SIDj);
in(sch2, xxcj:bitstring);
!
(
in(ch2,(uMIi:bitstring,uSIDj:bitstring,uB2:bitstring,
   uD1:bitstring, xxC0:bitstring));
if uSIDj = SIDj then
if uD1 = h(con(con(con(uMIi,uSIDj),xxcj),uB2)) then
event SensorStart(uSIDj);
new beta:bitstring;
let C1 = mul(P,beta) in
let C2 = mul(uB2,beta) in
let sks = h1(con(con(uB2,C1),C2)) in
let C3 = h(con(con(uMIi,SIDj),sks)) in
let C4 = h(con(con(con(mul(xxcj,xxC0),uMIi),SIDj),Yj)) in
let M3 = (C1,C3,C4) in
out(ch2,M3);
0
).
```

Figure 4: Code for the sensor's role

```
User registration
let GWReg1 =
in(sch1,(xMPi:bitstring,xMIi:bitstring,xIDi:bitstring));
let ei'= xor(con(con(IDGW,x),xMIi),xMPi) in
let fi'= xor(h(con(xMIi,x)),xMIi) in
insert d(xIDi);
out (sch1,(ei',fi')).
Sensor registration
let GWReg2 =
in(sch2,(ySIDj:bitstring));
let cj = h(con(ySIDj,x)) in
out(sch2,(cj)).
Authentication
let GWAuth =
in(ch1,(xxMIi:bitstring,xxSIDj:bitstring,xxB1:bitstring,
xxB2:bitstring, xxB3:bitstring,xxB4:bitstring,xxB5:bitstring));
let xr3 = xor(xxB1,con(con(IDGW,x),xxMIi)) in
let xMIinew = xor(xor(xxB3,h(con(xxMIi,x))),
h(con(xr3,xxMIi))) in
let xIDi = xor(xxB4,h(con(con(xr3,xMIinew),xxB2))) in
get d(=xIDi) in
new lambda:bitstring;
let C0 = mul(P,lambda) in
if xxB5 = h(con(con(con(xIDi,xxMIi),xMIinew),xxSIDj)) then
event UserAuth(xIDi);
let pcj = h(con(xxSIDj,x)) in
let xxD1 = h(con(con(con(xxMIi,xxSIDj),mul(pcj,C0)),xxB2)) in
let M2 =(xxMIi,xxSIDj,xxB2,xxD1,C0) in
out (ch2,M2);
in (ch2,(xxC0:bitstring,xxC1:bitstring,xxC3:bitstring,xxC4:bitstring));
if xxC4 = h(con(con(mul(pcj,C0),xxMIi),xxSIDj)) then
event SensorAuth(xxSIDj);
let D2 = xor(h(con(con(IDGW,x),xMIinew)),
h(con(xMIinew,xr3))) in
let D3 = xor(h(con(xMIinew,x)), h(con(xxMIi,xr3))) in
let D4 =
con(con(con(con(con(con(xIDi,xxMIi),xMIinew),xxSIDj),
D2),D3),xr3) in
let M4 = (xxC1,xxC3,D3,D4,D5) in
out(ch1,M4).
```

Figure 5: Code for the gateway's role



```
>
-- Query inj-event(SensorAuth(sid)) ==> inj-event(SensorStart(sid))
Completing...
Starting query inj-event(SensorAuth(sid)) ==> inj-event(SensorStart(sid))
RESULT inj-event(SensorAuth(sid)) ==> inj-event(SensorStart(sid)) is true.
-- Query inj-event(UserAuth(id)) ==> inj-event(UserStart(id))
Completing...
Starting query inj-event(UserAuth(id)) ==> inj-event(UserStart(id))
RESULT inj-event(UserAuth(id)) ==> inj-event(UserStart(id)) is true.
-- Query not attacker(sks[])
Completing...
Starting query not attacker(sks[])
RESULT not attacker(sks[]) is true.
-- Query not attacker(sku[])
Completing...
Starting query not attacker(sku[])
RESULT not attacker(sku[]) is true.
```

Figure 6: Results of the verification by ProVerif

[34] R. Watro, D. Kong, S. f. Cuti, C. Gardiner, C. Lynn, and P. Kruus, "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, 2004.

[35] F. Wu, L. Xu, S. Kumari, and X. Li, "A new and secure authentication scheme for wireless sensor networks with formal proof," *Peer-to-Peer Networking and Applications*, vol. 10, no. 1, pp. 16–30, 2017.

[36] F. Wu, L. Xu, S. Kumari, and X. Li, "A privacy-preserving and provable user authentication scheme for wireless sensor networks based on internet of things security," *Journal of Ambient Intelligence and Humanized Computing*, vol. 8, no. 1, pp. 101–116, Feb. 2017.

[37] L. Xu and F. Wu, "Cryptanalysis and improvement of a user authentication scheme preserving uniqueness and anonymity for connected health care," *Journal of Medical Systems*, vol. 39, no. 2, p. 10, Jan. 2015.

[38] H. L. Yeh, T. H. Chen, P. C. Liu, T. H. Kim, and H. W. Wei, "A secured authentication protocol for wireless sensor networks using elliptic curves cryptography," *Sensors*, vol. 11, no. 5, pp. 4767–4779, 2011.

[39] H. L. Yeh, T. H. Chen, P. C. Liu , T. H. Kim and H. W. Wei, "A secured authentication protocol for wireless sensor networks using elliptic curves cryptography," *Sensors*, vol. 11, no. 5, pp. 4767–4779, 2011.

[40] Y. Y. Zhang, X. Z. Li, and Y. A. Liu, "The detection and defence of DoS attack for wireless sensor network," *The Journal of China Universities of Posts and Telecommunications*, vol. 19, pp. 52–56, 2012.

Biography

Majid Bayat received his Ph.D. from the Department of Mathematics and Computer Sciences at Kharzmi University in Tehran, Iran. He is presently an assistant professor of Computer Engineering of Shahed University, Tehran, Iran. His research interests include cryptographic protocols, smart grid and IoT security.

Mohammad Beheshti Atashgah is a PhD candidate at ICT Complex, Malek-Ashtar University of Technology Tehran, Iran. He is a researcher in Information System and security lab (ISSl) in Sharif University. His research interests include IoT security and provable security.

Morteza Barari was born in Freydoonkenar, Iran. He received a Ph.D. degree from AmirKabir University of Technology in 2003. He is currently a faculty member at the Department of Electrical Engineering of the Malek-Ashtar University of Technology, Tehran, Iran. He has published more than 50 papers and 2 books. His research interests are in stochastic signal processing, radar design, satellite communication, and adaptive array processing.

Mohammad Reza Aref received the B.Sc. degree in 1975 from the University of Tehran, Iran, and the M.Sc. and Ph.D. degrees in 1976 and 1980, respectively, from Stanford University, Stanford, CA, USA, all in electrical engineering. He returned to Iran in 1980 and was actively engaged in academic affairs. He was a Faculty member of Isfahan University of Technology from 1982 to 1995. He has been a Professor of electrical engineering at Sharif University of Technology, Tehran, since 1995, and has published more than 230 technical papers in communication and information theory and cryptography in international journals and conferences proceedings. His current research interests include areas of communication theory, information theory, and cryptography.