A Lightweight ECC-based Three-Factor Mutual Authentication and Key Agreement Protocol for WSNs in IoT

Meriam Fariss, Hassan El Gafif, Ahmed Toumanari
Laboratory of Applied Mathematics and Intelligent Systems Engineering (MAISI)
National School of Applied Sciences (ENSA)
Agadir, Morocco

***Abstract*—The Internet of Things (IoT) represents a giant ecosystem where many objects are connected. They collect and exchange large amounts of data at a very high speed. One of the main parts of IoT is the Wireless Sensor Network (WSN), which is deployed in various critical applications such as military surveillance and healthcare that require high levels of security and efficiency. Authentication is a primary security factor that ensures the legitimacy of data requests and responses in WSN. Moreover, sensor nodes are characterized by their limited resources, which raise the need for lightweight authentication schemes applicable in IoT environments. This paper presents an informal analysis of the security of X. Li et al.'s protocol, which is claimed to be efficient and resistant to various attacks. The analysis results show that the reviewed protocol does not provide user anonymity and it is vulnerable to session key disclosure attack, many-time pad attack, and insider attack. To address all these requirements, a new three-factor authentication protocol is presented, which guarantees higher security using Physically Unclonable Function (PUF) and Elliptic Curve Cryptography (ECC). This protocol does not only withstand the security weaknesses in X. Li et al.'s scheme but also provides smart card revocation and is resistant to cloning attack. In terms of both computational and communicational costs, results demonstrate that the proposed scheme provides higher efficiency in comparison with other related protocols, which makes it notably suitable for IoT environments.**

***Keywords*—Mutual authentication; elliptic-curve cryptography; Physically Unclonable Function; wireless sensor networks; key-agreement; internet of things**

I. Introduction

It is widely believed that the Internet of Things (IoT) [1, 2] is the upcoming promising technology that will bring many revolutionary changes in different life sides. The IoT architecture connects a set of heterogeneous things belonging to our daily life use and enables them to exchange a huge amount of data, which are also processed and stored. One of the principal application domains of IoT is Wireless Sensor Networks (WSN)[3]. WSNs are generally deployed in unattended areas and consist of widely distributed autonomous sensing devices, gateway nodes (GWNs), and remote users communicating over the public channel. The GWN represents a bridge of communication between sensors and users. Sensors play an important role in WSN by monitoring environmental and physical conditions and providing real-time data, which are directly accessed by users as and when demanded. Therefore, a secure mutual authentication process represents a primary concern in WSNs allowing only legitimate users to access the sensed data. Moreover, WSN consists of many resource-constrained sensor nodes having limited power, low bandwidth and battery, small storage space, and limited computational abilities. These two main issues related to security concerns and performance limitations in WSN deployment represent an important challenge that must be taken into consideration in every proposed protocol. ECC represents an important security solution in WSN [4] by offering the same security level compared to other cryptography mechanisms (e.g. RSA) with much smaller key size and less computational power requirements, which makes it suitable for resource-constrained environments.

Contribution. Many authentication protocols are designed to ensure higher and efficient security in WSNs. However, some security and performance challenges are still not solved which makes these protocols vulnerable to several security attacks and not applicable in resource-constrained environments such as WSN. In this paper, an efficient three-factor mutual authentication and key agreement protocol is proposed that overcomes critical security concerns found in the studied protocols such as impersonation attacks, cloning attacks, and insider attacks. In addition, it guarantees user and sensor anonymity and untraceability. In contrast with the existing protocols, no secure channel assumption is required by this protocol. It is also more efficient than all the studied schemes in terms of communications cost. It is also computationally more efficient than most of the studied protocols. As a result, it is more suitable for WSN than the studied schemes. Additionally, this protocol provides smart card revocation (in case of lost/stolen smart card) and identity and password update feature. Another interesting feature provided by this protocol, that does not exist in the studied protocols, is that the GWN cannot decrypt the communicated messages between the user and the sensor since it does not possess the session key.

Organization of the paper. The remainder of this paper is organized as follows: Section II presents the related work. In Section III, an overview of the main preliminaries of the present paper is provided. Section IV, describes the principal weaknesses of X. Li et al.'s scheme [5]. The proposed protocol is described in detail in Section V. The security analysis of this

This work is supported by the National Center for Scientific and Technical Research (CNRST) [scholarship number: 4UIZ2017].

protocol and its performance analysis are described in Sections VI and VII respectively. Finally, some concluding remarks are given in Section VIII.

II. Related Work

In the literature, there are various mutual authentication protocols designed to address the security and performance challenges in WSNs. In 2009, Das [6] proposed a hash-based two-factor user authentication protocol for WSN using the smart card. Subsequently, many authentication protocols have been proposed to improve the security of Das' scheme [7–9]. He et al. (2010) [10] proposed an enhancement of [6] that overcomes insider and impersonation attacks. However, Kumar and Lee (2011) [11] have identified that He et al.'s protocol [10] lacks many security features such as no user anonymity and it does not establish a session key between the user and the sensor. In 2011, Yeh et al. [8] proposed an ECC-based authentication protocol for WSN to improve security with higher efficiency. Unfortunately, this protocol cannot provide mutual authentication and key agreement. To overcome the weaknesses detected in [8], Shi et al. (2013)[12] proposed a user authentication protocol based on ECC which improves security features, communicational, and computational costs. Nonetheless, [12] contains other weaknesses. Choi et al. (2014) [13] presented a review of [12] and have found that it is not secure since it cannot withstand session key attack and smart card attack. In the same year, Jiang et al. [14] proposed a two-factor authentication scheme for WSN. In 2015, Wu et al. [15] pointed out some weaknesses in [13] and [14] such as being vulnerable to off-line guessing attack and user forgery attack. Wu et al. presented an enhanced protocol based on ECC that addresses the security weaknesses detected and provides higher security. Nam et al. (2014) [16] proposed an authentication scheme for WSN using ECC that provides user anonymity and perfect forward secrecy. In 2015, Jiang et al. [17] designed an ECC-based two-factor authentication protocol. After reviewing He et al.'s scheme (2015) [18] and presenting its main security weaknesses such as stolen smart card attack and tracking attack, Jiang et al. proved that their proposed scheme achieves mutual authentication and key agreement between the user and the sensor, it also guarantees user anonymity and untraceability. In 2016, Lu et al. [19] proposed a two-factor mutual authentication and key agreement protocol using a smart card. They claimed that their proposition is resistant to insider attack due to the use of the hashed value of the password. It is also claimed to be resistant to many attacks such as known session-specific temporary information attack and a denial-of-service attack. In 2022, Chander et al. [20] proposed an improved two-factor authentication scheme for WSN using ECC.

To improve the security of two-factor authentication protocols, three-factor authentication has drawn researchers' attention and many three-factor authentication protocols are proposed [5, 21–25]. Moreover, biometric recognition [26] presents many advantages that guarantee a higher security level in WSNs compared with passwords. For this reason, many biometric-based authentication protocols have been proposed [27, 28]. In 2016, Park et al. [22] proposed a three-factor ECC-based authentication protocol using biometric information to overcome security weaknesses detected in Chang et al.'s scheme (2015) [23] such as incorrectness of password change and off-line guessing attack. Later on, Jung et al. (2017) [24] pointed out that [23] is vulnerable to password guessing attack and user impersonation. They also demonstrated that this protocol does not provide session key verification. To overcome these security weaknesses, Jung et al. proposed an improved authentication and key agreement protocol using the user's biometric information. In the same year, S.Challa et al. [25] proposed a signature-based authentication and key agreement protocol in IoT using ECC. They claimed that their protocol is secure against several attacks such as privileged insider attack and stolen smart card attack. In 2018, X. Li et al.[5] proposed a fingerprint-based mutual authentication protocol for WSN, which they claimed provides user and sensor anonymity and untraceability and many other security features.

As you can notice from the aforementioned literature, important research work has been done to detect and overcome security weaknesses in WSN environments for secure communication between different entities. Moreover, security issues are not the only factor that should be taken into consideration. Each security protocol should be efficient enough to be applied and suitable for IoT applications due to the resource-constrained feature of different devices used. To address the security and efficiency issues raised in the previous work, this paper proposes a three-factor mutual authentication and key agreement protocol for WSN based on ECC.

III. Preliminaries

This section gives an overview of the main preliminary concepts used in the present paper.

A. Physical Unclonable Function (PUF)

A PUF [29–31] is a low-cost technology that extracts entropy from uncontrollable manufacturing variations in the physical structure of identically produced devices. Typically, it is physically impossible to recreate the same conditions in another device even if the same manufacturing process is performed again, and it is mathematically impossible to accurately predict the PUF's behavior as well. PUF uses this entropy to generate a unique sequence of bits (response) for each device given an input (challenge) acting as the device's fingerprint that does not need to be stored in the device's memory. To measure the performance of a given PUF, researchers use two main metrics:

- Uniqueness (μ_{inter}): the average fraction of dissimilar bits between responses of different PUFs to a given challenge. The ideal value of μ_{inter} is 0.5 (random).
- Reliability (μ_{intra}): the average fraction of dissimilar bits between responses of a fixed PUF to a given challenge. This metric measure the average error resulted in PUF's output due to the undesirable noise. The ideal value of μ_{intra} is 0 (no error).

In our protocol, we use a recently proposed PUF scheme. HBN-PUF [32] is a strong, chaos-enhanced, and asynchronous PUF. According to [33], the creators of HBN-PUF aim to move quickly to commercialize this technology.

B. Fuzzy Commitment Scheme

The fuzzy commitment scheme was introduced by Juels and Wattenberg in 1999[34]. This technique is commonly used in biometric authentication schemes and it combines error-correcting code techniques and cryptography. For an error-correcting code over a message space $M = \{0,1\}^k$, we consider a set of codewords $C \subset \{0,1\}^n$ where $n > k$ to achieve redundancy. Before transmission, each message $m \in M$ is mapped to a codeword $c \in C$. We define the translation function g: $M \rightarrow C$, and the decoding function f: $\{0,1\}^n \rightarrow C \cup \{\varnothing\}$ that maps arbitrary an n-bit string to the nearest codeword, else it outputs $\varnothing$. Biometric-based applications use a reference template generated firstly at the registration phase. At the authentication phase, a new biometric sample is provided and compared to the reference template which needs to be securely and secretly stored. Due to many reasons, the provided biometric sample is not the same as the reference template.

Let's consider the secure one-way hash function $h:\{0,1\}^n \rightarrow \{0,1\}^l$. The fuzzy commitment scheme is defined as follows: F: $\{0,1\}^n, \{0,1\}^n \rightarrow \{0,1\}^l, \{0,1\}^n$. F commits a random codeword $c \in C$ to the biometric template b provided at the registration phase to the server. The server computes then $F(c,b) = (\alpha,\delta)$, where $\alpha = h(c)$ and $\delta = c \oplus b$. The server stores (α,δ) in its database. At the authentication phase, a noisy biometric data b' is input by the user. To open the commitment F using b', the server computes $c' = f(b' \oplus \delta)$, using the decoding function f. Then, it compares $h(c')$ with the stored value of α. If $h(c') = \alpha$, the commitment is opened successfully and the user is authenticated.

IV. Weaknesses of X. Li et al.'s Scheme [5]

This section describes the functional and security flaws of X.Li et al.'s three-factor anonymous authentication scheme [5]. It involves three main types of entities: the user U_i, the trusted gateway node *GWN*, and the sensor node S_j.

A. GWN Master Key Update

In the reviewed protocol, the *GWN* has its private key x and the master key K_{GWN}. When the *GWN* updates K_{GWN}, it must recalculate $B_i = h(ID_i \| K_{GWN}) \oplus h(RPW_i \| c_i)$ for each U_i and $K_{GWN\text{-}S} = h(SID_j \| K_{GWN})$ for each S_j. Moreover, when the *GWN* updates its private key x it must recalculate its public key X and send it to all users to be updated on their smart cards. Hence, the *GWN* master key update is a very expensive process.

B. No Smart Card Revocation

When a user's smart card is lost/stolen, he/she should be able to send a revocation request to the *GWN*. However, X. Li et al.'s protocol does not provide this feature.

C. No user Identity Change

In real life, the user needs to change his identity but X. Li et al.'s scheme does not provide this feature.

D. Insider Attack

The protocol of X. Li et al. is exposed to an insider attack by a legitimate user, who can start by performing successfully a usual login phase by inputting his valid biometric information, ID_i and PW_i. Then the login request message $\{M_2, M_4, M_5, M_6, M_7\}$ is sent to the *GWN*, which calculates $M_8 = ID_i \oplus K_{GWN\text{-}S}$ and sends it explicitly over the public channel to S_j. Since the U_i knows his own ID_i, he can compute the secret key $K_{GWN\text{-}S} = M_8 \oplus ID_i$. By knowing this secret information, the adversary can perform many other attacks that we detail in the following paragraphs.

E. Many-time Pad Attack

In X. Li et al.'s scheme, the same sensor node S_j uses the One-Time Pad $K_{GWN\text{-}S}$ for all the users with whom it communicates: for user U_1 we have $M_8^1 = ID_1 \oplus K_{GWN\text{-}S}$, for user U_2 we have $M_8^2 = ID_2 \oplus K_{GWN\text{-}S}$, etc. If an attacker intercepts the message M_8 corresponding to at least two different users U_1 and U_2, he can perform the Many-Time Pad attack ($M_8^1 \oplus M_8^2 = ID_1 \oplus ID_2$). Generally, the ID_i chosen by users is a low entropy information, hence the attacker can perform a dictionary attack to recover ID_1 (or ID_2) and can then calculate $K_{GWN\text{-}S} = M_8^1 \oplus ID_1$.

F. User Anonymity

X. Li et al. presumed that the user's real identity ID_i is shielded in their protocol; however, we proved in the aforementioned attack that user anonymity is not guaranteed. The adversary who has the secret key $K_{GWN\text{-}S}$ (from previous attacks) can easily reveal each user's identity by catching the exchanged message $\{M_8, M_9, M_{10}, M_{11}\}$ between the legitimate user and the *GWN* and computing $ID_i = M_8 \oplus K_{GWN\text{-}S}$.

G. Session Key Disclosure Attack

From the insider attack, the adversary, who also knows SID_j, gets the sensor node secret key $K_{GWN\text{-}S}$ that allows him to calculate the session key as follows: the adversary obtains ID_i from the user anonymity flaw, and calculates the *GWN* random number $r_g = M_9 \oplus h(ID_i \| K_{GWN\text{-}S})$ from the message $\{M_8, M_9, M_{10}, M_{11}\}$ sent by the *GWN*. The legal sensor S_j generates its private random number r_j and calculates $M_{12} = r_j \oplus K_{GWN\text{-}S}$ that it sends explicitly over the public channel to the *GWN*. The adversary can retrieve $r_j = M_{12} \oplus K_{GWN\text{-}S}$ from M_{12} and $r_i = r_g \oplus M_{10}$ from M_{10}. Hence, the attacker can easily calculate the session key $SK = h(ID_i \| SID_j \| r_i \| r_j \| r_g)$.

H. Sensor Impersonation Attack

X. Li et al. presumed that S_j cannot be impersonated since the S_j's secret key $K_{GWN\text{-}S} = h(SID_j \| K_{GWN})$ is unknown. Through the previous attacks, we have shown that $K_{GWN\text{-}S}$ can be calculated, thus an attacker can impersonate S_j.

I. Sensor Node Untraceability

Any adversary can trace different sessions between a particular user U_i and a sensor node S_j since the exchanged message $M_8 = ID_i \oplus K_{GWN\text{-}S}$ stays the same in all sessions.

V. Proposed Protocol

To overcome the functional and security flaws described in the previous section, the current paper proposes this improved protocol that involves three main entities: the Gateway Node (*GWN*) as a trusted entity, the Sensor Node (S_j), and the User (U_i). It consists of six phases: initialization phase, user and sensor registration, login and mutual authentication, user's identity update, user's password update, and smart card revocation. Table I summarizes the notations used throughout this section.

TABLE I. NOTATIONS USED IN THE PRESENT PAPER AND THEIR DESCRIPTIONS

Parameter	Description
U_i	User
S_j	Sensor node
GWN	Gateway node
F_p	Finite field of order *p*
$E(F_p)$	Elliptic Curve
P	Generator Point
n	Order of *P*
h()	Hash function
C	Set of codewords
pn_i	U_i's phone number
PUF()	Physical Unclonable Function
x_{GWN}	*GWN*'s private key (master key)
$X_{GWN} = x_{GWN}P$	*GWN*'s public key
y_i	U_i's private key
$Y_i = y_i$P	U_i's public key
$E_{AES}()_S$	AES Encryption using the key *S*
$D_{AES}()_S$	AES Decryption using the key *S*
TS	Timestamp
$status_i$	The status of the user (active or inactive)
ID_i	U_i's identity (64 bits)
SID_J	Sensor identity (64 bits)
SC	Smart Card
DB	Database

A. Initialization Phase

Before the execution of the protocol, the initialization phase must be performed by the *GWN*. It selects an additive group *G* and its generator point *P* of order *n* (a large prime number) on an elliptic curve $E(F_p)$ where F_p is a finite field. *GWN* chooses randomly its private key $x_{GWN} \in Z_n^*$ and computes the corresponding public key $X_{GWN} = x_{GWN}.P$. At last, *GWN* stores its private key and publishes the system parameters $\{E(F_p), G, P, h(), X_{GWN}\}$ where *h*() is a 128-bit hash function.

B. Registration Phase

Unlike the reviewed scheme, the registration phase in this protocol does not require a secure channel to exchange data with the *GWN* neither for the user nor for the sensor.

1) User registration: In the user registration phase, it is assumed that U_i already possesses a private key y_i and published its corresponding public key $Y_i=y_i.P$. This key pair is needed to encrypt/decrypt the parameters ID_i, b_i, and pn_i. This phase involves U_i and GWN. At the end of this phase, U_i becomes a legitimate user. The details of this phase are shown in Fig. 1.

2) Sensor registration: This phase involves *GWN* and S_j. *GWN* should store some data in each S_j's memory before deploying the sensors in the WSN. First, *GWN* selects SID_j for each S_j, generates a random number u_j, and stores SID_j and u_j in S_j's memory. Moreover, each S_j has its own pre-implemented $PUF_j()$.*GWN* calculates $K_{GWN-S}=PUF_j(u_j)$ and stores $\{SID_j, K_{GWN-S}\}$ in its DB.

C. Login and Mutual Authentication

To remotely access the sensed data of S_j, U_i should perform a successful login. Moreover U_i, S_j, and *GWN* must be mutually authenticated to exchange data. This phase is performed over a public channel as shown in Fig. 2.

D. User's Password Update

The password update phase allows a legitimate user U_i to change his/her old password PW_i^{old} to a new one PW_i^{new}. The steps of this phase are shown in Fig. 3.

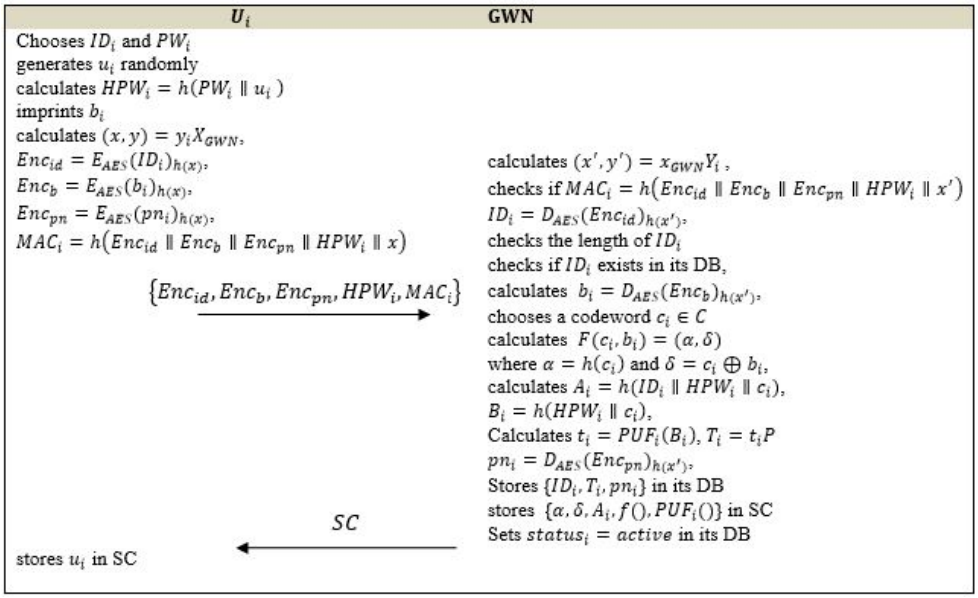

Fig. 1. User Registration Phase.

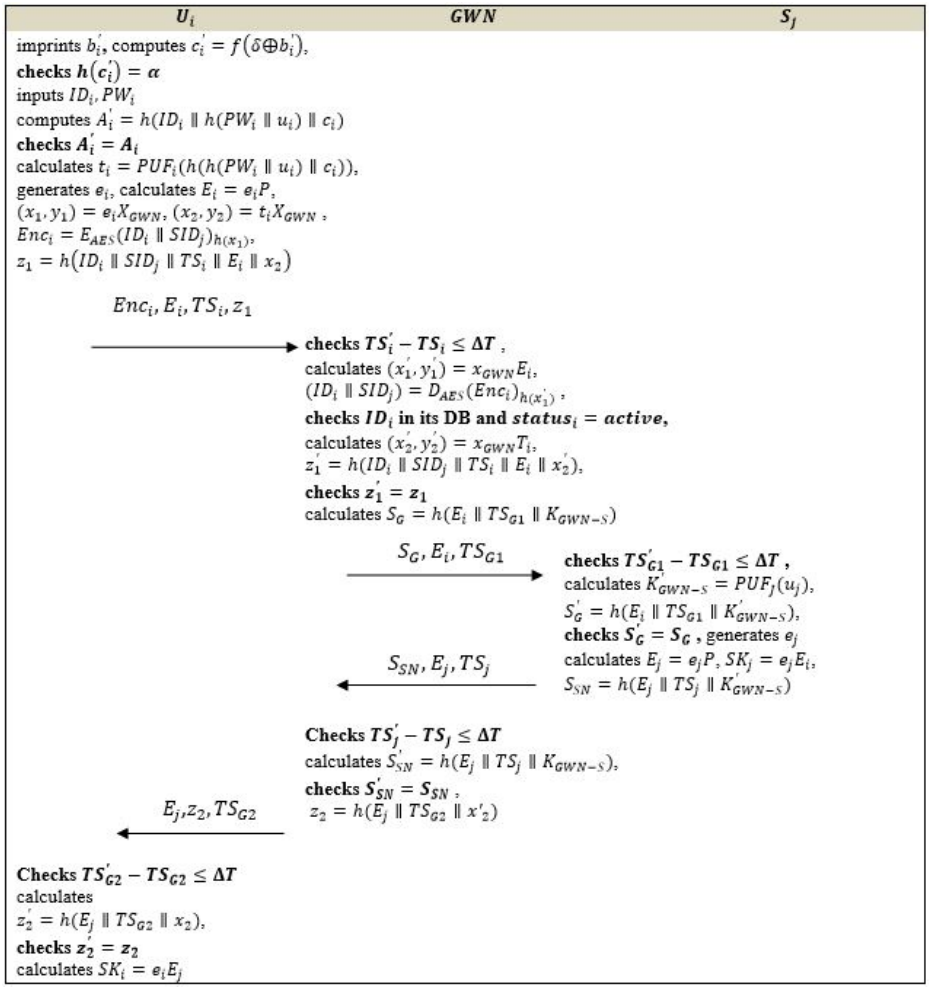

Fig. 2. Login and Mutual Authentication Phase.

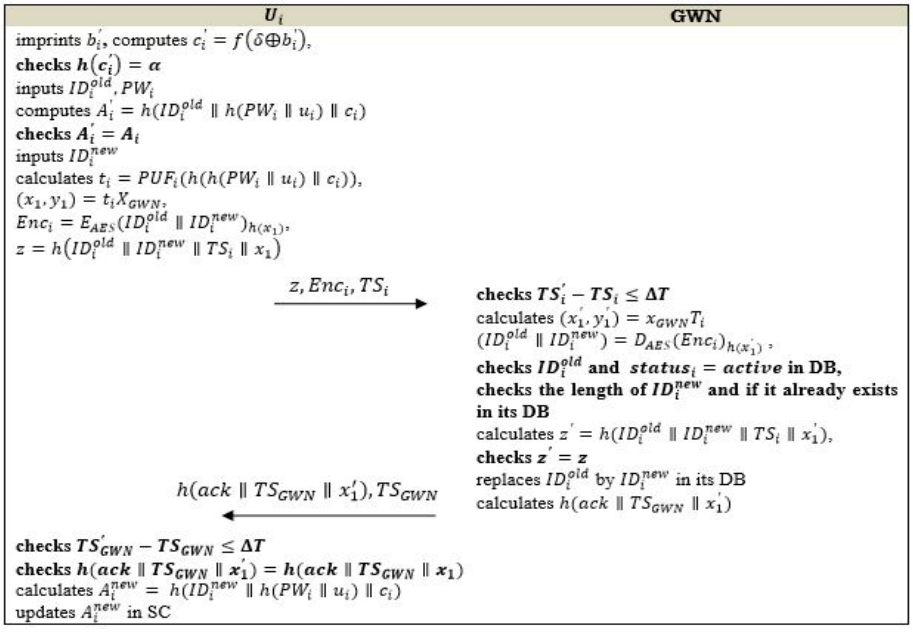

Fig. 3. User's Identity Update Phase.

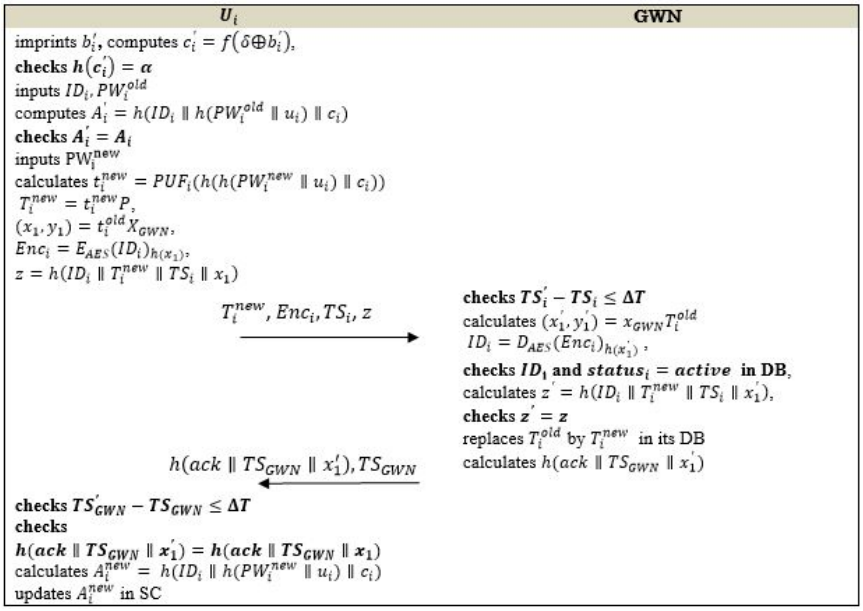

Fig. 4. User's Password Update Phase.

E. User's Identity Update

In addition to password update, this scheme allows each legitimate user U_i to change his/her ID_i whenever he/she wants, following the steps shown in Fig. 4.

F. Smart Card Revocation and Re-registration

A legal user U_i can revoke his/her lost or stolen SC through a secure process. Once U_i realizes the loss of his/her *SC*, he/she informs *GWN* to obtain a new one. SC revocation and re-registration phase steps are described below:

- Revocation request: it is performed remotely over the public channel. The main purpose of this step is to deactivate the stolen/lost SC immediately when the Ui sends a revocation request to the GWN. Therefore, the re-registration step can be performed later on. First, the revocation request is sent by Ui to GWN which verifies the user's identity through a one-time password mechanism using Ui's phone number pni which was already stored in the GWN's database at the user's registration phase. Then, GWN checks if statusi=active. If it holds, GWN updates Ti=Null and statusi=inactive in its DB. Else, the revocation request is terminated. From then on, the lost/stolen SC is no more valid.

- Re-registration: this step aims to securely authenticate Ui who wants to re-register in the system after the revocation of his/her old SC. At the end of this step, a new active SC is delivered to the legitimate user by the GWN. First, Ui presents in person his/her Personal Identification Card to GWN for verification. Then the authenticated user inputs his IDi, chooses PWinew, and calculates HPWinew=h(PWinew || uinew) (uinew is generated randomly). Then, Ui imprints binew. The GWN checks if IDi exists in its DB and statusi=inactive. If this is true, GWN chooses a codeword cinew∈C, calculates F(cinew, binew)=(αnew, δnew) where αnew=h(cinew) and δnew= cinew⊕ binew. Then GWN computes Ainew=h(IDi||HPWinew|| cinew), Binew=h(HPWinew||cinew), and Tinew=PUFinew(Binew).P. After performing these operations, GWN updates Tinew and sets statusi=active in its DB. Then GWN stores the parameters {αnew,δnew,Ainew,PUFinew(),f()} in the new SC. Finally, GWN delivers the new SC to Ui. When receiving the SC, Ui stores the random number uinew in his/her new SC. From then on, Ui can use his new active SC securely over the public channel.

VI. Security Analysis of the Proposed Protocol

A. Informal Security Analysis

This section informally discusses the security features of the current protocol. This scheme provides important security features and resists many security attacks in WSN. Remember that all communications in this protocol are performed over the public channel and no secure channel assumption is required. Moreover, Table II presents the security features comparison between the proposed protocol and the following schemes: X.Li et al. [5], Choi et al. [13], Zhang et al. [35], and S.Challa et al. [25]. The first 12 evaluation criteria are proposed in [36]. In addition, proposed three new criteria are provided:

- C13. SN Anonymity: the sensor's identity is protected and its activity cannot be traced.

- C14. Freely Identity Change: the identity is memorable, and can be chosen freely and changed by the user.

- C15. Suitable for IoT: no direct communication between the users and sensors, the communication should be through a gateway.

Table II shows that all the studied schemes satisfies *C1*, *C2*, and *C3* except [35] that does not satisfies *C2* since it does not support password update feature. Except for [35] that does not

support smart cards, all the studied protocols are resistant to SC loss attack (*C4*). For instance, in the current protocol, when an adversary steals a legitimate user's smart card, he needs B_i to calculate $t_i=PUF(B_i)$. However, $B_i=h(h(PW_i||u_i)||c_i)$ is not stored in SC, so the attacker has to compute it. This is not possible, because he needs to know the password PW_i and the codeword c_i. Moreover, when U_i's SC is stolen, the *GWN* changes the parameter $status_i$ to *inactive* after performing a SC revocation, as explained previously, making it no more valid.

TABLE II. COMPARISON OF SECURITY FEATURES OF THE STUDIED PROTOCOLS

Evaluation Criteria	**[5]**	**[13]**	**[35]**	**[25]**	**Our protocol**
C1. No password verifier-table	*Yes*	*Yes*	*Yes*	*Yes*	*Yes*
C2. Password friendly	*Yes*	*Yes*	*No*	*Yes*	*Yes*
C3. No password exposure	*Yes*	*Yes*	*Yes*	*Yes*	*Yes*
C4. No smart card loss attack	*Yes*	*Yes*	-	*Yes*	*Yes*
C5. Resistance to known attacks					
- Replay Attack	*Yes*	*Yes*	*Yes*	*Yes*	*Yes*
- Offline Password Guess	*Yes*	*Yes*	*Yes*	*Yes*	*Yes*
- User Impersonation attack	*Yes*	*No*	*Yes*	*Yes*	*Yes*
- SN Impersonation attack	*No*	*Yes*	*Yes*	*Yes*	*Yes*
- GWN Impersonation attack	*Yes*	*Yes*	*Yes*	*Yes*	*Yes*
- Cloning Attack	*No*	*No*	*No*	*No*	*Yes*
- Insider Attack	*No*	*Yes*	*Yes*	*Yes*	*Yes*
C6. Sound repairability	*No*	*No*	-	*Yes*	*Yes*
C7. Provision of key agreement	*Yes*	*Yes*	*Yes*	*Yes*	*Yes*
C8. No clock synchronization	*Yes*	*No*	*No*	*No*	*No*
C9. Timely typo detection	*Yes*	*Yes*	*No*	*Yes*	*Yes*
C10. Mutual authentication	*Yes*	*Yes*	*Yes*	*Yes*	*Yes*
C11. User anonymity	*No*	*No*	*No*	*Yes*	*Yes*
C12. Forward secrecy	*Yes*	*Yes*	*Yes*	*Yes*	*Yes*
C13. SN anonymity	*No*	*No*	*No*	*Yes*	*Yes*
C14. Freely identity change	*No*	*No*	*No*	*No*	*Yes*
C15. Suitable for IoT	*Yes*	*No*	*Yes*	*No*	*Yes*

The *C5* criteria consist of the following attacks:

- Replay attack: as shown in Table II, all the studied schemes are resistant to replay attack. In this protocol, timestamps are used to prevent this attack.
- Offline password guess: Table II shows that all the studied protocols are resistant to the offline password guess. In this protocol, if an attacker has Ui's SC, this means that he has access to four elements: α=h(ci), δ=ci⊕bi, Ai=h(IDi||h(PWi||ui)||ci), and ui. The goal of the attacker in this attack is to get PWi (we suppose that he already knows the IDi of SC owner). To guess PWi by performing a password dictionary attack on Ai, the attacker needs to know ci first. However, according to Juels and Wattenberg [34], it's impossible to retrieve ci from α and δ. In addition, the attacker cannot guess ci from Ai using a dictionary attack because ci is a random number with high entropy. As a result, the current protocol resists the offline password attack.
- User, GWN, and SN impersonation attack: this protocol is resistant to user, GWN, and SN impersonation attacks since the attacker cannot compute z1=h(IDi||SIDi ||TSi||Ea||x2), z2=h(Ea||TSG2||x2), SG=h(Ea||TSG1|| KGWN-Sj), or SSN=h(Ea||TSj||KGWN-Sj). That is because he is not able to get KGWN-Sj or to compute x2 as long as he does not have access to Ui's private key ti or GWN's master key xGWN. In contrast, [5] is vulnerable to the SN impersonation attack as showed in the "Weaknesses of X.Li et al.'s scheme" section above and [13] suffers from user impersonation attack.
- Cloning attack: In contrast with the other studied schemes, this protocol is resistant to cloning attack. To protect SC from the cloning attack, PUF is used to compute Ui's private key ti. Suppose that an attacker somehow succeeded to get the PWi and ci and cloned Ui's SC, then the attacker will try to compute ti'=PUFicloned(h(h(PWi||ui)||ci)). However, the computed ti' will be different from Ui's private key ti because the cloned PUFicloned is different from the original PUFi as explained previously in the preliminaries. Moreover, the current protocol is resistant to sensor cloning attack since if an attacker clones the sensor he will get uj and will try to compute K'GWN-S=PUFjcloned(uj). However, the computed K'GWN-S will be different from the original KGWN-S because the cloned PUFjcloned is different from the original PUFj.
- Insider attack: This attack is performed by a legitimate user Ua to gain additional privileges. The author in [5] is the only scheme that suffers from insider attack as we described in the "Weaknesses of X.Li et al.'s scheme" section above. In the following, a proof that this scheme is resistant to the insider attack is given. In this analysis, the goal is to prove that an insider attacker has no advantage compared with an outsider attacker. That is, all the insider attacker's additional information, that an outsider attacker does not possess, does not give him any additional power as an attacker. First, all the data that an insider attacker Ua possesses and outsider attackers do not have are specified as follows: IDa, PWa, ba,αa=h(ca),δa=ca⊕ba,ua,Aa=h(IDa||h(PWa||ua) ||ca),ta=PUFa(h(h(PWa||ua)||ca)), and in each session he has: ea and SIDj of the targeted sensor Sj. Notice that these data, except SIDj, are unique to Ua and do not consist of any information that is used by other users, sensor nodes, or GWN. Thus, these data give Ua no advantage to attack other users, sensor nodes, or GWN compared to an outsider attacker. Next, it is important to make sure that these data do not help Ua to extract some useful information from the messages exchanged in his sessions. In the following, the messages exchanged in each session of Ua are cited (excluding the messages generated by Ua): SG=h(Ei|| TSG1||KGWN-S),Ei,TSG1,SSN=h(Ej||TSj||KGWN-S),Ej,TSj, z2=h(Ej||TSG2||x2), TSG2. Obviously, the public elements Ei, TSG1, Ej, TSj, and TSG2 can be retrieved by any attacker not only an insider attacker; thus, they represent no advantage of an insider attacker

over the outsider attackers. Besides, since the messages SG, SSN, and z2 are unique to each session, the insider attacker cannot use them to attack other sessions. Additionally, assuming that the used hash function is secure, the insider attacker cannot extract KGWN-S or x2 from SG, SSN, or z2. As a result, you can conclude that an insider attacker does not possess and cannot extract any additional useful information as an attacker compared to an outsider attacker. Thus, insider and outsider attackers have the same capabilities.

According to [36], sound repairability (*C6*) means that the scheme provides SC revocation without requiring the user to change her identity. As you can see in the smart card revocation and re-registration sub-section, this protocol perfectly fits to this criteria and does not require any identity update after SC revocation. The authors in [5] and [13] do not support the SC revocation feature at all and [35] does not support smart cards. Table II also shows that all the studied schemes guarantee mutual authentication (*C10*) and provision of a key agreement (*C7*). In [5], the generated session key consists of a hash of the user ID, the sensor SID, and three fresh nonces that are new for each session. As a result, if an attacker disclosed one or more session keys, he would not be able to guess the other session keys as long as the hash function is secure. Thus, [5] satisfies the forward secrecy criteria (*C12*). The rest of studied protocols, including our protocol, satisfy *C12* as well since the generated session key depend on a fresh ECDH shared key and by knowing some session keys the attacker will not be able to affect the other sessions except if he can resolve the ECDH problem which is cryptographically hard.

Only [5] satisfies the "No clock synchronization" (*C8*) criteria since all the other protocols employ timestamps to prevent replay attacks. Timely typo detection criteria (*C9*) require that the user will be timely notified if she inputs wrong credentials by mistake when login. This is satisfied by all the studied schemes except [35] which do not employ any checking locally in the login phase. The credentials in [35] are checked afterward by the gateway node. The current protocol and [25] meet the user and sensor node anonymity criteria (*C11* and *C13*). In this protocol, The user and sensor anonymity is guaranteed by the fact that ID_i and SID_j are not sent clearly over the public channel, and they are encrypted $Enc_i=E_{AES}(ID_i \| SID_j)_{h(x1)}$. Thus, no adversary can reveal the user's or sensor's real identity from the exchanged messages. The untraceability of the user and the sensor node is provided in this scheme through the fact that all exchanged data vary from one session to another. This is because E_i and E_j are generated in each session randomly, thus Enc_i, z_1, S_G, S_{SN}, and z_2 vary from a session to another since they are computed based on E_i or ID_j. In contrast, [5] does not fulfill this criteria as showed in the "Weaknesses of X.Li et al.'s scheme" section above. The authors in [13] and [35] do not guarantee this criteria either since they communicate the user's ID and sensor's SID in clear during the authentication phase.

In addition to the abovementioned features, the current protocol is the only protocol that supports freely identity change (*C14*). Besides, it is suitable for IoT (*C15*) since there is no direct data exchange between the user and the sensor node, unlike [13] and [25], where the sensor node contacts the user directly.

B. Formal Security Analysis using AVISPA Tool

This section formally analyzes the security and authentication logic of the current protocol using the widely used AVISPA Tool. The implementation of this protocol in the HLPSL is provided. The analysis results will show that this protocol is safe. In this HLPSL specification, three basic roles representing the protocol's principals are defined: user, gateway, and sensor. In addition, there are two composed roles, session and environment, and a goals section where the security goals are specified. Table III contains the notations used in the specification and their corresponding notations in the protocol's login and authentication phase provided in Fig. 2.

TABLE III. HLPSL SPECIFICATION'S NOTATIONS AND THEIR CORRESPONDING PROTOCOL NOTATIONS

HLPSL Specification's Notations	**Corresponding Protocol's Notations**
U, GWN, and S	The agents' IDs
TS1, TS2, TS3, TS4	TS_i, TS_{G1}, TS_{G2}, TS_j
UU, US	u_i, u_j
DELTA	δ
B, PW	b_i, PW_i
P	The generator point *P*
XGWN, PXGWN	x_{GWN}, X_{GWN}
Hash, F, PUFU, PUFS	*h*, *f*, PUF_i, PUF_j
E1, PE1, E2, PE2	e_i, E_i, e_j, E_j
C, A	c_i, A_i
T, PT	t_i, T_i
SK	*SK*
KGWNS	K_{GWN-S}

```
role user( U, GWN, S: agent,
           TS1, TS4, UU, DELTA, B, PW: text,
           P: nat,
           PXGWN : message,
           Hash, F, PUFU: hash_func,
           SND, RCV: channel(dy) )
played_by U def=
local
        State: nat,
        E1, C, A, T: text,
        SK, PE2, X: message
init
        State := 0
transition
        1.State = 0 ∧ RCV(start) =|>
        State':=1
        ∧ C' := F(xor(DELTA, B))
        ∧ A' := Hash(U.Hash(PW.UU).C)
        ∧ T' := PUFU(Hash(Hash(PW.UU).C'))
        ∧ E1':= new()
        ∧ X':=exp(PXGWN,T')
        ∧  SND(exp(P,E1').{U.S}_Hash(exp(PXGWN,E1')).TS1.
Hash(U.S.TS1.exp(P,E1'). X'))
        ∧  witness(U,GWN,gateway_user_e1,Hash(U.S.TS1.exp(
P,E1').X'))
        2.State = 1 ∧ RCV(PE2'.TS4.Hash(PE2'.TS4.X)) =|>
        State':=2
        ∧ SK':=exp(PE2',E1)
        ∧ request(U,GWN,user_gateway_e2,Hash(PE2'.TS4.X))
end role
```

Fig. 5. User Role's HLPSL Specification.

Fig. 5 describes the user role specification. The gateway role's specification is provided in Fig. 6. The sensor role's specification is described in Fig. 7. In Fig. 8, the HLPSL specification of the session and environment roles is provided. Fig. 9 shows the analysis results obtained using the OFMC and CL-AtSe backends. Currently, AVISPA Tool provides four backends: OFMC, CL-AtSe, SATMC, and TA4SP. However, SATMC, and TA4SP do not support operations like xor() and exp(). Both OFMC and CL-AtSe analysis results show that this protocol is safe against active and passive attacks.

```
role gateway(
        U, GWN, S: agent,
        KGWNS: message,
        TS1, TS2, TS3, TS4, XGWN: text,
        PT: message,
        Hash: hash_func,
        SND, RCV: channel(dy) )
played_by GWN def=
local
        State: nat,
        PE1, PE2: message
init
        State := 0
transition
        1.State = 0 ∧ RCV(PE1'.{U.S}_Hash(exp(PE1',
XGWN)).TS1.Hash(U.S.TS1.PE1'.exp(PT ,XGWN))) =|>
        State' := 1
        ∧ SND(PE1'.TS2.Hash(PE1'.TS2.KGWNS))
        ∧ request(GWN,U,gateway_user_e1,Hash(U.S.TS1.PE1'
.exp(PT,XGWN)))
        ∧ witness(GWN,S,sensor_gateway_e1,Hash(PE1'.TS2.
KGWNS))
        2.State = 1 ∧ RCV(PE2'.TS3.Hash(PE2'.TS3.KGWNS))
=|>
        State' := 2
        ∧ SND(PE2'.TS4.Hash(PE2'.TS4.exp(PT,XGWN)))
        ∧ request(GWN,S,gateway_sensor_e2,Hash(PE2'.TS3.
KGWNS))
        ∧ witness(GWN,U,user_gateway_e2,Hash(PE2'.TS4.
exp(PT,XGWN)))
end role
```

Fig. 6. Gateway Node Role's HLPSL Specification

```
%%%% SESSION ROLE %%%%%%%%%%%%%%%%
role session(
        U, GWN, S: agent,
        P : nat,
        TS1, TS2, TS3, TS4, XGWN, UU, DELTA, B, PW, US: text,
        Hash, F, PUFU, PUFS: hash_func)
def=
local
        SU, RU, SGWN, RGWN, SS, RS: channel(dy)
composition
        user(U, GWN, S, TS1, TS4, UU, DELTA, B, PW, P, exp(P, XGWN),
Hash, F, PUFU, SU, RU )
        ∧ gateway(U, GWN, S, PUFS(US), TS1, TS2, TS3, TS4, XGWN,
exp(P,PUFU(Hash(Hash(PW.UU).F(xor(DELTA, B))))), Hash, SGWN, RGWN)
        ∧ sensor(U, GWN, S, P, TS2, TS3, US, Hash, PUFS, SS, RS)
end role
%%%% ENVIRONMENT ROLE %%%%%%%%%%%%%%
role environment() def=
const
        u, gwn, s: agent,
        p: nat,
        xgwn, ts11, ts21, ts31, ts41, ts12, ts22, ts32, ts42, ts13, ts23, ts33, ts43,
ts14, ts24, ts34, ts44, uu, deltau, bu, pwu, us, ui, deltai, bi, pwi, ui: text,
        sk, gateway_user_e1, user_gateway_e2, sensor_gateway_e1,
gateway_sensor_e2: protocol_id,
        h, f, pufu, pufs, pufi: hash_func
intruder_knowledge = {u, gwn, s, i, p, exp(p, xgwn), ui, deltai, bi, pwi,
exp(p,pufu(h(h(pwu.uu) .f(xor(deltau, bu))))), h, pufi, f, ts11, ts21, ts31, ts41,
ts12, ts22, ts32, ts42, ts13, ts23, ts33, ts43, ts14, ts24, ts34, ts44}
composition
        session(u,gwn,s,p, ts11, ts21, ts31, ts41, xgwn, uu, deltau, bu, pwu, us, h,
f, pufu, pufs)
        ∧ session(u, gwn, s, p, ts12, ts22, ts32, ts42, xgwn, uu, deltau, bu, pwu, us,
h, f, pufu, pufs)
        ∧ session(u, gwn, i, p, ts13, ts23, ts33, ts43, xgwn, uu, deltau, bu, pwu, ui,
h, f, pufu, pufi)
        ∧ session(i, gwn, s, p, ts14, ts24, ts34, ts44, xgwn, ui, deltai, bi, pwi, us, h,
f, pufi, pufs)
end role
%%%% GOALS %%%%%%%%%%%%%%%%
goal
authentication_on gateway_user_e1
authentication_on user_gateway_e2
authentication_on sensor_gateway_e1
authentication_on gateway_sensor_e2
secrecy_of sk
end goal

environment()
```

Fig. 7. Session, Environment, and Goals' HLPSL Specification.

```
role sensor(
        U, GWN, S: agent,
        P : nat,
        TS2, TS3, US: text,
        Hash, PUFS: hash_func,
        SND, RCV: channel(dy))
played_by S def=
local
        State: nat,
        E2 : text,
        SK, PE1 : message,
        KGWNS : message
init
        State := 0 ∧ KGWNS := PUFS(US)
transition
        1.State = 0 ∧ RCV(PE1'.TS2.Hash(PE1'.TS2.KGWNS))
=|>
        State' := 1
        ∧ E2' := new()
        ∧ SK' := exp(PE1', E2')
        ∧ SND(exp(P,E2').TS3.Hash(exp(P,E2').TS3.KGWNS))
        ∧ request(S,GWN,sensor_gateway_e1,Hash(PE1'.TS2.
KGWNS))
        ∧ secret(SK', sk, {U,S})
        ∧ witness(S,GWN,gateway_sensor_e2,Hash(exp(P,E2').
TS3.KGWNS))
end role
```

Fig. 8. Sensor Node Role's HLPSL Specification.

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/MAKA-
LoginAuth.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 5.96s
  visitedNodes: 550 nodes
  depth: 14 plies
```

```
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL
PROTOCOL
  /home/span/span/testsuite/results/MAKA-
LoginAuth.if
GOAL
  As Specified
BACKEND
  CL-AtSe
STATISTICS
  Analysed  : 119282 states
  Reachable  : 89374 states
  Translation: 0.05 seconds
  Computation: 67.03 seconds
```

Fig. 9. Analysis Result using OFMC and CL-AtSe Backends.

VII. Performance Comparison

This section presents the computational and communicational costs of this proposed protocol in comparison with other related schemes [5, 13, 25, 35]. To calculate the communication costs of the current protocol in comparison with the other related schemes, assume that the length of each element is as follows: user identity (64 bits), sensor identity (64 bits), hash (128 bits), timestamp (64 bits), ECC point (320 bits), AES (128 bits). From Table IV, you can obviously see that this protocol communication cost is the best compared to the other protocols. In terms of computation costs, the following notations are used: T_{EPM} denotes the time cost of one point multiplication computation on ECC, T_h denotes the time cost of one hash function computation and T_{AES} denotes the time cost of one AES encryption/decryption operation [37]. Note that T_{AES} requires much less time compared to T_{EPM} [38]. Note also that compared to hash functions, PUFs require much less hardware overhead to implement [39], thus the negligible execution time of PUF will not be included in the comparison. Table IV provides a summary of the computation costs comparison. The computation cost of this protocol is considerably higher than the computation cost of X.Li et al.'s scheme [5]. This is explained by the fact that the reviewed protocol does not use ECC point multiplication operations in all steps of the mutual authentication phase, which makes it vulnerable to many attacks as previously discussed in Section IV. This protocol requires slightly more computational costs than Choi et al.'s scheme [13], but it is more secure as

shown previously in Table IV. In addition, note that at the sensor side, which is a resource-constrained device, the current protocol generates less computation cost compared to Choi et al.'s scheme. Compared to the protocols proposed by S.Challa et al. [25] and Zhang et al. [35], this protocol achieves a better efficiency level since it only requires the execution of 8 ECC point-scalar multiplications in total, while the former protocols require the execution of 14 ECC point-scalar multiplications.

VIII. Conclusion

After reviewing X.Li et al.'s protocol and finding it to be vulnerable to many serious attacks such as insider attack, many time pad attack, lack of anonymity, and impersonation attacks, this paper presented a new mutual authentication and key agreement scheme that strengthens its security using three factors: password, smart card, and biometrics. The informal security analysis proved that this protocol resists all the attacks found in the studied protocol including X.Li et al.'s protocol. Additionally, a formal security analysis was given using AVISPA tool, which shows that the protocol is safe. In contrast with the studied protocols, this protocol also provides a freely identity change feature in addition to the password update.

TABLE IV. Comparison of Computation and Communication Costs of the Studied Protocols

	[5]	**[13]**	**[35]**	**[25]**	**Our protocol**
Computation time of U_i	$2T_{EPM}+8T_h$	$3T_{EPM}+9T_h$	$5T_{EPM}+4T_h$	$5T_{EPM}+5T_h$	$4T_{EPM}+8T_h+T_{AES}$
Computation time of *GWN*	$T_{EPM}+9T_h$	$T_{EPM}+5T_h$	$5T_{EPM}+5T_h$	$5T_{EPM}+4T_h$	$2T_{EPM}+5T_h+T_{AES}$
Computation time of S_j	$4T_h$	$2T_{EPM}+6T_h$	$4T_{EPM}+4T_h$	$4T_{EPM}+3T_h$	$2T_{EPM}+2T_h$
Total of computation costs	$3T_{EPM}+21T_h$	$6T_{EPM}+20T_h$	$14T_{EPM}+13T_h$	$14T_{EPM}+12T_h$	$8T_{EPM}+15T_h+2T_{AES}$
Communication costs (bits)	2368	3072	3168	2464	2176

On the other side, the proposed protocol achieved better results in terms of communication compared to the studied protocols. Computationally, X.Li et al.'s protocol is the most efficient. However, this efficiency advantage comes at the expense of security as shown previously. In terms of security-efficiency ratio, the current protocol can be considered better than all the studied schemes. In the future, it will be interesting to improve this protocol to be suitable for mobile IoT applications where objects can jump from a gateway node zone to another in the same session without repeating the authentication process.

References

[1] Ammar, M., Russello, G., Crispo, B.: Internet of Things: A survey on the security of IoT frameworks. J. Inf. Secur. Appl. 38, 8–27 (2018).

[2] Atzori, L., Iera, A., Morabito, G.: The Internet of Things: A survey. Comput. Networks. 54, 2787–2805 (2010).

[3] Ray, P.P.: A survey on Internet of Things architectures. J. King Saud Univ. - Comput. Inf. Sci. 30, 291–319 (2018).

[4] Suárez-Albela, M., Fernández-Caramés, T.M., Fraga-Lamas, P., Castedo, L.: A practical evaluation of a high-security energy-efficient gateway for IoT fog computing applications. Sensors (Switzerland). 17, 1–39 (2017).

[5] Li, X., Niu, J., Kumari, S., Wu, F., Sangaiah, A.K., Choo, K.K.R.: A three-factor anonymous authentication scheme for wireless sensor networks in internet of things environments. J. Netw. Comput. Appl. 103, 194–204 (2018).

[6] Das, M.L.: Two-factor user authentication in wireless sensor networks. IEEE Trans. Wirel. Commun. 8, 1086–1090 (2009).

[7] Nyang, D., Lee, M.-K.: Improvement of Das's Two-Factor Authentication Protocol in Wireless Sensor Networks. IACR Cryptol. ePrint Arch. 2009, 631 (2009).

[8] 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, 4767–4779 (2011).

[9] Chen, T.H., Shih, W.K.: A robust mutual authentication protocol for wireless sensor networks. ETRI J. 32, 704–712 (2010).

[10] 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. Networks. 10, 361–371 (2010).

[11] Kumar, P., Lee, H.J.: Cryptanalysis on two user authentication protocols using smart card for wireless sensor networks. 2011 Wirel. Adv. WiAd 2011. 241–245 (2011).

[12] Shi, W., Gong, P.: A new user authentication protocol for wireless sensor networks using elliptic curves cryptography. Int. J. Distrib. Sens. Networks. 2013, (2013).

[13] 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 (Switzerland). 14, 10081–10106 (2014).

[14] Jiang, Q., Ma, J., Lu, X., Tian, Y.: An efficient two-factor user authentication scheme with unlinkability for wireless sensor networks. Peer-to-Peer Netw. Appl. 8, 1070–1081 (2015).

[15] Wu, F., Xu, L., Kumari, S., Li, X.: A new and secure authentication scheme for wireless sensor networks with formal proof. Peer-to-Peer Netw. Appl. 10, 16–30 (2017).

[16] Nam, J., Kim, M., Paik, J., Lee, Y., Won, D.: A provably-secure ECC-based authentication scheme for wireless sensor networks. Sensors (Switzerland). 14, 21023–21044 (2014).

[17] Jiang, Q., Ma, J., Wei, F., Tian, Y., Shen, J., Yang, Y.: An untraceable temporal-credential-based two-factor authentication scheme using ECC for wireless sensor networks. J. Netw. Comput. Appl. 76, 37–48 (2016).

[18] He, D., Kumar, N., Chilamkurti, N.: A secure temporal-credential-based mutual authentication and key agreement scheme with pseudo identity for wireless sensor networks. Inf. Sci. (Ny). 321, 263–277 (2015).

[19] Lu, Y., Li, L., Peng, H., Yang, Y.: An energy efficient mutual authentication and key agreement scheme preserving anonymity for wireless sensor networks. Sensors (Switzerland). 16, 1–21 (2016).

[20] Chander, B., Kumaravelan, G.: An Improved 2-Factor Authentication Scheme for WSN Based on ECC. IETE Tech. Rev. 1–12 (2022).

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

[22] Park, Y.H., Park, Y.H.: Three-factor user authentication and key agreement using elliptic curve cryptosystem in wireless sensor networks. Sensors (Switzerland). 16, (2016).

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

[24] Jung, J., Moon, J., Lee, D., Won, D.: Efficient and security enhanced anonymous authentication with key agreement scheme in wireless sensor networks. Sensors (Switzerland). 17, (2017).

[25] Challa, S., Wazid, M., Das, A.K., Kumar, N., Goutham Reddy, A., Yoon, E.J., Yoo, K.Y.: Secure Signature-Based Authenticated Key Establishment Scheme for Future IoT Applications. IEEE Access. 5, 3028–3043 (2017).

[26] Delac, K., Grgic, M.: A survey of biometric recognition methods. Proc. Elmar - Int. Symp. Electron. Mar. 184–193 (2004).

[27] Althobaiti, O., Al-Rodhaan, M., Al-Dhelaan, A.: An efficient biometric authentication protocol for wireless sensor networks. Int. J. Distrib. Sens. Networks. 2013, (2013).

[28] Chaudhry, S.A., Naqvi, H., Farash, M.S., Shon, T., Sher, M.: An improved and robust biometrics-based three factor authentication scheme for multiserver environments. J. Supercomput. 74, 3504–3520 (2018).

[29] Herder, C., Yu, M.D., Koushanfar, F., Devadas, S.: Physical unclonable functions and applications: A tutorial. Proc. IEEE. 102, 1126–1141 (2014).

[30] Maiti, A., Gunreddy, V., Schaumont, P.: A systematic method to evaluate and compare the performance of physical unclonable functions. Embed. Syst. Des. with FPGAs. 9781461413, 245–267 (2013).

[31] Suh, G.E., Devadas, S.: Physical unclonable functions for device authentication and secret key generation. Proc. - Des. Autom. Conf. 9–14 (2007).

[32] Charlot, N., Canaday, D., Pomerance, A., Gauthier, D.J.: Hybrid Boolean Networks as Physically Unclonable Functions. IEEE Access. 9, 44855–44867 (2021).

[33] Jeff Grabmeier: Scientists harness chaos to protect devices from hackers,https://news.osu.edu/scientists-harness-chaos-to-protect-devices-from-hackers/.

[34] Juels, A., Wattenberg, M.: A fuzzy commitment scheme. In: Proceeding CCS '99 Proceedings of the 6th ACM conference on Computer and communications security. pp. 28–36 (1999).

[35] Zhang, K., Xu, K., Wei, F.: A provably secure anonymous authenticated key exchange protocol based on ECC for wireless sensor networks. Wirel. Commun. Mob. Comput. 2018, (2018).

[36] Wang, D., Wang, P.: Two Birds with One Stone: Two-Factor Authentication with Security beyond Conventional Bound. IEEE Trans. Dependable Secur. Comput. 15, 708–722 (2018).

[37] Salman, R.S., Farhan, A.K., Shakir, A.: Lightweight Modifications in the Advanced Encryption Standard (AES) for IoT Applications: A Comparative Survey. In: 2022 International Conference on Computer Science and Software Engineering (CSASE). IEEE. pp. 325–330 (2022).

[38] De La Piedra, A., Braeken, A., Touhafi, A.: A performance comparison study of ECC and AES in commercial and research sensor nodes. IEEE EuroCon 2013. 347–354 (2013).

[39] Bolotnyy, L., Robins, G.: Physically unclonable function -based security and privacy in RFID systems. Proc. - Fifth Annu. IEEE Int. Conf. Pervasive Comput. Commun. PerCom 2007. 211–218 (2007).