User-Authentication Protocol to Secure Wireless Sensor Network Access in the Internet of Things Context

Benfilali Mostefa, Gafour Abdelkader, and Belghachi Mohamed

Original scientific article

***Abstract*—Recently, the numerous academic papers have been published on Authentication and Key Agreement (AKA) schemes for securing wireless sensor networks (WSN) in the context of the Internet of Things (IoT). The goal of these schemes is to protect external users' access to data collected by WSNs . Due to limited resources and the wireless communication medium, the sensor nodes are vulnerable to multiple attacks performed by malicious user's. Unfortunately, most of the proposed schemes are insecure and require higher storage, communication and computing costs. This paper presents a User Authentication protocol to secure WSNs Access in the IoT context (UAWSNA-IoT). The BAN-Logic and Real-Or-Random (ROR) models are used to demonstrate the reliability of UAWSNA-IoT in meeting all requirements for mutual authentication and session key security, respectively. In addition, UAWSNA-IoT offers high security with low computational, storage and communication overhead, making it an ideal choice for resource-constrained IoT devices such as WSNs.**

***Index Terms*—Internet of Things, Wireless Sensor Networks, Mutual Authentication, Authentication and Key Agreement, User Authentication, Session Key, BAN-Logic, ROR model.**

I. Introduction

THE Internet of Things (IoT) is a global information society infrastructure that enables the delivery of advanced services by connecting physical and virtual objects. These heterogeneous objects are interconnected, locatable, addressable, and readable in the internet world [1]. IoT covers almost all areas of Information Technology (IT) such as smart cities, machine-to-machine systems, connected vehicles, and Wireless Sensor Networks (WSN), etc. WSN's represent a centrepiece of the success of the IoT, because they use small intelligent objects that are generally limited in terms of computational, storage, and energy capabilities. WSN consists of hundreds or thousands of sensor nodes deployed randomly or manually to monitor the hostile areas [4]. The primary goal of WSN is to detect and gather data on physical phenomena such as pressure, temperature, humidity, and blood pressure, and other [37]. Ultimately, this gathered data is transmitted to end users through wireless communication, facilitated by the GateWay Node (GWN) overseeing and coordinating the process. Furthermore, the Gateway Node (GWN) serves as a link between the WSN and the outside world (Internet), as all incoming and outgoing network data must pass through it [2]. In contrast to sensor nodes, the GWN has greater capabilities in terms of computing power, energy reserves and memory size. In addition , the wireless medium used in the various communications among network entities provides a suitable environment for attackers to carry out various attacks [6]. As a result, attackers can intercept, insert, delete, modify and redirect messages exchanged between legitimate parties. In such a scenario, message integrity, confidentiality and authentication are crucial to ensure the security and reliability of information exchanged between the entities [7]. Therefore, we require an efficient authentication protocol that is better suited to the resource-constrained environment of WSNs to secure the network and prevent attacks. In the literature, there are several user authentication protocols available for accessing wireless sensor networks [13], [24]. These protocols aim to ensure secure user access to data collected by sensor nodes. One commonly used protocol is mutual authentication, which incorporates the Authentication and Key Agreement (AKA) technique. In this protocol, a trusted third party called the gateway is involved in the authentication process. In the login/authentication stage, the gateway validates user's identities and provides them with authorization to access data collected by the sensor nodes. The authentication techniques can be divided into three categories: single-factor, two-factor, and three-factor [5], [18]. The AKA technology enables the creation of session keys shared between users and sensor nodes, which secure future communications between them. This ensures that only legitimate users with session keys are authorized to access to WSN.

Our paper's contribution addresses the following main points in light of the aforementioned challenges:

- We propose a novel lightweight and efficient authentication scheme named "UAWSNA-IoT" to secure the wireless sensor networks access from unauthorized users by using only a secure one-way hash function and bitwise XOR operations.
- Due to lack of sensor node resources such as memory storage space, we aim to reduce the numbers of authen-

Manuscript received March 19, 2024; revised April 19, 2024. Date of publication May 16, 2024. Date of current version May 16, 2024. The associate editor prof. Toni Perković has been coordinating the review of this manuscript and approved it for publication.

M. Benfilali and A. Gafour are with the EEDIS Laboratory, Djillali Liabes University, Sidi Bel Abbes, Algeria (e-mails: benfilali.mostefa@univ-bechar.dz, gafour1@yahoo.com).

M. Belghachi is with the Tahri Mohamed University, Department of Mathematic and Computer Sciences, Bechar, Algeria (e-mails: belghachi.mohamed@univ-bechar.dz).

Digital Object Identifier (DOI): 10.24138/jcomss-2023-0068

tication parameters stored in the sensor node's memory.

- Through formal analysis employing BAN-Logic and Real-Or-Random models, we demonstrated that UAWSNA-IoT guarantees both mutual authentication and session key security respectively. Additionally, informal assessment verifies its ability to withstand various known attacks.
- UAWSNA-IoT is capable of integrating the additional sensor nodes as needed, ensuring scalability to meet growing service demands.
- UAWSNA-IoT achieves higher security with acceptable computational, storage space and communication cost compared to related schemes [27], [14], [20], [25], [6], [35], and [38].

The remainder of this paper is organized as follows: We review some related literature on existing schemes in section II. In section III, the system model is presented to provides the information about the network model and the threat assumptions against UAWSNA-IoT. In Section IV, we presents an explanation of our scheme UAWSNA-IoT, which includes four phases such as: registration phase, Login/Authentication phase, changing passwords phase, and adding new sensor nodes phase. We provide a formal and informal security analysis of UAWSNA-IoT in section V. Finally, section VI conducts a comparative evaluation of UAWSNA-IoT and related protocols [27], [14], [20], [25], [6], [35] and [38], considering the costs of computational, storage and communication.

II. Related Works

This section discusses recent works focusing on authentication schemes to protect a Wireless Sensor Network (WSN) from unauthorized access.

The use of passwords for remote authentication was initially proposed by authors in [10]. This technique relies on one-way hash functions and authentication through session keys and signatures. The utilization of session keys, signatures, and location privacy plays a significant role in addressing specific security vulnerabilities [9, 11, and 36]. In [12] authors introduce the user-authentication protocol in Wireless Sensor Networks (WSNs) using lightweight hash functions and symmetric cryptosystems. However, the authors in [3] identified vulnerabilities in this protocol, including stolen verifier, replay, and forgery attacks. To address these issues, the authors propose a new user-authentication scheme that utilized passwords managed and controlled by a gateway. This approach gained popularity and became widely adopted in authentication systems. However, it lacked mutual authentication and session key security. The mutual authentication is important for verifying the legitimacy of the sender's identity during a current session. Recent research has focused on protecting user identity by using user anonymity to hide their real identities [14], [15], [17], [18]. Several techniques have been explored, including the use of randomly selected strings as pseudo-identities for users [16]. However, these methods are vulnerable to user tracking attacks, especially when multiple sessions use the same pseudo-identity. To improve security, it is recommended that each new session generates a fresh and random string to verify the user's true identity. In 2013, the authors in [19] propose an authentication method for WSNs, where temporary credentials are hashed using a one-way hash function. The temporary credentials used in this approach serve as reference information and include a timestamp and user-identity. In 2015, the authors in [16] pointed out the vulnerabilities present in the protocol introduced in [19], highlighting that the scheme is susceptible to attacks user-tracking and identity-guessing attacks. Subsequently, in 2017, researchers in [17] declared that the two-factor authentication mechanism proposed in [16] is exposed to offline guessing and desynchronization attacks, leading to its lack of security. In 2021, the authors in [27] proposed a three-factor authentication scheme for wireless sensor networks in the context of IoT. Nevertheless, the authors in [14] highlighted that this scheme is susceptible to stolen-verifier attacks and lacking perfect forward secrecy. Alternatively, the authors in [14] introduce a secure anonymous three-factor authentication system utilizing elliptic curve cryptography. Regrettably, in this scheme the compromise of a sensor node by an adversary can lead to the retrieval of the user's identity. In broad terms, protocols that do not employ the Diffie-Hellman key exchange algorithm for session key generation are generally unable to attain perfect forward secrecy [23]. Recently, the authors in [6] introduced a three-factor authentication scheme for wireless sensor networks, utilizing elliptic curve cryptography. Nevertheless, their scheme is susceptible to replay attacks, sensor node capture attacks, and off-line password guessing attacks. Additionally, it lacks the capability to uphold session key secrecy, perfect forward secrecy, anonymity, and unlinkability. In 2021, the authors in [35] introduce an authentication scheme for wireless sensor networks in smart cities. This protocol aimed to resolve various existing several flaws in scheme proposed in [42], including vulnerability to offline password guessing attacks and impersonation attacks, along with the absence of session key secrecy, identity unlinkability, and perfect forward secrecy. In 2022, the authors in [22] propose a security-enhanced two-factor authentication scheme for WSN in IoT environment based on ECC, and apply the formal verification using ”ProVerif tool” to prove the security of the proposed scheme. In 2023, the author in [38] proposes a wireless sensor network authentication and key-agreement scheme for IoT that uses multiple gateways. However, the scheme has potential vulnerabilities, such as susceptibility to replay and man-in-the-middle attacks.

III. The System Model

In this section, we present the network and adversary model. The notations used in UAWSNA-IoT are defined in the table I.

A. Network Model

The network model comprise four participants : System Administrator (*SA*), Gateway, User, and Sensor Nodes. The user and sensor nodes undergo registration with the gateway via a secure channel. After the registration phase, a process of mutual authentication is initiated among the entities: User,

TABLE I
LIST OF NOTATIONS USED IN UAWSNA-IOT.

Notation	Description
SA	System Administrator
SN_i	ith Sensor Node
U_k	kth User
GWN_j	jth Gateway
ID_{gwn_j}	Identities of the jth Gateway
IDS_i	Identities of the ithSensor Node
IDU_k	Identities of the kth User
σ_{G_j}	160 bits Master Private key of jth GWN_j
α_{G_j}	160 bits Mask Key of jth GWN_j
n	160 bits public parameter chosen by *SA*
$PIDS_{\mathbf{i}}$	The pseudo-identity of SN_i
$PIDU_k$	The pseudo-identity of U_k
PID_{GWN_j}	The pseudo-identity of $\mathrm{GWN_j}$
SC_k	Smart Card of kth User
ρ_i	private key shared with SN_i and GWN_j
ρ_k	private key shared with $\mathrm{U_k}$ and $\mathrm{GWN_j}$
N_1	160 bits random numbers generated by U_k
N_2	160 bits random numbers generated by SN_i
ST_1	Current timestamp
$\triangle T$	Maximum time threshold of accepting messages
PW_k	KthPassword associated to user U_k
$h(.)$	one-way hash function, where h: $\{0,\ 1\}^* \rightarrow \mathrm{Z_n^*}$
SK_{ik}	Session Key shared between SN_i and U_k

Gateway, and Sensor Node, respectively. Once the authentication phase has been successfully completed, the communication between users and sensor is established over the public channel using the shared session key. The network model is visually represented in Figure 1.

- *System Administrator* (*SA*): *SA* is responsible for generating the confidential parameters, registration, and updating the gateway (GWN_j). Additionally, *SA* is responsible to registering of the new U_k's and SN_i's after the network deployment.
- *User (U_k) :* During the registration phase, the user (U_k) with a Smart Card (SC_k) receives their secret parameters from the gateway (GWN_j). U_k must first be verified by the gateway before being able to access and communicate with a sensor node (SN_i).
- *Gateway (GWN_j):* GWN_j is considered a trusted entity responsible for registering every user and sensor node. GWN_j is responsible to generates the secret parameters for each user (U_k) and sensor node (SN_i) based on their respective identities
- *Sensor Node (SN_i):* During registration, SN_i receives its secret key from GWN_j. After confirming the legitimacy of U_k through GWN_j, SN_i and U_k establish a session key (SK_{ik}) to ensure the security of future communications.

B. Adversary Model

In accordance with the attack model suggested in [26], the adversary ”A” model against our protocol ”UAWSNA-IoT,” is delineated as follows:

- “A” has the capability to intercept all transmitted messages, he/she enable to capturing, replaying, modifying, and rerouting of messages.

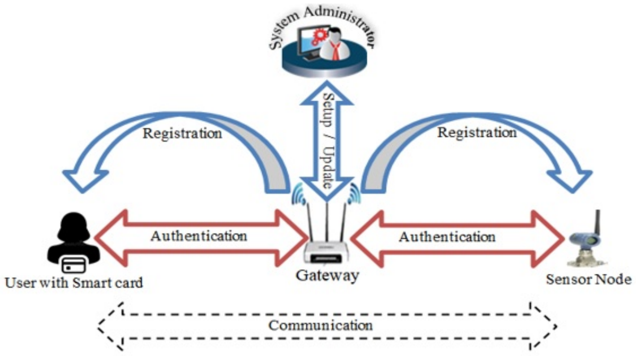

Fig. 1. The Network model

- The protocol is susceptible to off-line password guessing attacks, potentially leading to compromise of user identities.
- The sensor nodes and smart cards are vulnerable to capture attack, allowing the extract all information stored within the captured nodes and stolen smart cards.
- The trustworthy entities are specifically the system administrator and gateway.

IV. PROPOSED AUTHENTICATION SCHEME

In this section, we provide a comprehensive explanation of UAWSNA-IoT. UAWSNA-IoT consists of five phases namely the System setup phase, users/sensor nodes registration phase, login/authentication phase, password renewal phase, and new sensor node addition phase.

A. System Setup Phase

The System Administrator (*SA*) initiates the generation of essential parameters for setup of the gateway (GWN_j). The steps of this phase are described below:

$Step1$: *SA* chooses the Master private key (σ_{G_j}), mask key (α_{G_j}), and the public system parameter (*n*), each with a size of 160 bits.

$Step2$: *SA* chooses a secure one-way hash function $h\ :\ \{0,\ 1\}^* \rightarrow \mathrm{Z_n^*}$, and selects an identity (ID_{gwn_j}) for the specific gateway (GWN_j) and proceeds to calculate its pseudo-identity, such as: $PID_{GWN_j} = h(ID_{gwn_j} \parallel \alpha_{G_j})$.

$Step3$: *SA* stores this information (σ_{G_j}, α_{G_j}, ID_{gwn_j}, *h(.)*, *n*) in its database (DB_{SA}) and sends this information (σ_{G_j}, α_{G_j},PID_{GWN_j}, *h(.)*, *n*) to GWN_j to storing them secretly in its database (DB_{GWN_j}).

$Step4$: Afterward, GWN_j publishes these parameters *h(.)*, *n*) to sensor nodes and users in registration phase. Finally, *SA* and GWN_j use a shared Master Private Key (σ_{G_j}) as symmetric key for the future communication.

B. Registration Phase

This phase is divided into two parts: The Sensor node registration and User registration, as depicted in Figure 2 and Figure 3 respectively. The Both registrations take place over a secure channel.

B.1. Sensor Node registration

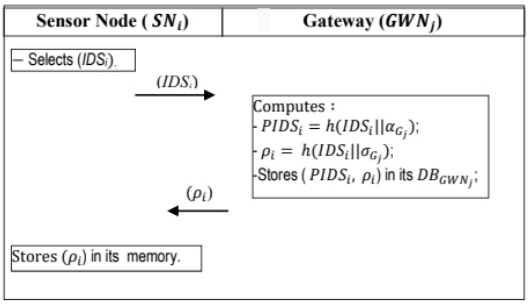
Sensor Node (SN_i)
Gateway (GWN_j)
– Selects (IDS_i).
(IDS_i)
Computes :
- $PIDS_i = h(IDS_i||\alpha_{G_j})$;
- $\rho_i = h(IDS_i||\sigma_{G_j})$;
-Stores ($PIDS_i$, ρ_i) in its DB_{GWN_j};
(ρ_i)
Stores (ρ_i) in its memory.

Fig. 2. Sensor Node Registration phase.

$Step1$:SN_i chooses a unique identity IDS_i (i = 1 to L, L: number of sensor nodes in WSN). After that, SN_i sends its IDS_i to GWN_j;

$Step2$: Upon receiving the message, GWN_j compute: $\rho_i = h(IDS_i \parallel \sigma_{G_j})$ and $PIDS_i = h(IDS_i \parallel \alpha_{G_j})$;

$Step3$: Ultimately, GWN_j stores (ρ_i, $PIDS_i$) within its database (BD_{GWN_j}) and sends (ρ_i) to SN_i, to saving them discreetly in its memory.

B.2. User Registration Phase

$Step1$: User chooses its identity (IDU_k) along with its password (PW_k) , and secretly transmit his (IDU_k) to GWN_j;

$Step2$: After receiving the message, GWN_j computes: $PIDU_k = (IDU_k \parallel \alpha_{G_j})$; ρ_k=$h(IDU_k \| \sigma_{G_j})$ and $vu_k = h(PIDU_k \| PID_{GWN_j} \| \rho_k)$. Next, GWN_j secretly stores this information (ρ_k, $PIDU_k$, vu_k) in its database (DB_{GWN_j});

$Step3$: After that, GWN_j chooses the set of $PIDS_i$ of SN_i, sends secretly ($PIDU_k$, $PIDS_i$, ρ_k, vu_k, PID_{GWN_j}) to U_k and sends these information (ID_{gwn_j}, IDU_k) to SA to saving them in its DB_{SA};

$Step4$: As soon as, U_k receives ($PIDU_k$, $PIDS_i$, ρ_k, PID_{GWN_j}, vu_k), U_k stores them in its smart card (SC_k). Then U_k inputs its IDU_k and PW_k, after that, SC_k compute: *MPW* = $h(IDU_k \| PW_k \| \rho_k)$; $\rho^s_{\mathrm{k}} = h(IDU_k \| PW_k) \oplus \rho_k$; and $vu^s_k = h(IDU_k \| PW_k) \oplus vu_k$. Finally, SC_k covertly retains (ρ^s_{k}, vu^s_k, *MPW*) in its memory by replacing ρ_k with ρ^s_{k}, and vu_k with vu^s_k. Hence, the ultimate information's stored in SC_k encompasses ($PIDU_k$, ρ^s_{k}, $PIDS_i$, PID_{GWN_j}, vu^s_k, *MPW*).

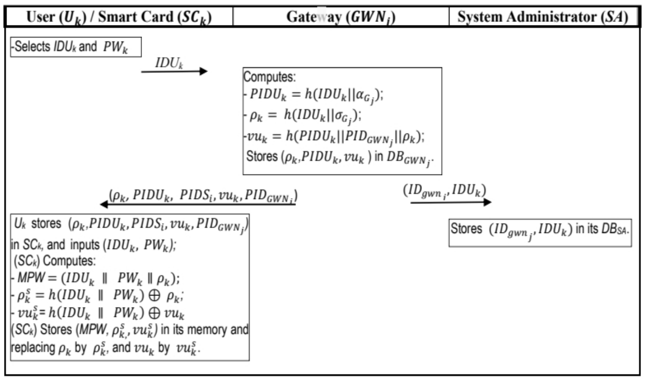
User (U_k) / Smart Card (SC_k)
Gateway (GWN_j)
System Administrator (SA)
-Selects IDU_k and PW_k
IDU_k
Computes:
- $PIDU_k = h(IDU_k||\alpha_{G_j})$;
- $\rho_k = h(IDU_k||\sigma_{G_j})$;
-$vu_k = h(PIDU_k||PID_{GWN_j}||\rho_k)$;
Stores (ρ_k,$PIDU_k$, vu_k) in DB_{GWN_j}.
(ρ_k, $PIDU_k$, $PIDS_i$, vu_k, PID_{GWN_j})
(ID_{gwn_j}, IDU_k)
U_k stores (ρ_k,$PIDU_k$,$PIDS_i$,vu_k,PID_{GWN_j}) in SC_k, and inputs (IDU_k, PW_k);
(SC_k) Computes:
- $MPW = (IDU_k \parallel PW_k \parallel \rho_k)$;
- $\rho^s_k = h(IDU_k \parallel PW_k) \oplus \rho_k$;
- vu^s_k= $h(IDU_k \parallel PW_k) \oplus vu_k$
(SC_k) Stores (MPW, ρ^s_k, vu^s_k) in its memory and replacing ρ_k by ρ^s_k, and vu_k by vu^s_k.
Stores (ID_{gwn_j}, IDU_k) in its DB_{SA}.

Fig. 3. User Registration phase.

C. Login/Authentication Phase

The login/authentication phase between the User , gateway, and Sensor Node is illustrated in Figure 4. This phase can be described as follows:

$Step1$: U_k inserts his Smart Card (SC_k) into the card reader and inputs its (IDU_k, PW_k); SC_k compute: $\rho_k = \rho^s_{\mathrm{k}} \oplus$ $h(IDU_k \| PW_k)$; MPW'=$h(IDU_k \| PW_k \| \rho_k)$. Subsequently, SC_k checks (MPW' = *MPW*)?, if not equal, then SC_k revokes the login/authentication phase.

- Otherwise, SC_k prompts U_k to entering the Pseudo-identity ($PIDS_i$) associated with SN_i that he/she intends to establish communication with it. Once U_k selects $PIDS_i$, SC_k compute: vu_k = $h(IDU_k \| PW_k) \oplus vu^s_k$ and choose a random number $N_1 \in z^*_n$ to compute: $V_1 = vu_k \oplus N_1$; V_2 = $h(PIDU_k \| PID_{GWN_j} \| \rho_k \| N_1)$ $\oplus$ $PIDS_i$; V_3 = h $(PIDU_k \| PIDS_i \| PID_{GWN_j} \|$ ρ_k $\| N_1)$.
- Finally, SC_k selects current TS_1 and sends $Messg_1$ (V_1, V_2, V_3, $PIDU_k$, TS_1) to GWN_j over a public channel.

$Step2$: Upon receiving $Messg_1$, GWN_j checks TS_1 (*Time* – $TS_1 \leq \triangle$T, *Time*: represents the current time at which a message is received). If not true, the login request is ignored. Otherwise, GWN_j retrieves (ρ'_k, vu'_k) associated with $PIDU'_k$, stored in its database (DB_{GWN_j}) .

- After that, GWN_j computes: $N'_1 = V_1 \oplus vu'_k$; $PIDS'_i = V_2 \oplus h(PIDU_k \parallel PID_{GWN_j} \parallel \rho'_k \| N'_1)$ and $V'_3 = h(PIDU_k \| PIDS'_i \parallel PID_{GWN_j} \| \rho'_k \| N'_1)$;
- After that, GWN_j checks equality ($V_3 = V'_3$?), if not equal, then GWN_j rejects the authentication request. Otherwise, GWN_j retrieves ρ'_i in its DB_{GWN_j} according to values $PIDS'_i$, and computes: V_4= $h(PIDS'_j \parallel PIDU_k \parallel PID_{GWN_j} \| N'_1) \oplus \rho'_i$; V_5 =$h(PIDS'_i \parallel PIDU_k \parallel PID_{GWN_j} \parallel N'_1) \oplus N'_1$; $V_6 = h(h(PIDS'_i \parallel PIDU_k \parallel PID_{GWN_j} \parallel N'_1) \parallel PIDU'_k)$;
- Finally, GWN_j chooses the current (TS_2) and sends $Messg_2$ (V_4, V_5, V_6, $PIDU'_k$, TS_2) to SN_i via a public channel.

$Step3$: Once SN_i receives $Messg_2$, SN_i checks (*Time* – $TS_2 \leq \triangle$T ?), if not true, SN_i rejects the request message. Otherwise, SN_i compute: $h^*(PIDS'_i \parallel PIDU'_k \parallel PID_{GWN_j} \| N'_1) = V_4 \oplus \rho_i$; $N''_1 = V_5 \oplus h^*(PIDS'_i \| PIDU'_k \| PID_{GWN_j} \| N'_1)$ and V'_6= $h(h^*(PIDS'_i \parallel PIDU'_k \parallel PID_{GWN_j} \| N'_1) \| PIDU''_k)$.

- After that, SN_i checks the legality (V'_6= V_6?), if not equal ($V'_6 \neq V_6$), SN_i rejects the request message. Otherwise, SN_i randomly chooses a random number $N_2 \in z^*_n$ and current TS_3. After, SN_i compute : $V_7 = \rho_i \oplus N_2$; $\gamma = N''_1 \oplus N_2$; $\rho^{new}_i = h(\rho_i \| N_2)$; $V_8 = h$ $(h^*(PIDS'_i \parallel PIDU'_k \parallel PID_{gwn_j} \| N'_1) \parallel N_2 \parallel \gamma)$; $SK_{ik} = h$ $(h^*(PIDS'_i \parallel PIDU'_k \parallel PID_{GWN_j} \parallel N'_1) \parallel \gamma)$.
- Finally, SN_i replaces ρ_i with ρ^{new}_i in its memory, chooses the current TS_3, and sends $Messg_3$ (V_7, V_8, TS_3) to GWN_j.

$Step4$: Upon receiving $Messg_3$, GWN_j checks (*Time* – $TS_3 \leq \triangle$T ?), if not true, GWN_j aborts the session.

Otherwise, GWN_j compute: $N_2^{'}= V_7 \oplus \rho_i$; $\gamma^{'} = N_1 \oplus N_2^{'}$; $V_8^{'}= h(h^*(PIDS_i^{'} \parallel PIDU_k^{'} \parallel PID_{GWN_j} \parallel N_1^{'}) \| N_2^{'} \| \gamma^{'})$.

- After that, GWN_j checks equality ($V_8^{'} = V_8$?), if not equal, then GWN_j rejects the request message. Otherwise, GWN_j compute $\rho_i^{new}= h(\rho_i^{'} \| N_2)$; $SK_{ik}= h(h^*(PIDS_i^{'} \parallel PIDU_k^{'} \parallel PID_{GWN_j} \| N_1^{'}) \parallel \gamma^{'})$; $V_9 = N_2^{'} \oplus$ h*$(PIDS_i^{'} \parallel PIDU_k \parallel PID_{GWN_j} \| N_1^{'})$; $PIDU_k^{new} = h(PIDU_k \parallel N_2^{'}$), and replaces $PIDU_k$ with $PIDU_k^{new}$ and $\rho_i^{'}$ associated to $PIDS_i^{'}$ with ρ_i^{new} in its DB_{GWN_j}.
- Finally, GWN_j chooses current TS_4 and sends $Messg_4(V_5, V_8, V_9, TS_4)$ to U_k.

$Step5$: Upon receiving $Messg_4$, SC_k checks (*Time* $- TS_4 \leq$ $\triangle$T ?), if not true, the request message is ignored. Otherwise, SC_k compute: $h^*(PIDS_i^{'} \| PIDU_k^{'} \| PID_{GWN_j} \| N_1^{'}) = V_5 \oplus N_1$; $N_2^{"}= V_9 \oplus h^*(PIDS_i^{'} \| PIDU_k^{'} \| PID_{GWN_j} \| N_1^{'})$; $\gamma^{"} = N_1 \oplus N_2^{"}$; $V_8^{'} = h\ (h^*(PIDS_i^{'} \parallel PIDU_k^{'} \parallel PID_{GWN_j} \parallel N_1^{'}) \parallel N_2^{"} \parallel \gamma^{''})$.

- After, SC_k Cheeks ($V_8^{'} = V_8$?), if not equal, then SC_k rejects the request message. Otherwise, SC_k computes: $SK_{ki} = h(h^*(PIDS_i^{'} \| PIDU_k^{'} \| PID_{GWN_j} \| N_1^{'}) \| \gamma^{'})$; $PIDU_k^{New} = h(PIDU_k \parallel N_2^{"})$ and replaces $PIDU_k$ with $PIDU_k^{New}$ in SC_k memory;

D. Password Renewal Phase

The user (U_k) has the autonomy to change their password(PW_k) at regular intervals, without involving GWN_j or SA. This process is solely between U_k and their own SC_k and is optional. The following steps are involved in this phase:

1) U_k insert its SC_k into the card reader, then he/she enters his IDU_k and old password PW_k. SC_k needs to compute the private key ρ_k such as: $\rho_k = \rho_k^s \oplus h(IDU_k \parallel PW_k)$ to compute the mask: $MPW^{'}=h(IDU_k \parallel PW_k \parallel \rho_k)$.
2) After that, SC_k check the equality ($MPW^{'}= MPW$?), if not equal, then SC_k rejects the password change request. Otherwise, SC_k will ask U_k to enter the new password PW_k^{new} according to its choices.
3) After U_k enters his new password PW_k^{new}, then SC_k calculates the new mask MPW^{new} according to PW_k^{new} such as: $MPW^{new}= h(IDU_k \| PW_k^{new} \| \rho_k)$;
4) Afterwards, SC_k computes: ρ_k^s and vu_k^saccording to the new password PW_k^{new}such that: $\rho_k^{s\,new}= h(IDU_k \| PW_k^{new}) \oplus \rho_k$; $vu_k^{s\,new} = h(IDU_k \| PW_k^{new}) \oplus vu_k$
5) Finally, SC_k stores these new values ($\rho_k^{s\,new}$, $vu_k^{s\,new}$, MPW^{new}) in its memory secretly by replacing the old values.

E. Sensor Node Addition phase

To achieve scalability, UAWSNA-IoT needs to be adapted to dynamically integrate the new sensor nodes ${SN_i}^{new}$. In this phase, The System Administrator (SA) is responsible for registering new sensor node ${SN_i}^{new}$ even in the absence of gateway (GWN_j). The steps followed by SA are as follows:

- SA chooses the appropriate GWN_j, where ${SN_i}^{new}$ will be deployed in order to calculate their confidential parameters.
- Subsequently, both SA and ${SN_i}^{new}$ follow the identical steps as outlined in the sensor node registration phase mentioned above in the section IV.B.1.
- Upon the registration phase of ${SN_i}^{new}$ is completed, SA utilizes the shared symmetric key: $Key_{(SA-GWN_j)}$ $=\sigma_{G_j}$ to encrypt these information (${PIDS_i}^{new}$, and ρ_i^{new}) associated with new ${SN_i}^{new}$ and transmits them to the corresponding GWN_j.
- Upon receiving this information, GWN_j decrypt them and store its in DB_{GWN_j}. Ultimately, SA deploys ${SN_i}^{new}$ in the chosen capture area. Following this deployment U_k is promptly notified about this new addition to including the pseudo-identity (${PIDS_i}^{new}$) associated with ${SN_i}^{new}$in his smart card SC_k.

V. Security Analysis

In this section, we conduct a comprehensive security evaluation of UAWSNA-IoT using both formal and informal analyses.

A. Formal Security Analysis using (ROR) Model

The Real-Or-Random (ROR) model [8] is employed to evaluate the session key security in UAWSNA IoT protocol's. In this model, the network is vulnerable to various attacks conducted by an adversary "A", including eavesdropping, capturing, inserting, and deleting messages [26]. In the security analysis, we use symbols $\prod_{U_k}^t$,$\prod_{G_j}^u$ and $\prod_{SN_i}^v$ to represent specific instances denoted by *t*, *u* and *v* respectively, which act as oracles in the system. We apply the principles of the ROR model to UAWSNA-IoT scheme, where "A" possesses the capability to execute different attacks, as indicated by the following queries:

- *Execute*($\prod_{U_k}^t$, $\prod_{G_j}^u$, $\prod_{SN_i}^v$): "A" performs this type of query in order to intercept the messages exchanged between the oracles of legitimate participants. This query models a passive-type attack.
- *Send* ($\prod^x, M$): The goal of this query is to simulate an active attack. By executing this query, "A" is capable of sending a message *M* to a participating instance $\prod^x$ and receiving a response message in return.
- *Test* (*x*,*i*): If the oracle accepted and has the session key (SK_{ik}), then a bit b is chosen randomly. If $b = 1$, then "A" gets the freshly SK_{ik}, else, if $b = 0$, it means "A" gets a random SK_{ik}. However, if b$\neq$ 0 and $b \neq 1$, "A" gets a NULL value. This query is used to model an attacker's ability to distinguish between a real and a random SK_{ik}. To ensure the security of SK_{ik} in UAWSNA-IoT, "A" can never distinguish between a random and the real session key (SK_{ik}) generated as a result.
- *Reveal* (x, i): If the oracle $\prod_x^i$ is accepted and has a session key (SK_{ik}) , then we give SK_{ik} to "A". This model simulates the robustness of UAWSNA-IoT, i.e. that is disclosing a SK_{ik} affects only the current session.

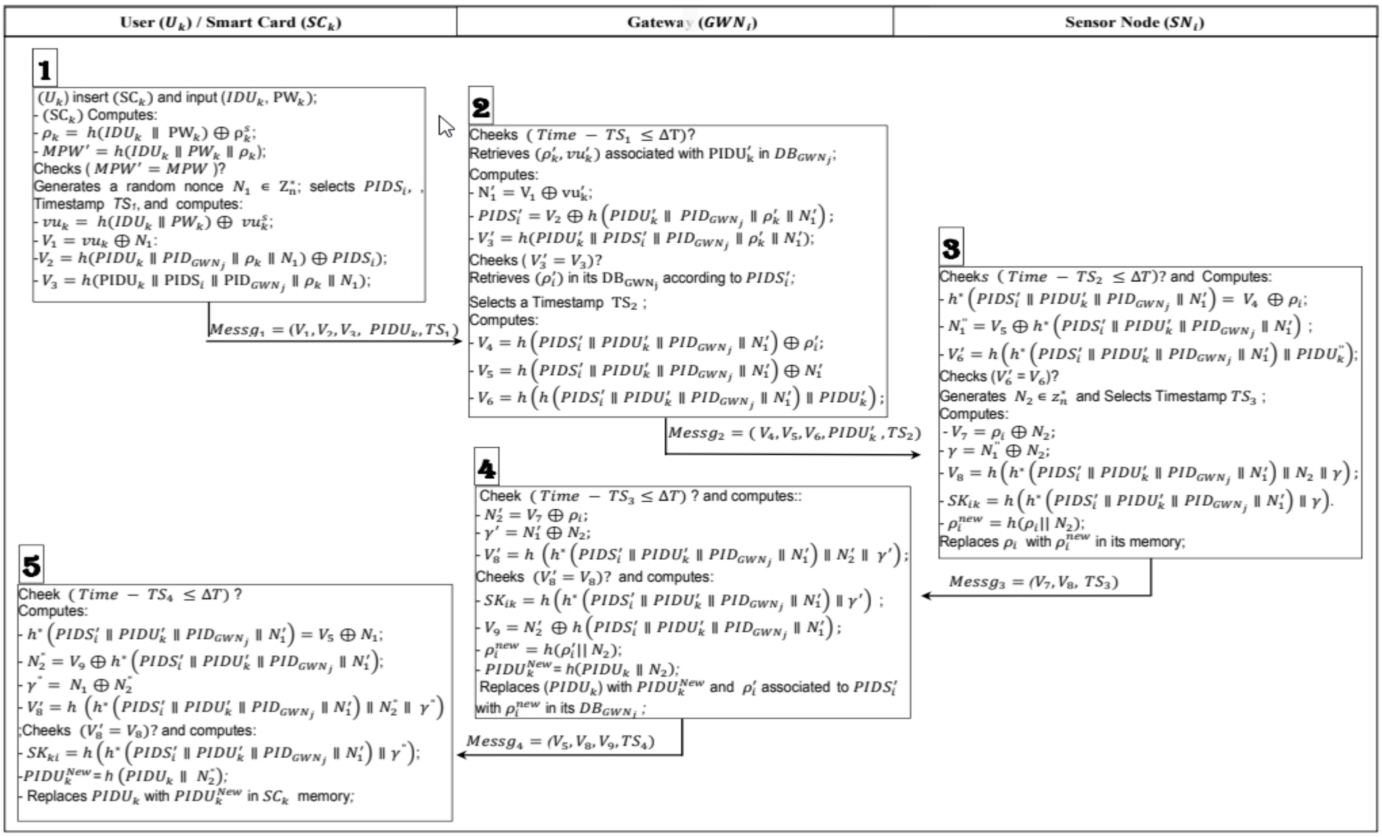
User (U_k) / Smart Card (SC_k)
Gateway (GWN_j)
Sensor Node (SN_i)
1
(U_k) insert (SC_k) and input (IDU_k, PW_k);
- (SC_k) Computes:
- $\rho_k = h(IDU_k \parallel PW_k) \oplus \rho_k^s$;
- $MPW' = h(IDU_k \parallel PW_k \parallel \rho_k)$;
Checks ($MPW' = MPW$)?
Generates a random nonce $N_1 \in Z_n^*$; selects $PIDS_i$, Timestamp TS_1, and computes:
- $vu_k = h(IDU_k \parallel PW_k) \oplus vu_k^s$;
- $V_1 = vu_k \oplus N_1$;
- $V_2 = h(PIDU_k \parallel PID_{GWN_j} \parallel \rho_k \parallel N_1) \oplus PIDS_i$);
- $V_3 = h(PIDU_k \parallel PIDS_i \parallel PID_{GWN_j} \parallel \rho_k \parallel N_1)$;
$Messg_1 = (V_1, V_2, V_3, PIDU_k, TS_1)$
2
Cheeks ($Time - TS_1 \leq \Delta T$)?
Retrieves (ρ_k', vu_k') associated with $PIDU_k'$ in DB_{GWN_j};
Computes:
- $N_1' = V_1 \oplus vu_k'$;
- $PIDS_i' = V_2 \oplus h\left(PIDU_k' \parallel PID_{GWN_j} \parallel \rho_k' \parallel N_1'\right)$;
- $V_3' = h(PIDU_k' \parallel PIDS_i' \parallel PID_{GWN_j} \parallel \rho_k' \parallel N_1')$;
Cheeks ($V_3' = V_3$)?
Retrieves (ρ_i') in its DB_{GWN_j} according to $PIDS_i'$;
Selects a Timestamp TS_2 ;
Computes:
- $V_4 = h\left(PIDS_i' \parallel PIDU_k' \parallel PID_{GWN_j} \parallel N_1'\right) \oplus \rho_i'$;
- $V_5 = h\left(PIDS_i' \parallel PIDU_k' \parallel PID_{GWN_j} \parallel N_1'\right) \oplus N_1'$
- $V_6 = h\left(h\left(PIDS_i' \parallel PIDU_k' \parallel PID_{GWN_j} \parallel N_1'\right) \parallel PIDU_k'\right)$;
$Messg_2 = (V_4, V_5, V_6, PIDU_k', TS_2)$
3
Cheeks ($Time - TS_2 \leq \Delta T$)? and Computes:
- $h^*\left(PIDS_i' \parallel PIDU_k' \parallel PID_{GWN_j} \parallel N_1'\right) = V_4 \oplus \rho_i$;
- $N_1'' = V_5 \oplus h^*\left(PIDS_i' \parallel PIDU_k' \parallel PID_{GWN_j} \parallel N_1'\right)$;
- $V_6' = h\left(h^*\left(PIDS_i' \parallel PIDU_k' \parallel PID_{GWN_j} \parallel N_1'\right) \parallel PIDU_k''\right)$;
Checks ($V_6' = V_6$)?
Generates $N_2 \in z_n^*$ and Selects Timestamp TS_3 ;
Computes:
- $V_7 = \rho_i \oplus N_2$;
- $\gamma = N_1'' \oplus N_2$;
- $V_8 = h\left(h^*\left(PIDS_i' \parallel PIDU_k' \parallel PID_{GWN_j} \parallel N_1'\right) \parallel N_2 \parallel \gamma\right)$;
- $SK_{ik} = h\left(h^*\left(PIDS_i' \parallel PIDU_k' \parallel PID_{GWN_j} \parallel N_1'\right) \parallel \gamma\right)$.
- $\rho_i^{new} = h(\rho_i \parallel N_2)$;
Replaces ρ_i with ρ_i^{new} in its memory;
$Messg_3 = (V_7, V_8, TS_3)$
4
Cheek ($Time - TS_3 \leq \Delta T$) ? and computes::
- $N_2' = V_7 \oplus \rho_i$;
- $\gamma' = N_1' \oplus N_2$;
- $V_8' = h\left(h^*\left(PIDS_i' \parallel PIDU_k' \parallel PID_{GWN_j} \parallel N_1'\right) \parallel N_2' \parallel \gamma'\right)$;
Cheeks ($V_8' = V_8$)? and computes:
- $SK_{ik} = h\left(h^*\left(PIDS_i' \parallel PIDU_k' \parallel PID_{GWN_j} \parallel N_1'\right) \parallel \gamma'\right)$;
- $V_9 = N_2' \oplus h\left(PIDS_i' \parallel PIDU_k' \parallel PID_{GWN_j} \parallel N_1'\right)$;
- $\rho_i^{new} = h(\rho_i' \parallel N_2)$;
- $PIDU_k^{New} = h(PIDU_k \parallel N_2)$;
Replaces ($PIDU_k$) with $PIDU_k^{New}$ and ρ_i' associated to $PIDS_i'$ with ρ_i^{new} in its DB_{GWN_j} ;
$Messg_4 = (V_5, V_8, V_9, TS_4)$
5
Cheek ($Time - TS_4 \leq \Delta T$) ?
Computes:
- $h^*\left(PIDS_i' \parallel PIDU_k' \parallel PID_{GWN_j} \parallel N_1'\right) = V_5 \oplus N_1$;
- $N_2'' = V_9 \oplus h^*\left(PIDS_i' \parallel PIDU_k' \parallel PID_{GWN_j} \parallel N_1'\right)$;
- $\gamma'' = N_1 \oplus N_2''$
- $V_8' = h\left(h^*\left(PIDS_i' \parallel PIDU_k' \parallel PID_{GWN_j} \parallel N_1'\right) \parallel N_2'' \parallel \gamma''\right)$
;Cheeks ($V_8' = V_8$)? and computes:
- $SK_{ki} = h\left(h^*\left(PIDS_i' \parallel PIDU_k' \parallel PID_{GWN_j} \parallel N_1'\right) \parallel \gamma''\right)$;
- $PIDU_k^{New} = h(PIDU_k \parallel N_2'')$;
- Replaces $PIDU_k$ with $PIDU_k^{New}$ in SC_k memory;

Fig. 4. Login/Authentication phase.

- *CorruptSC* ($\prod_{U_k}^t$): The goal of this query is to model the smart card (SC_k) loss attack. This query produces the output to indicate that “A” extracts all parameters stored in SC_k.

- Theorem 1

Let A, try to get the session key of our UAWSNA-IoT protocol in polynomial time t by following Real-Or-Random (ROR) model. Let $Adv^{(UAWSNA-IoT)}(A)$, represents the probability that the adversary ”A” will be successful in breaking the session key. Let q_{hash}^2, q_{send}, HASH, represent the number of hash requests, number of send requests and hash function space domain $h(.)$ respectively. The parameters s' and C' are the Zipf's parameters defined in[28].

- The formal proof

Following the proof technique used in [29], [30], we perform four game rounds called $Game_j$, where j ∈ [0,3]. Let $Succ_{(A,game_j)}$ be an event where ”A” can guess the correct bit *b* in the game $Game_j$ with a probability equal to Pr[$Succ_{(A,game_j)}$]. The game starts with $Game_0$ which is a real attack, and ends with $Game_3$. We can accomplish $Game_j$ as follows with these parameters:

- $Game_0$: This game models a real attack performed by adversary ”A” against UAWSNA-IoT based on RoR model. Initially, $Game_0$ randomly chooses a bit b, which we can derive as follows:

$$\text{Adv}^{(UAWSNA-IoT)}(A) = \mid 2.Pr[Succ_{(A,game_0)}] - 1 \mid \quad (1)$$

- $Game_1$: We assume that ”A” intercepts the messages $Messg_1\{V_1,\ V_2,\ V_3,\ PIDU_k,\ TS_1\}$, $Messg_2$ $\{\text{V}_4, V_5, V_6, PIDU_k^{'}, TS_2\}$,Messg$_3\{V_7,\ V_8,\ TS_3\}$, and $Messg_4\{V_5,\ V_8,\ V_9,\ TS_4\}$ using Execute ($\prod_{U_k}^t$, $\prod_{G_j}^u$, $\prod_{SN_i}^v$) query. Then “A” executes *Test()* and *Reveal()* queries to obtain SK_{ik}. SK_{ik} is computed using the following secret parameters: $PIDU_k$, $PIDS_i$, PID_{GWN_j} and γ = $N_1^{'}$ ⊕ N_2 such as SK_{ik} = $h(h^*(PIDS_i^{'}$ ∥ $PIDU_k^{'}$ ∥ PID_{gwn_j}∥ $N_1^{'})$ ∥ $\gamma)$. So “A” needs the pseudo-identity's PID_{GWN_j} and $PIDS_i$ associated with GWN_j, and SN_i respectively. In addition, the random numbers N_1 and N_2 as the main parameters to calculate SK_{ik}. “A” is unable to compute a real SK_{ik} without knowledge of these secret parameters. This means $Game_0$ and $Game_1$ are indistinguishable. Therefore, the probability that “A” is a winner of $Game_1$ remains similar to $Game_0$.

$$Pr[\text{Succ}_{(A,game_1)}] = \text{Pr}[Succ_{(A,game_0)}] \quad (2)$$

- $Game_2$: In this game, ”A” perform the *Send()* and HASH() queries which are an active attack. So, ”A” uses $Messg_1$, $Messg_2$, $Messg_3$ and $Messg_4$ exchanged to get SK_{ik}. But, these messages contain the values which are embedded in HASH*(.)* query. More precisely, the values $\{V_2,\ V_3,\ V_4,\ V_5,\ V_6\}$ are calculated according to random number N_1. Additionally, the values V_7, V_8, and V_9 are computed using N_1 and N_2. We use random numbers N_1 and N_2 to prevent collision between the sessions. In addition, UAWSNA-IoT use the timestamps (TS_1, TS_2, TS_3, and TS_4,) associated with the exchanged messages. Therefore, ”A” cannot get the collision cases in the hash function according to the birthday paradox [34], we can get the following equation:

$$\begin{aligned} \mid Pr[Succ_{(A,game_2)}] - Pr[Succ_{(A,game_1)}] \mid \leq \\ \frac{q_h^2}{\mid HASH \mid} \end{aligned} \quad (3)$$

- $Game_3$: In this game, Adversary ”A” employs the *CorruptSC()* query to simulate stolen smart card (SC_k) attacks. ”A” has the capability of obtaining the following information: *MPW*, ρ_k^s, vu_k^s, $PIDS_i$, and PID_{GWN_j} stored in SC_k by using the power analysis attack. Assuming that the password PW_k has low entropy, “A” tries to use a brute-force attack with an online dictionary, exploiting the information extracted from SC_k. However, we assume that our system limits the number of attempts to enter the correct PW_k. Therefore, “A” cannot obtain the required secret information ρ_k, and PW_k from the parameters ρ_k^s, vu_k^s, and *MPW* extracted from the SC_k. This difficulty lies in the adversary inability to know IDU_k and PW_k at the same time, since they are embedded in the *MPW*. Therefore, ”A” cannot distinguish between $Game_2$ and $Game_3$, without knowing IDU_k and PW_k, which is an impossible task. For this reason, we get the result according to Zipf’s law [28].

$$\begin{aligned} \mid Pr[Succ_{(A,game_3)}] - Pr[Succ_{(A,game_2)})] \mid \\ \leq C^{'} q_{send}^{s^{'}} \end{aligned} \quad (4)$$

Afterwards, ”A” acquires the guessed bit b, because the games are over.

$$\Pr[Succ_{(A,game_3)}] = \frac{1}{2} \quad (5)$$

From equations (1) and (2), we deduce the following result:

$$\begin{aligned} \frac{1}{2} Adv^{(UAWSNA-IoT)}(A) &= \\ \mid Pr[Succ_{(A,game_0)}] - \tfrac{1}{2} \mid &= \\ \mid Pr[Succ_{(A,game_1)}] - \frac{1}{2} \mid \end{aligned} \quad (6)$$

We use equations (5) and (6) to easily obtain the following equation:

$$\begin{aligned} \frac{1}{2} Adv^{(UAWSNA-IoT)}(A) = \mid Pr[Succ_{(A,game_1)}] \\ - Pr[Succ_{(A,game_3)}] \mid \end{aligned} \quad (7)$$

We apply the triangular inequality, we easily obtain the following result:

$$\begin{aligned} \frac{1}{2} Adv^{(UAWSNA-IoT)}(A) = \\ \mid Pr[Succ_{(A,game_1)}] - Pr[Succ_{(A,game_3)}] \leq \\ \mid Pr[Succ_{(A,game_1)}] - Pr[Succ_{(A,game_2)}] \mid + \\ \mid Pr[Succ_{(A,game_2)}] - Pr[Succ_{(A,game_3)} \mid \\ \leq \frac{q_h^2}{2 \mid HASH \mid} + C^{'} q_{send}^{s^{'}} \end{aligned} \quad (8)$$

TABLE II
NOTATIONS USED IN BAN-LOGIC PROOF.

NOTATION	SIGNIFICATION
A, B	Two principals
X,Y	Two statements
SK_{ik}	The session key
A $\mid\equiv$ B	A believes B
$\#(B)$	B is fresh
A$\mid\Rightarrow$ B	A control B
A $\mid\sim$ B	A once said B
A $\mid\triangleleft$ X	A receives X
$\{X\}_{Key}$	X is encrypted with Key
A $\overset{K}{\rightleftharpoons}$B	A and B have shared secret key K

Finally, by multiplying the formula (8) by 2, we will obtain the following equation:

$$Adv^{(UAWSNA-IoT)}(A) \leq \frac{q_h^2}{\mid HASH \mid} + 2\ C^{'} q_{send}^{s^{'}}$$

B. Formal Security Analysis Using BAN Logic

In this section we prove that *UAWSNA-IoT* fulfils all conditions to achieve mutual authentication using BAN-logic analysis [39]. The exchange messages :

- $Messg_1$ = $\{V_1$, V_2, V_3, $PIDU_k$, $TS_1\}$;
- $Messg_2$ = $\{V_4$, V_5, V_6, $PIDU_k$, $TS_2\}$;
- $Messg_3$ = $\{V_7$, V_8, $TS_3\}$;
- $Messg_4$= $\{V_5$, V_8, V_9, $TS_4\}$.

- The forms idealized of the exchanged messages:

- $Messg_1 : U_k \rightarrow GWN_j : \{PIDS_i, N_1, TS_1\}_{\rho_k}$;
- $Messg_2$ $GWN_j \rightarrow SN_i$: $\{\mathrm{N}_1, h(PIDS_i^{'} \parallel PIDU_k^{'} \parallel PID_{GWN_j} \parallel \mathrm{N}_1^{'}), TS_2\}_{\rho_i}$;
- $Messg_3 : SN_i \rightarrow GWN_j : \{N_2, TS_3\}_{\rho_i}$;
- $Messg_4$: $GWN_j \rightarrow U_k$: $\{N_2$, $h(PIDS_i^{'} \parallel PIDU_k^{'} \parallel PID_{GWN_j} \parallel N_1^{'}), TS_2\}_{N_1}$.

- BAN-logic rules:

- *Rule 1*: Message-meaning rule: $\frac{A\mid\equiv A\overset{K}{\rightleftharpoons}B, A\mid\triangleleft(X)_K}{A\mid\equiv B\mid\sim X}$
- *Rule 2*: *Nonce-verification rule:* $\frac{A\mid\equiv\#(X), A\mid\equiv B\mid\sim X}{A\mid\equiv B\mid\equiv X}$
- *Rule 3*: *Jurisdiction rule:* $\frac{A\mid\equiv B\mid\Rightarrow X, A\mid\equiv B\mid\equiv X}{A\mid\equiv X}$
- *Rule 4*: *Belief rule:* $\frac{A\mid\equiv(X,Y)}{A\mid\equiv X}$
- *Rule 5*: *Freshness rule:* $\frac{A\mid\equiv\#(X)}{A\mid\equiv\#(X,Y)}$

- Proof

We define the following goals:

- Goal (1): $U_k \mid\equiv U_k \overset{SK_{iK}}{\rightleftharpoons} GWN_j$;
- Goal (2): $U_k \mid\equiv GWN_j \mid\equiv U_k \overset{SK_{iK}}{\rightleftharpoons} SN_i$;
- Goal (3): $GWN_j \mid\equiv U_k \overset{SK_{iK}}{\rightleftharpoons} GWN_j$;
- Goal (4): $GWN_j \mid\equiv U_k \mid\equiv U_k \overset{SK_{iK}}{\rightleftharpoons} SN_i$;

- Goal (5): $SN_i \mid\equiv SN_i \xleftrightarrow{SK_{iK}} GWN_j$;
- Goal (6): $SN_i \mid\equiv GWN_j \mid\equiv SN_i \xleftrightarrow{SK_{iK}} GWN_j$;
- Goal (7): $GWN_j \mid\equiv SN_i \xleftrightarrow{SK_{iK}} GWN_j$;
- Goal (8): $GWN_j \mid\equiv SN_i \mid\equiv SN_i \xleftrightarrow{SK_{iK}} GWN_j$;

we will define the following assumptions:

- A(1): $GWN_j \mid\equiv \#(N_1)$;
- A(2): $GWN_j \mid\equiv \#(N_2)$;
- A(3): $U_k \mid\equiv \#(N_2)$;
- A(4): $SN_i \mid\equiv \#(h(PIDS_i^{'} \parallel PIDU_k^{'} \parallel PID_{GWN_j} \parallel N_1^{'}))$;
- A(5): $U_k \mid\equiv GWN_j \mid\Rightarrow (U_k \xleftrightarrow{SK_{ik}} GWN_j)$;
- A(6): $GWN_j \mid\equiv U_k \mid\Rightarrow (U_k \xleftrightarrow{SK_{ik}} GWN_j)$;
- A(7): $SN_i \mid\equiv GWN_j \mid\Rightarrow (SN_i \xleftrightarrow{SK_{iK}} GWN_j)$;
- A(8) $GWN_j \mid\equiv SN_i \mid\Rightarrow (SN_i \xleftrightarrow{SK_{iK}} GWN_j)$;
- A(9): $U_k \mid\equiv (U_k \overset{N_1}{\rightleftharpoons} GWN_j)$;
- A(10): $GWN_j \mid\equiv (U_k \overset{\rho_k}{\rightleftharpoons} GWN_j)$;
- A(11): $SN_i \mid\equiv (SN_i \overset{\rho_i}{\rightleftharpoons} GWN_j)$;
- A(12): $GWN_j \mid\equiv (SN_i \overset{\rho_i}{\rightleftharpoons} GWN_j)$.

-Proof postulates

$Step1$: D_1 can be acquired from $Messg_1$:
D_1: $GWN_j \mid\triangleleft \{PIDS_i, N_1, TS_1\}_{\rho_k}$.

$Step2$: D_2 can be derived by applying Rule1 using D_1 and A(10): D_2 : $GWN_j \mid\equiv U_k \mid\sim$ ($PIDS_i$, N_1, TS_1).

$Step3$: D_3 is induced by applying the Rule 5 using A(1):

$$D_3\text{: } GWN_j \mid\equiv \#(PIDS_i, N_1, TS_1)$$

$Step4$: D_4 is induced by applying the Rule2 using D_2 and D_3: D_4: $GWN_j \mid\equiv U_k \mid\equiv (PIDS_i, N_1, TS_1)$.

$Step5$: D_5 is in induced from rule 4 using D_4:
D_5: $GWN_j \mid\equiv U_k \mid\equiv (N_1)$.

$Step6$: D_6 is obtained from $Messg_2$:
D_6 : $SN_i \mid \triangleleft\{N_1, h(PIDS_i \parallel PIDU_k \parallel PID_{GWN_j} \parallel N_1), TS_2\}_{\rho_i}$.

$Step7$: D_7 is deduced from Rule 1 using D_6 and A(11): D_7 : $SN_i \mid\equiv GWN_j \mid\sim (N_1, h(PIDS_i \parallel PIDU_k \parallel PID_{GWN_j} \parallel N_1), TS_2)$.

$Step8$: D_8 is obtained by applying the Rule 5 using A(4): D_8 : $SN_i \mid\equiv \#(N_1, h(PIDS_i \parallel PIDU_k \parallel PID_{GWN_j} \parallel N_1), TS_2)$.

$Step9$: D_9 can be acquired by applying Rule 2 using D_7 and D_8: D_9: $SN_i \mid\equiv GWN_j \mid\equiv (N_1, h(PIDS_i \parallel PIDU_k \parallel PID_{GWN_j} \parallel N_1), TS_2)$.

$Step10$: D_{10} is deduced by applying Rule 4 using D_9, we obtain: D_{10}: $SN_i \mid\equiv GWN_j \mid\equiv (N_1, h(PIDS_i \parallel PIDU_k \parallel PID_{GWN_j} \parallel N_1))$.

$Step11$: D_{11} is obtained from $Messg_3$:
D_{11}: $GWN_j \mid\triangleleft \{N_2, TS_3\}_{\rho_i}$

$Step12$: D_{12} is deduced by applying Rule 1 using D_{11} and A(12): D_{12} : $GWN_j \mid\equiv SN_i \mid\sim$ (N_2, TS_3).

$Step13$: D_{13} is obtained by applying Rule 5 and Rule 2 using A(2) and D_{12} respectively: D_{13}: $GWN_j \mid\equiv SN_i \mid\equiv$ (N_2, TS_3).

$Step14$: D_{14} is deduced by applying Rule 4 using D_{13}: D_{14}: $GWN_j \mid\equiv SN_i \mid\equiv (N_2)$.

$Step15$: D_{15} and D_{16} are deduced from D_{10} and D_{14},: SN_i and GWN_j can compute the session key SK_{ik}= $h'($ $h(PIDS_i^{'} \parallel PIDU_k^{'} \parallel PID_{gwn_j} \parallel N_1^{'}) \parallel\gamma)$ such as $\gamma = N_1^{"} \oplus N_2$:

$$D_{15}\text{: } GWN_j \mid\equiv SN_i \mid\equiv SN_i \xleftrightarrow{SK_{ik}} GWN_j \text{ (Goal 8).}$$

$$D_{16}\text{: } SN_i \mid\equiv GWN_j \mid\equiv SN_i \xleftrightarrow{SK_{ik}} GWN_j \text{ (Goal 6).}$$

$Step16$: D_{17} and D_{18} are obtained applying Rule3 using D_{15} and A(8), and D_{16} and A(7) respectively.

$$D_{17} : GWN_j \mid\equiv (SN_i \xleftrightarrow{SK_{ik}} GWN_j) \text{ (Goal 5).}$$

$$D_{18} : SN_i \mid\equiv (SN_i \xleftrightarrow{SK_{ik}} GWN_j) \text{ (Goal 7).}$$

$Step17$: D_{19} is obtained from $Messg_4$: D_{19}: $U_k \mid\triangleleft \{N_2, h(PIDS_i \parallel PIDU_k \parallel PID_{GWN_j} \parallel N_1), ST_4\}_{N_1}$

$Step18$: D_{20} is deduced from Rule 1 using D_{19} and A(9). D_{20}: $U_k \mid\equiv GWN_j \mid\sim$ (N_2, $h($ $PIDS_i \parallel PIDU_k \parallel PID_{GWN_j} \parallel N_1)$, ST_4).

$Step19$: D_{21} is obtained by applying Rule 5 using A(3): D_{21}: $U_k \mid\equiv \#(N_2, h(PIDS_i \parallel PIDU_k \parallel PID_{GWN_j} \parallel N_1)$, $ST_4)$.

$Step20$: D_{22} is deduced by applying Rule 2 using D_{19} and D_{20}: D_{22}: $U_k \mid\equiv GWN_j \mid\equiv (N_2, h(PIDS_i \parallel PIDU_k \parallel PID_{GWN_j} \parallel N_1)$, $ST_4)$.

$Step21$: D_{23} is deduced applying Rule 4 using D_{22}: D_{23}: $U_k \mid\equiv GWN_j \mid\equiv (N_2, h(PIDS_i \parallel PIDU_k \parallel PID_{GWN_j} \parallel N_1))$.

$Step22$: D_{24} and D_{25} are deduced by applying D_5 and D_{23}. U_k and GWN_j can compute the Session Key SK_{ik} = $h(h($ $PIDS_i^{'} \parallel PIDU_k^{'} \parallel PID_{gwn_j} \parallel N_1^{'}) \parallel \gamma)$ such as $\gamma = N_1^{"} \oplus N_2$.

$$D_{24}\text{: } U_k \mid\equiv GWN_j \mid\equiv U_k \xleftrightarrow{SK_{ik}} GWN_j. \text{ (Goal 2).}$$

$$D_{25}\text{: } GWN_j \mid\equiv U_k \mid\equiv U_k \xleftrightarrow{SK_{ik}} GWN_j. \text{ (Goal 4)}$$

$Step23$: D_{26} and D_{27} are induced by applying Rule3 using D_{24} and A(5), and D_{25} and A(6) respectively.

$$D_{26}\text{: } U_k \mid\equiv U_k \xleftrightarrow{SK_{ik}} GWN_j. \text{ (Goal 1).}$$

$$D_{27}\text{: } GWN_j \mid\equiv U_k \xleftrightarrow{SK_{ik}} GWN_j. \text{ (Goal 3).}$$

C. Informal Security Analysis

This section provides an informal analysis of the performance and effectiveness of UAWSNA-IoT against some of the most know attacks.

1) *Privileged Insider Attack*: In the scenario where a privileged insider adversary “A”, intercepts the registration message IDU_k from a legitimate user U_k , “A” endeavours to calculate U_k’s session key using the messages specified in Login/Authentication phase. Nevertheless, “A“ is unable to calculate U_k’s session key SK_{ik}. In order to calculate SK_{ik}= h’$(h(PIDS_i^{'} \parallel PIDU_k^{'} \parallel PID_{GWN_j}) \parallel N_1^{'}) \parallel \gamma$), such as $\gamma = N_1 \oplus N_2$, “A” needs to compute $h^*(PIDS_i^{'} \parallel PIDU_k^{'} \parallel PID_{GWN_j} \parallel N_1^{'})$ and γ which are considered as main parameters. Additionally, the parameters vu_k and ρ_k are shared secrets between U_k and GWN_j, which are essential for calculating N_1 and $PIDS_i^{'}$, respectively. Yet, A is unable to calculate $h^*(PIDS_i^{'} \parallel PIDU_k^{'} \parallel PID_{GWN_j} \parallel N_1^{'})$ and γ from the login request message $\{V_1, V_2, V_3, PIDU_k, TS_1\}$ without knowledge the shared secrets $\{vu_k, \rho_k\}$ and random numbers $\{N_1, N_2\}$. As a result, UAWSNA-IoT guarantees protection against privileged insider attacks.

2) *Stolen smart card attack:* If adversary ”A” steals a legitimate user’s smart card (SC_k) using a power analysis attack [40] to extract its stored data. This data comprises $PIDU_k$, vu_k^s, ρ_k^s, $PIDS_i$, and MPW obtained by the following operations: $vu_k^s = h(IDU_k \parallel PW_k\ ) \oplus vu_k$; $\rho_k^s = h(IDU_k \parallel PW_k) \oplus \rho_k$ and $MPW = h\ (IDU_k \parallel PW_k \parallel \rho_k\ )$. While the adversary ”A”may be able to make guesses about the user’s password PW_k, they lack the means to confirm its accuracy without having knowledge of the user’s identity IDU_k. As a result, UAWSNA-IoT effectively defends against stolen smart card attacks.

3) *Offline password guessing Attack*: If Adversary “A”succeeded in extracting the information (ρ_k^s, vu_k^s, $PIDU_k$, $PIDS_i$, and MPW) stored in the smart card (SC_k) memory via a power analysis attack [40]. Then “A” tries to impersonate U_k and tries to guess IDU_k and PW_k by extracting them from the knowing values ρ_k^s , vu_k^s and MPW that as: $vu_k^s = h(IDU_k \parallel PW_k) \oplus vu_k$; $\rho_k^s = h(IDU_k \parallel PW_k) \oplus \rho_k$ and $MPW = h(IDU_k \| PW_k \| \rho_k)$. That is, it is difficult to guess IDU_k and PW_k only based on the information’s stocked in SC_k. As they are merged and hashed alongside other values within parameters such as $PIDU_k$, ρ_k^s, and MPW. Therefore, UAWSNA-IoT is secure against offline password guessing Attack. Therefore, UAWSNA-IoT is secure against offline password guessing Attack.

4) *Stolen Verifier Attack:* Suppose an adversary “A”, illicitly acquires the database DB_{gwn_j} of GWN_j , which includes $\alpha_{G_j}\sigma_{G_j}$, ρ_k, $PIDU_k$, vu_k, ρ_i, $PIDS_i$, PID_{GWN_j}. Nevertheless, ”A” is unable to calculate the session key for the legitimate user U_k using these parameters. To compute the session key $SK_{ik} = h(h^*(PIDS_i^{'} \parallel PIDU_k^{'} \parallel PID_{gwn_j}) \parallel N_1^{'}\ ) \| \gamma)$ such as $\gamma = N_1 \oplus N_2$, ”A” must be aware of the both random numbers N_1 and N_2 generated in each session. As “A” lacks information about the values of N_1 and N_2, it is unable to compute the correct SK_{ik}. As a result, UAWSNA-IoT has the resistance against the stolen verifier attacks.

5) *Mutual Authentication:* To establish mutual authentication in UAWSNA-IoT, each participant executes verification processes to confirm the legitimacy of connected entities. The gateway GWN_j examines the correctness of ($V_3 = V_3^{'}$) and ($V_8 = V_8^{'}$). Simultaneously, the sensor node SN_i validates whether($V_6 = V_6^{'}$), and user (U_k) verifies that ($V_8 = V_8^{'}$). If the verification process is successful in its entirety, it can be inferred that each participant has been mutually authenticated. Consequently, UAWSNA-IoT ensures mutual authentication.

6) *Replay Attacks:* In the authentication phase, the legitimate participants U_k and SN_i, generate the random numbers N_1 and N_2 respectively. These random numbers are utilized to calculate the values V_3, V_6, and V_8, which are included in exchanged messages. These values play a crucial role in verifying the freshness of N_1 and N_2. Additionally, participants using the timestamps TS_1, TS_2, TS_3, and TS_4 in the messages to assess there freshness. Therefore, GWN_j, SN_i and U_k can distinguish the replayed message from the received messages. As a result, UAWSNA-IoT provides security against replay attacks.

7) *Anonymity and unlinkability:* Anonymity is guaranteed in UAWSNA-IoT by transmitting the user’s identity IDU_k in embedded form such as the shared private key $\rho_k = h(IDU_k \parallel \sigma_{G_j})$ and pseudo-identity $PIDU_k = h(IDU_k \parallel \alpha_{G_j})$. The adversary ”A” faces insurmountable challenges in determining the user’s identity without possessing knowledge of the master private key σ_{G_j}or mask key α_{G_j}, possesses a size of 160 bits. In the authentication phase, unique random nonce’s N_1, N_2 and the current timestamp TS_1, TS_2, TS_3 , and TS_4 are deliberately chosen for different sessions. This selection ensures that the messages V_1, ..., V_9 sent by the participants in each session are distinct from one another. The adversary “A” lacks the ability to identify any correlation among the messages exchanged by U_k, GWN_j, and SN_i. Additionally, “A” is unable to trace the sender of these messages. Therefore, UAWSNA-IoT ensures the preservation of both anonymity and unlinkability.

8) *Known session key attack*: If the Adversary ”A” reveals a session key SK_{ik}, it is important to note that $SKij$ is generated from the hash value of a pseudo-identity $PIDU_k$, $PIDS_i$, and PID_{GWN_j} linked to authorized participants and random numbers N_1, and N_2. Due to the properties inherent in a collision-resistant secure one-way hash function, the adversary ”A” is incapable of extracting the random numbers from SK_{ik}. Furthermore, for any other sessions, ”A” lacks the capability to

compute the correct session key unless in possession of the current random numbers. Consequently, UAWSNA-IoT is secure against known session key attacks.

9) *Perfect Forward Secrecy*: In UAWSNA-IoT, the session key is computed using the expression $SK_{ik} = h($ $h^{*}(PIDS_i^{'} \parallel PIDU_k^{'} \parallel PID_{GWN_j} \parallel N_1^{'})\parallel \gamma^{''})$ with $\gamma^{''} = N_1 \oplus N_2^{''}$. The values N_1 and N_2 represent random numbers generated by U_k and SN_i, respectively. Despite that the adversary “A” know all the keys (ρ_k, ρ_i), including the password PW_k and the shared secret parameters PID_{GWN_j}, $PIDS_i$, and $PIDU_k$. ”A” cannot determine the current or past session keys without knowing the values of the random numbers N_1, N_2. As a result, it is evident that UAWSNA-IoT effectively guarantees perfect forward secrecy.
10) *Sensor node capture Attack*: We assume a scenario where an adversary “A”, takes control of a particular sensor node SN_i and obtains shared key (ρ_i) from SN_i’s memory by employing a power analysis attack[40]. Subsequently, “A” has the capability to authenticate with gateway GWN_j and user U_k. Nevertheless, ”A” does not pose a threat to other sensor nodes. As the shared secret key (ρ_i) is determined by the formula $\rho_i = h(IDS_i \parallel \sigma_{G_j})$, “A” is limited to authenticating solely with the particular sensor node SN_i. “A” is incapable of computing any information pertaining to other sensor nodes. Thus, UAWSNA-IoT effectively withstands sensor node capture attacks.
11) *Man-in-middle attack*: During the login/authentication phase, GWN_j verifies the authenticity of U_k by validating the shared secret key (ρ_k) and the associated value vu_k in its DB_{GWN_j}. Similarly, SN_i can authenticate GWN_j by leveraging its knowledge of SN_i’s secret key (ρ_i). Furthermore, GWN_j can identify SN_i through his $h^{*}(PIDS_i^{'} \parallel PIDU_k^{'} \parallel PID_{GWN_j}) \parallel N_1^{'})$ knowledge. Finally, U_k authenticates GWN_j through his N_1 knowledge. As a result, all participants are able to mutually authenticate each other. This robust authentication mechanism makes UAWSNA-IoT resistant to man-in-the-middle attacks.

TABLE III
SECURITY EFFICIENCY COMPARISON.

ATTACK	[27]	[14]	[20]	[25]	[6]	[35]	[38]	Our
A_1	○	○	×	×	×	○	○	○
A_2	○	○	×	×	×	○	○	○
A_3	○	○	×	○	×	○	○	○
A_4	×	○	×	○	○	○	○	○
A_5	○	○	○	○	○	○	○	○
A_6	○	○	○	○	×	○	×	○
A_7	○	○	×	×	×	○	○	○
A_8	○	○	×	○	×	○	○	○
A_9	○	○	×	○	○	○	○	○
A_{10}	×	○	×	×	×	○	○	○
A_{11}	×	×	×	○	×	×	○	○
A_{12}	○	○	○	○	○	○	×	○

A_1: Privileged Insider Attack; A_2: Stolen smart card attack; A_3: Offline password guessing Attack; A_4: Stolen Verifier Attack; A_5: Mutual Authentication; A_6: Replay Attacks; A_7: Anonymity; A_8: unlinkability; A_9:Known session key attack; A_{10}: Perfect Forward Secrecy; A_{11}: Sensor node capture Attack; A_{12}: Man-in-middle attack ○ The protocol s secure ×: The protocol is not secure.

VI. PERFORMANCE ANALYSIS

In this section, we evaluate the proposed scheme by contrasting it with various related schemes ([27], [14], [20], [25], [6], [35], and [38]) concerning security performances and computational, communication, and storage costs. The corresponding results are presented in Tables III, IV, and V respectively.

A. Security Performance Evaluation

We compare the security performance of UAWSNA-IoT with related protocols ([27], [14], [20], [25], [6], [35], and [38]) against various well-known attacks. As shown in Table III, UAWSNA-IoT provides superior security performance when compared to related protocols.

B. Computational Cost

We assess the computational costs of UAWSNA-IoT and compare its performance to the related protocols [27], [14], [20], [25], [6], [35], and [38] as is presented in Table IV. In our study, we symbolize T_h, T_{Eccm} and $T_{E/D}$, which respectively represent the time to execute the hash function 0.068 millisecond (ms), ECC points multiplication 2.501 ms, and symmetric encryption/decryption 0.56 ms respectively [30]. In our study, we did not take into account the computational cost of the XOR operation as it is considered negligible. According to our study, the computational cost of UAWSNA- IoT is lower compared to the related protocols ([27], [14], [25], [6], [35], and [38]) at both the sensor node and network levels. However, it does a higher computational cost than the protocol ([20]). Unfortunately, the protocol [20] is susceptible to several attacks, as shown in Table III.

C. Communication Cost

In this section, we will evaluate the communication cost of our protocol UAWSNA-IoT in comparison to protocols [27], [14], [20], [25], [6], [35], and [38]. We will assume that the output size of hash function *h(.)*, random number, and timestamp (TS_i such as i∈[1,4]) are 160, 160, and 32 bits, respectively. In order to compute the communication costs of UAWSNA-IoT, we consider the values $(V_i, i = 1..9)$ that go into the messages calculation used in the authentication phase, which have a size of 160 bits. We also calculate the size of the messages sent by each entity (U_k, GWN_j, and SN_i) separately in authentication phase. The size of the messages exchanged are: $Messg_1\{V_1, V_2, V_3, PIDU_k, TS_1\}$ requires (4 × 160 + 32 = 672 bits), $Messg_2$ $\{V_4, V_5, V_6, PIDU_k^{'},$ $TS_2\}$ requires (4 × 160 + 32 = 672 bits), $Messg_3\{V_7, V_8,$ $TS_3\}$, and $Messg_4\{V_5, V_8, V_9, TS_4\}$, require (2 × 160 + 32 = 352 bits) and (3 × 160 + 32 = 512 bits) respectively. Thus, the total communication cost for the three entities U_k, GWN_j, and SN_i are 672 + 672 + 352 + 512 = 2 208 bits. The Table V shows the communication costs comparison of UAWSNA-IoT and related schemes ([27], [14], [20], [25], [6], [35], and [38]).

TABLE IV
COMPUTATIONAL COST COMPARISON.

Schemes	U_k (ms)	GWN_j (ms)	SN_i (ms)	Total Time (ms)
[27]	$7T_h$+$2T_{E/D}$=1,596	$4T_h$=0,272	$10T_h$+$2T_{E/D}$=1.8	$21T_h$+$4T_{E/D}$=3,668
[14]	$6T_h$+$3T_{Eccm}$=7,503	$7T_h$+T_{Eccm}=2.977	$4T_h$+$2T_{eccm}$=5.274	$17T_h$+$6T_{Eccm}$=15,754
[20]	$3T_h$=0,204	$3T_h$=0,204	$2T_h$=0,136	$8T_h$=0.544
[25]	$7T_h$+$2T_{E/D}$=1,12	$12T_h$+$2T_{E/D}$=1.936	$6T_h$=0.408	$25T_h$+$4T_{E/D}$=3.94
[6]	$3T_h$+$T_{E/D}$+$2T_{Eccm}$=5.766	$12T_h$+$2T_{Eccm}$=5.818	T_h+$2T_{E/D}$=1.188	$16T_h$+$3T_{E/D}$+$4T_{Eccm}$=12.024
[35]	$8T_h$+$T_{E/D}$+$3T_{Eccm}$=7,503	$7T_h$+$2T_{E/D}$+T_{Eccm}=4.097	$5T_h$+$T_{E/D}$+$2T_{Eccm}$=5.902	$20T_h$+$6T_{Eccm}$+$4T_{E/D}$=18.606
[38]	$8T_h$=0,544	$13T_h$=0.884	$6T_h$=0.408	$27T_h$=1,836
UAWSNA-IoT	$8T_h$=0,544	$8T_h$=0,544	$4T_h$=0.272	$20T_h$=1,36

It can be concluded that UAWSNA-IoT is more efficient in terms of total communication costs than the related schemes ([27], [20], [25], [6], [35], and [38]). However, UAWSNA-IoT has a higher total communication cost than the protocol ([14]). Unfortunately, the scheme ([14]) is insecure due to its vulnerability to sensor node capture attack. Moreover, at the sensor node level, UAWSNA-IoT boasts a reduced message size compared to protocols ([14], [20], [25], [6], [35], and [38]), while higher then protocol [27]. But the protocol [27], which is not secure against some attacks as presented in Table III.

TABLE V
COMMUNICATION COST COMPARISON.

Schemes	U_k	GWN_j	SN_i	Total cost	$Num_{Messages}$
[27]	512	544	192	2912	4
[14]	512	512	384	1408	3
[20]	640	1440	480	2560	4
[25]	800	1440	480	2720	4
[6]	960	1440	640	3040	3
[35]	736	864	704	2304	4
[38]	672	2784	480	3936	7
Our	672	1184	352	2 208	4

D. Storage Cost

In this part, we focuses on the storage space cost study in the sensor node level. To simplify the analysis, the storage space required for hash functions is excluded. Table VI outlines the storage space cost at the sensor node level in the UAWSNA-IoT scheme, compared to protocols ([27], [14], [20], [25], [6], [35], and [38]). In the UAWSNA-IoT protocol, the authentication parameters stored in sensor node is solely ρ_i, necessitating only 160 bits of storage space. Table VI outlines the storage space cost at the sensor node level in the UAWSNA-IoT scheme, compared to protocols ([27], [14], [20], [25], [6], [35], and [38]). Furthermore, the storage cost at sensor node in UAWSNA-IoT is the same as one in [6], and it is less than the schemes [27, 14, 20, 25, 35, and 38]. Unfortunately the scheme [6] is not secure against some attacks as shown in Table III.

TABLE VI
STORAGE COST COMPARISON

Schemes	Sensor Node
[27]	320 bits
[14]	420 bits
[20]	420 bits
[25]	420 bits
[6]	160 bits
[35]	320 bits
[38]	420 bits
UAWSNA-IoT	160 bits

VII. CONCLUSION

In this paper, we presented a new user authentication protocol to secure the wireless sensor networks access using two factors called " UAWSNA-IoT". UAWSNA-IoT also mitigates network congestion by decreasing the size of authentication messages during the authentication phase. Furthermore, UAWSNA-IoT allows users access to data in WSN after their authentication process conducted by gateway via internet. UAWSNA-IoT is designed to be adaptable, allowing easy addition of sensors nodes as needed, ensuring scalability to meet growing services demands. Our protocol provides robust security measures against several attacks. It ensures anonymity, offers complete mutual authentication among all authentication entities, and maintains perfect forward secrecy during the authentication phase. The efficiency, lightweight design, and impressive performance of UAWSNA-IoT, as demonstrated in Section VI, making it an ideal choice for resource-constrained IoT devices like WSN.

REFERENCES

[1] H. Khemissa and D. Tandjaoui, "A novel lightweight authentication scheme for heterogeneous wireless sensor networks in the context of Internet of Things", 2016 Wireless Telecommunications Symposium (WTS), 2016, pp. 1-6.

[2] R. Amin and G.P. Biswas, "A Secure Light Weight Scheme for User Authentication and Key Agreement in Multi-gateway Based Wireless Sensor Networks", Ad Hoc Networks, vol. 36, no. 1, pp. 58-80, January 2016.

[3] M. L. Das, "Two-factor user authentication in wireless sensor networks," in IEEE Transactions on Wireless Communications, vol. 8, no. 3, pp. 1086-1090, March 2009, doi: 10.1109/TWC.2008.080128.

[4] M. Benfilali and A. Gafour, "A survey of wireless sensor network security in the context of Internet of Things," 2017 4th International Conference on Information and Communication Technologies for Disaster Management (ICT-DM), Münster, Germany, 2017, pp. 1-8, doi: 10.1109/ICT-DM.2017.8275691.

[5] D. Ashok Kumar. (2015). "A secure and effective biometric-based user authentication scheme for wireless sensor networks using smart card and fuzzy extractor". International Journal of Communication Systems. 30. 10.1002/dac.2933.

[6] T.M. Butt, R.Riaz, C. Chakraborty, S.S. Rizvi, A. Paul, "Cogent and energy efficient authentication protocol for wsn in IoT", Comput. Mater. Contin. 2021, 68, 1877–1898.

[7] D. Zhang, Y. Qian, J. Wan, S. Zhao, "An efficient rfid search protocol based on clouds", Mobile Netw. Appl. 20 (3) (2015) 356—362.

[8] M. Abdalla, P. Fouque, D. Pointcheval, "Password-based authenticated key exchange in the three-party setting", In Proceedings of the 8th International Workshop on Theory and Practice in Public Key Cryptography (PKC'05), Lecture Notes in Computer Science (LNCS), Les Diablerets, Switzerland, 23–26 January 2005; pp. 65–84.

[9] C. Li, C. Lee, Chi-Yao Weng, and Chun-I Fan. 2013. "An Extended Multi-Server-Based User Authentication and Key Agreement Scheme with User Anonymity." KSII Transactions on Internet Information Systems 7 (1): 119–31.

[10] L. Lamport,"Password authentication with insecure communication", Commun. ACM 24, 11 (Nov. 1981), 770–772. https://doi.org/0.1145/358790.358797.

[11] G. Sun, V. Chang, M. Ramachandran, Z. Sun, G. Li, Y. Hongfang, D. Liao, (2016)," Efficient Location Privacy Algorithm for Internet of Things (IoT) Services and Applications", Journal of Network and Computer Applications. 89. 10.1016/j.jnca.2016.10.011.

[12] K.H. Wong, Y. Zheng, J. Cao, S. Wang, "A dynamic user authentication scheme for wireless sensor networks", in: IEEE International Conference on Sensor Net- works, Ubiquitous, and Trustworthy Computing (SUTC'06), June, vol1, IEEE, 2006,p. 8.

[13] L. Fagen, H. Yanan, J. Chunhua, "Practical access control for sensor networks in the context of the Internet of Things, Computer Communications", Volumes 89–90, 2016, Pages 154-164, ISSN 0140-3664.

[14] Q. Xie, Z. Ding, B. Hu, "A secure and privacy-preserving three-factor anonymous authentication scheme for wireless sensor networks in Internet of Things", Secur. Commun. Netw. 2021, 2021, 4799223.

[15] R. Amin, S.H. Islam, G. Biswas, M.S. Obaidat,"A robust mutual authentication protocol for WSN with multiple base-stations", AdHoc Networks.75, 1–18 (2018).

[16] Q. Jiang, J. Ma, ;X. Lu, Y. Tian,"An efficient two-factor user authentication scheme with unlinkability for wireless sensor networks", Peer-to-Peer Netw. Appl. 8(6), 1070–1081 (2015).

[17] F. Wu, Xu, L., Kumari, S. et al. "A new and secure authentication scheme for wireless sensor networks with formal proof", Peer-to-Peer Netw. Appl. 10, 16–30 (2017). https://doi.org /10.1007/s120 83-015-0404-5.

[18] F. Wu, L. Xu, S. Kumari, X. Li, "A novel and provably secure biometrics-based three-factor remote authentication scheme for mobile client–server networks", Comput. Electr. Eng. 45, 274–285 (2015).

[19] K. Xue, Ma, C. Hong, P. R. Ding, "A temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks", J. Netw. Comput. Appl. 36(1), 316–323 (2013).

[20] M. Masud, G.S. Gaba, K. Choudhary, M.S. Hossain, M.F. Alhamid, G. Muhammad, "Lightweight and anonymity-preserving user authentication scheme for IoT-based healthcare", IEEE Internet Things J. 2021, 9, 2649–2656.

[21] K. Sharmila, P. Bhushan, "Secure Key Management and Mutual Authentication Protocol for Wireless Sensor Network by Linking Edge Devices using Hybrid Approach", Wireless PersCommun (2023).

[22] B. Hu, W. Tang, Q. Xie, "A two-factor security authentication scheme for wireless sensor networks in IoT environments",Neurocomputing, Volume 500, 2022, Pages 741-749, ISSN 0925-2312.

[23] L. Kou, Y. Shi, L. Zhang, D. Liu, Q. Yang, "A lightweight three-factor user authentication protocol for the information perception of IoT", CMC-Comput. Mater. Contin. 2019, 58, 545–565.

[24] F. Wu , L. Xu, S. Kumari, X. Li, J. Shen, K.K.R Choo, M. Wazid, A.K. Das, "An efficient authentication and key agreement scheme for multi-gateway wireless sensor networks in IoT deployment", J. Netw. Comput. Appl. 89, 72–85 (2017). 24.

[25] L. Kou, Y. Shi, L. Zhang, D. Liu, Q. Yang, "A lightweight three-factor user authentication protocol for the information perception of IoT", CMC-Comput. Mater. Contin. 2019, 58, 545–565.

[26] C. Wang, D. Wang, Y. Tu, G. Xu, "Wang, H. Understanding node capture attacks in user authentication schemes for wireless sensor networks", IEEE Trans. Dependable Secur. Comput. 2020, 19, 507–523.

[27] M. Shuai, N. Yu, H. Wang, L. Xiong, Y. Li, "A lightweight three-factor Anonymous authentication scheme with privacy protection for personalized healthcare applications", J. Organ. End User Comput. JOEUC 2021, 33, 1–18.

[28] D. Wang, H. Cheng, P. Wang, X. Huang, G. Jian, "Zipf's law in passwords", IEEE Trans. Inf. Forensics Secur. 2017, 12, 2776–2791.

[29] M. Wazid, P. Bagga, A.K. Das, S. Shetty, J.J. Rodrigues, Y. Park, "AKM-IoV: Authenticated key management protocol in fog computing-based internet of vehicles deployment", IEEE Internet Things J. 2019, 6, 8804–8817.

[30] Z. Ding, Q. Xie, "Provably Secure Dynamic Anonymous Authentication Protocol for Wireless Sensor Networks in Internet of Things", Sustainability 2023, 15, 5734.

[31] M. Turkanovic, M. Holbl, "An improved dynamic password-based user authentication scheme for hierarchical wireless sensor networks", Elektronikair Elektrotechnika 19 (6) (2013) 109–117.

[32] S. Szymoniak and S. Kesar, "Key Agreement and Authentication Protocols in the Internet of Things: A Survey," Applied Sciences, vol. 13, no. 1, p. 404, Dec. 2023, doi: 10.3390/app13010404.

[33] B. Turkanović, M. Brumen, et al, "A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion", Ad Hoc Netw. 20 (2014) 96–112.

[34] V. Boyko, P. MacKenzie, S. Patel, "Provably secure password-authenticated key exchange using Diffie-Hellman", In Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Bruges, Belgium, 14–18 May 2000; pp. 156–171.M.

[35] Q. Xie, K. Li, X. Tan, L. Han, W.Tang, B.Hu, "A secure and privacy-preserving authentication protocol for wireless sensor networks in smart city", EURASIP J. Wirel. Commun. Netw. 2021, 2021, 119.

[36] L. Xu, F. Wu, "Cryptanalysis and improvement of a user authentication scheme preserving uniqueness and anonymity for connected health care, J. Med. Syst. 39 (2) (2015) 10.

[37] J. Wang, C. Ju, H.-j. Kim, R. S. Sherratt, and S. Lee, "A mobile assisted coverage hole patching scheme based on particle swarm optimization for WSNs," Cluster Comput., vol. 22, pp. 1787–1795, 2019.

[38] JH. Yang, "A multi-gateway authentication and key-agreement scheme on wireless sensor networks for IoT". EURASIP J. on Info. Security 2023, 2 (2023). https://doi.org/10.1186/s13635-023-00138-z.

[39] M. Burrows, M. Abadi, R.M. A Needham,: "logic of authentication". ACM Trans. Comput. Syst. 1990, 8, 18–36.

[40] T. S. Messerges, E. A. Dabbish and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks", in IEEE Transactions on, computers, vol. 51, no. 5, pp. 541-552, May 2002, doi: 10.1109/TC.2002.1004593.

[41] F. Wu, L. Xu, S. Kumari, "A new and secure authentication scheme for wireless sensor networks with formal proof". Peer-to-Peer Netw. Appl. 10, 16–30 (2017). https://doi.org /10.1007/s120 83-015-0404-5.

[42] S. Banerjee, C. Chunka, S. Sen, "An Enhanced and Secure Biometric Based User Authentication Scheme in Wireless Sensor Networks Using Smart Cards", Wireless Pers Commun 107, 243–270 (2019).

Benfilali Mostefa obtained the Engineering degree in Computer Sciences from Mohamed Boudiaf University (USTO), Oran, Algeria, in 1998, followed by a Master's degree from Tahri Mohamed University in 2008. He is a member of the (Evolutionary Engineering and Distributed Information Systems Laboratory) and Network and Communication research team at the U.D.L. His research interests include Wireless sensor network, Internet of things security, and cryptography.

Gafour Abdelkader is Full Professor in Computer Science Department of Djillali Liabes University (U.D.L) of Sidi Bel Abbes, Algeria. He is a member of the (Evolutionary Engineering and Distributed Information Systems Laboratory) and Network and Communication research team at the U.D.L. His research interests are in networking, including wireless sensor networks, Data-mining, Internet of Things (IoT), Networks security.

Belghachi Mohamed is a distinguished Professor Researcher affiliated with Tahri Mohamed University of Bechar in Algeria. He holds a prominent position as a member of the"Ad-hoc Networks" research team within the STIC (Systems and Information and Communication Technologies) Laboratory at Abou Bekr Belkaid Tlemcen University. With a profound knowledge and experience in various domains, his expetise spans across Wireless Sensor Networks (WSN), Internet of Things (IoT), Internet of Vehicles (IoV), Flying Ad-hoc Networks (FANETs), and Artificial Intelligence (AI). His contributions have significantly impacted these fields, further advancing the realms of connectivity, communication, and intelligent systems.