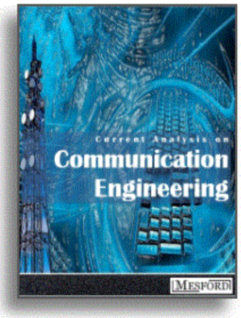
Unlinkable User Authenticated Key Agreement for Multi-Gateway Wireless Sensor Networks



Solomon Kuonga Nkhoma[1], Patrick Ali[1], Levis Eneya[1], and Hyunsung Kim[1,2,*]

[1] *Departmen of Mathematical Sciences, Chancellor College, University of Malawi P.O.Box 280, Zomba, Malawi*
[2] *Department of Cyber Security, Kyungil University Kyungsan, Kyungbuk 38428, Korea*

Background

Wireless sensor network (WSN) has wide potential application in various fields such as military, agricultural and healthcare. WSNs need effective security mechanisms because they are deployed in hostile unattended environments. Various user authentication schemes were proposed for WSNs security. However, there are many previous schemes that have various security vulnerabilities including masquerading, password guessing attack and traceability. This paper proposes an unlinkable user authenticated key agreement scheme (UAKA) for multi-gateway WSNs that could achieve desirable security and privacy attributes. It preserves all the original merits of the related schemes and the security of UAKA is analyzed using the BAN logic. Furthermore, the performance of UAKA is comparable to the related existing schemes.

Publication History: Received: 21 September 2018 | Revised: 26 October 2018 |Accepted: 31 October 2018

Keywords:
Wireless sensor network, Information security, Multi-gateway, Unlinkability, Anonymity, Authentication, Key agreement.

1. INTRODUCTION

Wireless sensor networks (WSNs) consisted of many small devices each with sensing, processing, and communication capabilities to monitor the targeting environment. There are numerous fields of application of WSNs like environmental monitoring, agriculture, military, health care and so on. A WSN is basically consist of numerous sensor nodes with the wireless channel to connect the nodes. Number of nodes in any network varies from hundreds to thousands which makes it different from other wireless networks [1-3]. Since, most of the time, deploying a WSN in a hostile environment is done by a random distribution, it is difficult to know the topology of WSNs a priori. Generally, there are three entities in WSN, sensor, a gateway node (GWN), and user. Sensors collect data from the deployed environment and send it to GWN. GWNs are responsible for conveying the sensed data to users and the request data to sensors. Registered users can access data obtained by the sensors after registering to GWN [4]. Such users need to be authorized and, if done positively, allowed to gather data from or send commands to the sensor node. Since the most important characteristic of WSNs is their resource constrained feature, a lightweight security solution is required. A key challenge is how to establish a shared cryptographic key in a secure manner between the sensor node and the user.

Numerous user authentication and key agreement schemes for enhancing the security of WSNs have been proposed [5-14]. Wong *et al.* proposed a user authentication scheme for WSNs, which requires only the computation of hash functions [5]. However, it was proved to be vulnerable to stolen verifier attack and many logged-in users with the same login identity (ID) attack. Das proposed an efficient password-based user authentication to solve the weaknesses in Wong et al.'s scheme, which uses the temporal credentials for verification [6]. Das's scheme is also shown to be vulnerable to denial-of-service attack and node capture attack. Khan and Alghathbar presented an improvement of Das's scheme [7]. Vaidya *et al.* identified the security pitfalls in Khan and Alghathbar's scheme [8]. To overcome these security pitfalls, Vaidya et al. proposed an improved version of Khan and Alghathbar's scheme. Deebak identified that Vaidya et al.'s scheme is vulnerable to stolen smart card, GWN bypassing and sensor node key impersonation [9]. Independently, Das *et al.* proposed authentication and key agreement schemes for WSNs using smart cards, which supports a user to viably and securely connect to the nodes of a WSN [10]. Turkanovic et al. proposed a user authentication and key agreement model to overcome the security flaws of the earlier designed schemes [11]. Farash *et al.* shown that Turkanovic et al.'s scheme is insecure and inefficient for various security drawbacks such as

[*] Address correspondence to this author at the Department of Cyber Security, Kyungil University Kyungsan, Kyungbuk 38428, Korea;
E-mail: kim@kiu.ac.kr

Mesford Publisher Inc
Office Address: Suite 2205, 350 Webb Drive, Mississauga, ON L5B3W4, Canada; T: +1 (647) 7109849 |E: cace@mesford.ca, contact@mesford.ca, https://mesford.ca/journals/cace/

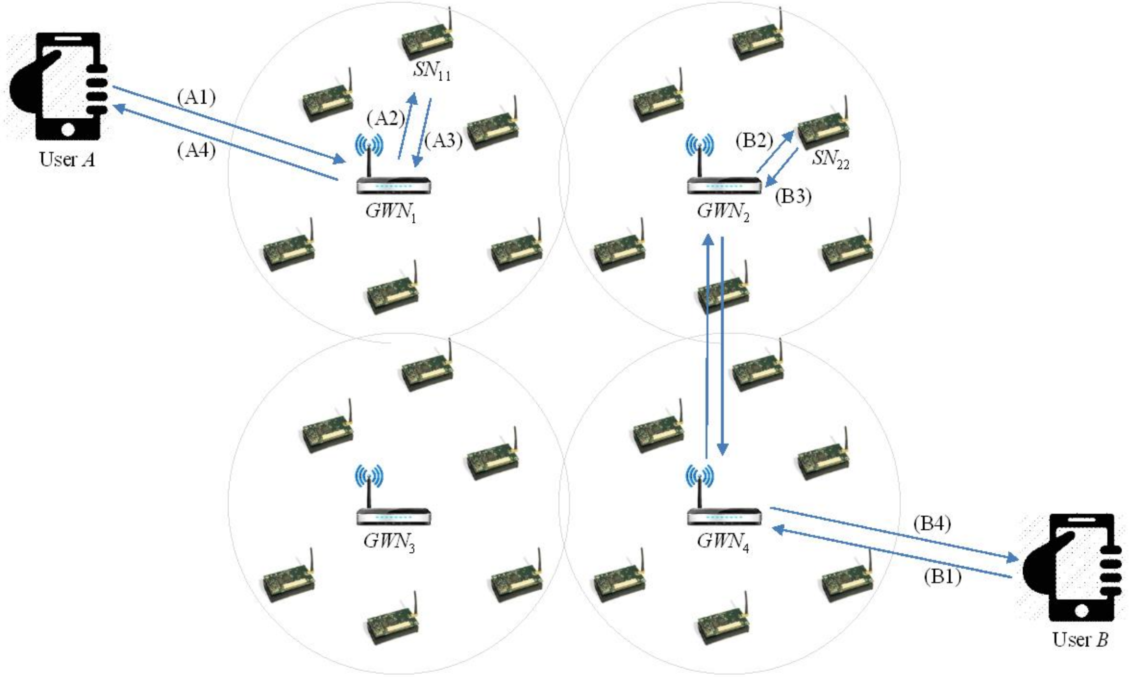


Fig. (1). Multi GWN based WSN model.

a session key agreement, mutual authentication between all parties, traceability, preservation of user anonymity, privileged insider attack and password guessing attack [12]. To overcome these shortcomings, Amin and Biswas presented a secure lightweight scheme for user authentication and key agreement in multi-gateway based WSNs [13]. However, Srinivas et al. showed that Amin and Biswas's scheme has leakages of sensors secret keys and the system key, and is weak against server spoofing attack, user impersonation attack, stolen smart card attack, off-line password guessing attack and identity guessing attack [14].

The purpose of this paper is to design an unlinkable user authenticated key agreement scheme (UAKA) for multi-gateway WSNs that could achieve desirable security and privacy attributes and can also be applicable for practical applications. The contributions are listed below:

- To overcome the security weaknesses of the previous schemes, we have proposed an efficient and more secure WSNs authentication scheme that can preserve all the original merits of the related schemes and withstands the possible known attacks.

- To strengthen UAKA, the security analysis using the BAN logic has been presented. Using the informal security analysis, we have also shown that UAKA can resist numerous security attacks which include the attacks found in the previous schemes.

- Furthermore, the performance of UAKA is comparable to the related existing schemes.

2. PRELIMINARIES

This section provides preliminaries for the targeting WSN environment and threat model. They are important to understand UAKA.

2.1. WSN Model

Fig. (**1**) shows our proposed target WSN model. The proposed model consists of three types of entities, sensor nodes, GWNs and users. Their roles are defined as follows

- Sensor nodes: They are responsible for sensing the real-time data and forward them to the nearest GWN node directly.

- GWNs: They are responsible for receiving and forwarding the relevant data to the user and sensor node. Furthermore, they keep a database of sensor nodes to be related among GWNs.

- Users: They can access the sensed data of the sensor node through GWN after performing mutual authentication and key agreement.

GWN and sensor nodes are stationary after deployment, which is shown in Fig. (**1**). As mentioned in [15-17], the receiver end can measure the distance based on the received signal strength. Therefore, it is our valid assumption that all the deployed sensor nodes execute registration phase to the nearest GWN. In order to access the desired sensor node, the user can execute registration phase to any one of GWNs of our WSN model. While a user completes the registration procedure to any one of GWNs, called as home GWN (HGWN) and rest of the others are foreign GWNs (FGWN) with respect to that user. It is our effortless contribution that the user can access all GWNs of WSNs, although he (or she) has performed registration to only one HGWN. There are two scenarios in Fig. (**1**), which are for users A and B. The first case is for the situation when he (or she) wants to access sensor node in HGWN. A can communicate with GWN_1 as his (or her) HGWN to access data from SN_{11}. However, if the GWN could not find the target sensor node in its own database, it checks the sensor node and GWN database and forwards the

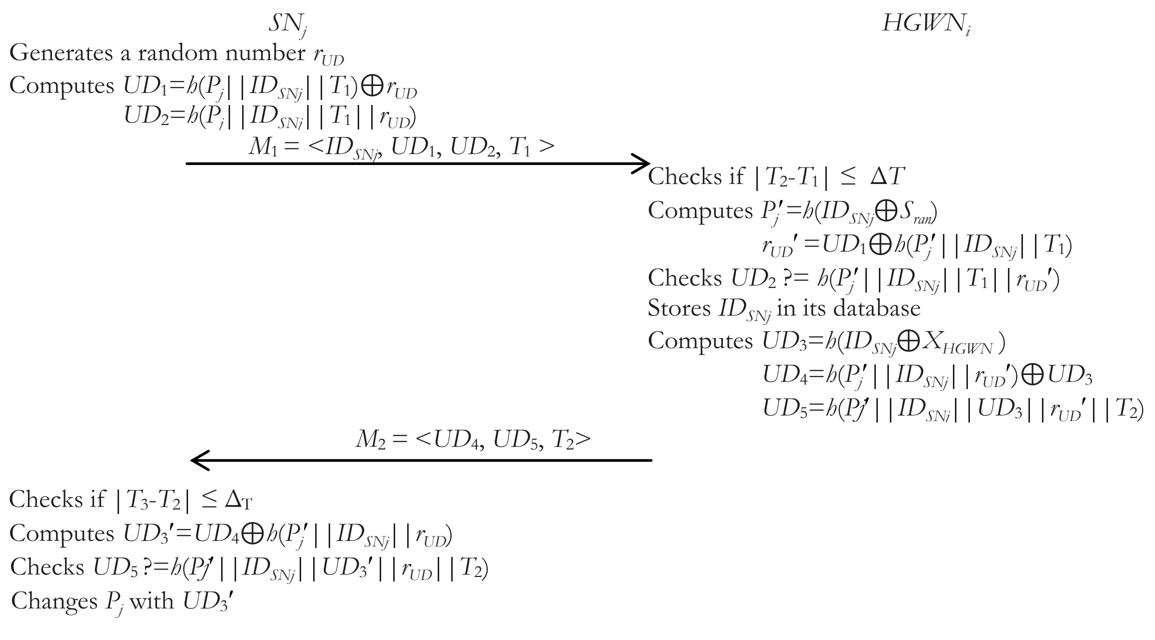

SN_j
$HGWN_i$
Generates a random number r_{UD}
Computes $UD_1=h(P_j||ID_{SNj}||T_1)\oplus r_{UD}$
$UD_2=h(P_j||ID_{SNj}||T_1||r_{UD})$
$M_1 = <ID_{SNj}, UD_1, UD_2, T_1 >$
Checks if $|T_2-T_1| \le \Delta T$
Computes $P_j'=h(ID_{SNj}\oplus S_{ran})$
$r_{UD}' =UD_1\oplus h(P_j'||ID_{SNj}||T_1)$
Checks UD_2 ?= $h(P_j'||ID_{SNj}||T_1||r_{UD}')$
Stores ID_{SNj} in its database
Computes $UD_3=h(ID_{SNj}\oplus X_{HGWN})$
$UD_4=h(P_j'||ID_{SNj}||r_{UD}')\oplus UD_3$
$UD_5=h(Pj'||ID_{SNi}||UD_3||r_{UD}'||T_2)$
$M_2 = <UD_4, UD_5, T_2>$
Checks if $|T_3-T_2| \le \Delta_T$
Computes $UD_3'=UD_4\oplus h(P_j'||ID_{SNj}||r_{UD})$
Checks UD_5 ?=$h(Pj'||ID_{SNj}||UD_3'||r_{UD}||T_2)$
Changes P_j with UD_3'

Fig. (2). Sensor node registration phase.

request to the target GWN as in case *B*. It is recommendable that the user cannot directly access the desired sensor nodes but only via GWNs.

2.2. Threat Model

In this threat model, we discuss some widely accepted valid assumptions regarding user authenticated key agreement scheme.

- An attacker can extract the information from the smart card by examining the power consumption or leaked information [18-19].

- An attacker has ability to eavesdrop all the communications between the parties in WSN over a public channel.

- An attacker has the potential to modify, delete, redirect and resent the eavesdropped transmitted messages.

- An attacker can be a legal user or an outsider in any system.

- An attacker can guess low entropy password and identity individually easily but guessing two secret parameters at the same time are computationally infeasible in polynomial time.

- Practically, it is assumed that the scheme used in the authentication system is known to the attacker.

- Kerckhoffs's principle: A cryptosystem should be secure even if everything about the system, except the key, is public knowledge [20].

3. UNLINKABLE USER AUTHENTICATED KEY AGREEMENT FOR MULTI-GATEWAY WSNS

This section proposes an unlinkable user authenticated key agreement for multi-gateway WSNs, which is denoted as UAKA. UAKA uses a similar concept with Srinivas et al.'s scheme in [14] but provides further security and privacy aspects. UAKA has six phases, system setup phase, registration phase, system environment phase, login authentication and key.

3.1. System Setup Phase

It is an off-line mode, where a system administrator (*SA*) generates the identity and security parameters for each sensor nodes (SNs). First, *SA* generates the identities { $ID_{SN}1$, $ID_{SN}2$, $\cdots$, ID_{SNm} } for each *SN* { SN_1, SN_2, …, SN_m } such that no two distinct SNs will get same identity. Then, *SA* computes $P_j=h(ID_{SNj}\oplus S_{ran})$ for $1 \le j \le m$, where S_{ran} is a random secret known to all GWNs. SA stores $<ID_{SNj}, P_j>$ for $1 \le j \le m$ into the memory of SNs before their deployment.

3.2. Registration Phase

It is divided into two sub-phases, *SN* registration and user registration. *SN* should be registered to its H_{GWN} only once right after the deployment for the security reason. Also, for a user to get the services from any *SN*, any user needs to be registered. Once the user gets registered, he (or she) will be able to connect with the opted *SN*. Both users and SNs undergo this registration process, respectively.

3.2.1. Sensor Node Registration Phase

Soon after the deployment, *SN* has to be one of GWNs realm by applying this phase. This is done through an open channel. SN registration phase is outlined in Fig. (**2**) and the detailed phase is as follows

SR1. SN_j generates a random number r_{UD} and computes $UD_1=h(P_j||ID_{SNj}||T_1)$ and $UD_2=h(P_j||ID_{SNj}||T_1||r_{UD})$, where T_1 is the time stamp of SN_j. SN_j sends a registration request message $M_1=<ID_{SNj}, UD_1, UD_2, T_1>$ to the nearest *HGWN* through an open channel.

SR2. *HGWN* checks $|T_2 - T_1| \le \Delta T$, where ΔT is the predefined permitted transmission delay. Only if it holds, *HGWN* computes $P_j'=h(ID_{SNj}\oplus S_{ran})$ and $r_{UD}'=UD_1\oplus h(P_j'||ID_{SNj}||T_1)$. After that *HGWN* verifies if UD_2 equals to $h(P_j'||ID_{SNj}||T_1||r_{UD}')$. *HGWN* terminates the session if the verification fails.

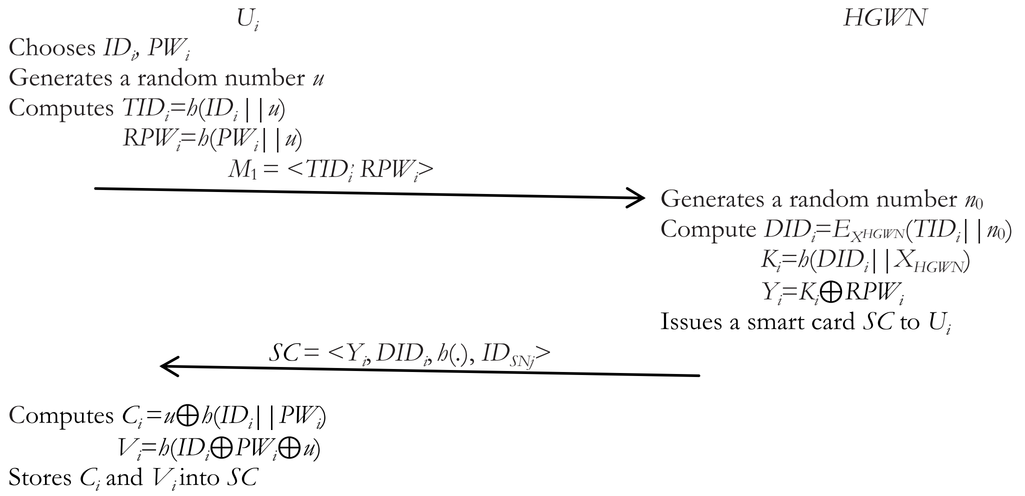

U_i
$HGWN$
Chooses ID_i, PW_i
Generates a random number u
Computes $TID_i=h(ID_i||u)$
$RPW_i=h(PW_i||u)$
$M_1=<TID_i, RPW_i>$
Generates a random number n_0
Compute $DID_i=E_{X_{HGWN}}(TID_i||n_0)$
$K_i=h(DID_i||X_{HGWN})$
$Y_i=K_i \oplus RPW_i$
Issues a smart card SC to U_i
$SC=<Y_i, DID_i, h(.), ID_{SNj}>$
Computes $C_i=u \oplus h(ID_i||PW_i)$
$V_i=h(ID_i \oplus PW_i \oplus u)$
Stores C_i and V_i into SC

Fig. (3). User registration phase.

Otherwise, *HGWN* stores ID_{SNj} in its database and computes $UD_3=h(ID_{SNj} \oplus X_{HGWN})$, $UD_4=h(P_j'||ID_{SNj}||r_{UD}') \oplus UD_3$ and $UD_5=h(P_j'||ID_{SNj}||UD_3||r_{UD}'||T_2)$, which X_{HGWN} is a 1024 bits secret key of *HGWN*. Then *HGWN* sends $M_2=<UD_4, UD_5, T_2>$ to SN_j.

SR3. Upon receiving the message, SN_j checks if $|T_3 - T_2| \leq \Delta T$. Only if it satisfies, SN_j computes $UD_3'=UD_4 \oplus h(P_j'|| ID_{SNj}||r_{UD})$ and verifies if UD_5 equals to $h(Pj'||ID_{SNj}|| UD_3'||r_{UD}||T_2)$. Only if the verification holds, SN_j changes P_j with UD_3'.

This phase has two major functions, to update the secret parameter S_{ran} in P_j so that every *HGWN* has its own secret parameter and also to make sure that *HGWN* knows which SN_j is in its region.

3.2.2. User Registration Phase

User completes his (or her) registration at *HGWN*, and achieves personalized security parameters to access *SN*. The registration phase is discussed in Fig. (**3**) and the detailed description is as follows

UR1. U_i selects his (or her) identity ID_i, password PW_i and a random number u. U_i computes $TID_i=h(ID_i||u)$ and $RPW_i=h(PW_i||u)$, and then submits a registration message $M_1=<TID_i, RPW_i>$ to *HGWN* via a secure channel.

UR2. On receiving the registration request, *HGWN* generates a random number n_0, encrypts $DID_i=E_{X_{HGWN}}(TID_i||n_0)$, and computes $K_i=h(DID_i||X_{HGWN})$ and $Y_i=K_i \oplus RPW_i$. *HGWN* issues a smart card *SC* for U_i, such that $SC=<Y_i, DID_i, h(\cdot), ID_{SNj}>$ and sends it to U_i.

UR3. Upon receiving *SC*, U_i computes $C_i=u \oplus h(ID_i||PW_i)$ and $V_i=h(ID_i \oplus PW_i \oplus u)$. U_i stores C_i and V_i into *SC*.

3.3. System Environment Phase

HGWN maintains the public directory of all the SNs. So, whenever a registered user, U_i wants to get services from an SN, SN_j, U_i can pick the appropriate SN_j's identity in the service environment of WSNs. In order to access the services, U_i first initiates the login session using his (or her) smart card. The authenticity of U_i is verified in the smart card authentication. Once the legitimacy of U_i is verified, a login message is forwarded to the *HGWN* which includes SN's identity, ID_{SNj}, where the existence of ID_{SNj} is checked in its database. If ID_{SNj} exists in *HGWN*'s database.

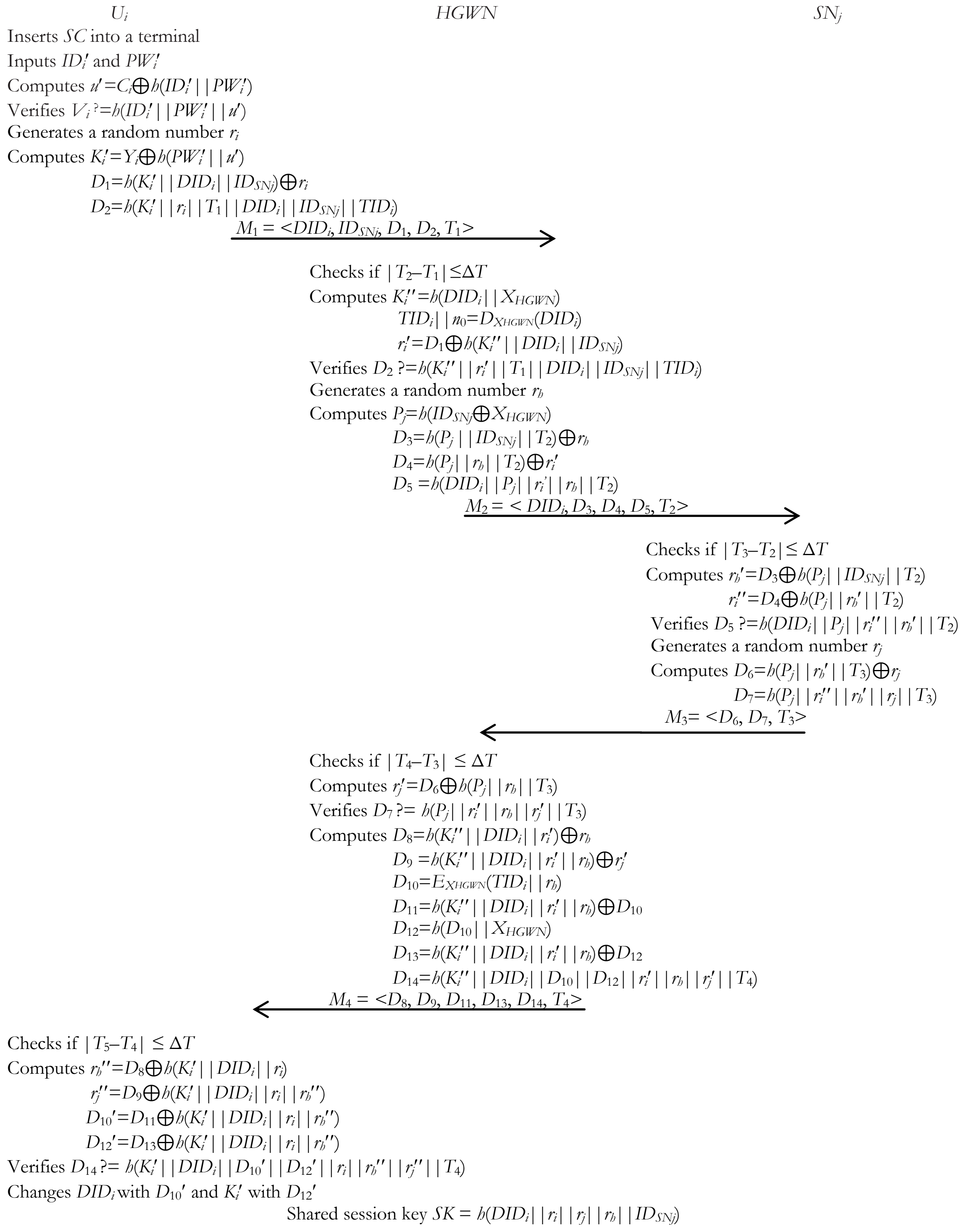
3.4. Login and Authenticated Key Agreement Phase

HGWN maintains the public directory, which comprises the SNs identities in WSN. This enables any user to select a SN as per his (or her) requirement. To get services from SN_j, U_i extracts ID_{SNj} from the public directory of *HGWN*. The registered U_i inserts his (or her) smart card into a card reader to initiate the login and authenticated key agreement phase. This phase can be seen as two phases. Once the legitimacy of U_i is verified, a login message is forwarded to the proper *HGWN* through the public channel in order to login to the desired SN_j. The procedure of the login and authenticated key agreement phase is described in the following sub-sections.

3.4.1. Login Phase

First, the smart card needs to verify the legitimacy of U_i. For the valid U_i, smart card processes the login request. U_i executes the login request as follows

LG1. To start the login process, U_i inserts *SC* into the terminal and inputs his (or her) ID_i' and PW_i'. Then *SC* computes $u'=C_i \oplus h(ID_i'||PW_i')$ and checks if V_i equals to $h(ID_i' \oplus PW_i' \oplus u')$. *SC* inquires for sensor identity as per requirement to *HGWN* upon the successful verification, by observing the user requirement and sensors availability, *HGWN* sends the available sensor's identity ID_{SNj} to U_i. Only if the verification holds, *SC* generates a random number r_i, and computes $K_i'=Y_i \oplus h(PW_i'||u')$, $D_1=h(K_i'||DID_i||ID_{SNj}) \oplus r_i$ and $D_2=h(K_i'||r_i||T_1||DID_i||ID_{SNj}||TID_i)$. After that, U_i sends a login message $M_1=<DID_i, ID_{SNj}, D_1, D_2, T_1>$ to *HGWN*.



U_i $HGWN$ SN_j

Inserts SC into a terminal
Inputs ID_i' and PW_i'
Computes $u'=C_i \oplus h(ID_i' \| PW_i')$
Verifies $V_i ?= h(ID_i' \| PW_i' \| u')$
Generates a random number r_i
Computes $K_i'=Y_i \oplus h(PW_i' \| u')$
$D_1=h(K_i' \| DID_i \| ID_{SNj}) \oplus r_i$
$D_2=h(K_i' \| r_i \| T_1 \| DID_i \| ID_{SNj} \| TID_i)$

$M_1 = <DID_i, ID_{SNj}, D_1, D_2, T_1>$ →

Checks if $|T_2 – T_1| \leq \Delta T$
Computes $K_i''=h(DID_i \| X_{HGWN})$
$TID_i \| n_0=D_{XHGWN}(DID_i)$
$r_i'=D_1 \oplus h(K_i'' \| DID_i \| ID_{SNj})$
Verifies $D_2 ?= h(K_i'' \| r_i' \| T_1 \| DID_i \| ID_{SNj} \| TID_i)$
Generates a random number r_h
Computes $P_j=h(ID_{SNj} \oplus X_{HGWN})$
$D_3=h(P_j \| ID_{SNj} \| T_2) \oplus r_h$
$D_4=h(P_j \| r_h \| T_2) \oplus r_i'$
$D_5 =h(DID_i \| P_j \| r_i' \| r_h \| T_2)$

$M_2 = < DID_i, D_3, D_4, D_5, T_2>$ →

Checks if $|T_3 – T_2| \leq \Delta T$
Computes $r_h'=D_3 \oplus h(P_j \| ID_{SNj} \| T_2)$
$r_i''=D_4 \oplus h(P_j \| r_h' \| T_2)$
Verifies $D_5 ?= h(DID_i \| P_j \| r_i'' \| r_h' \| T_2)$
Generates a random number r_j
Computes $D_6=h(P_j \| r_h' \| T_3) \oplus r_j$
$D_7=h(P_j \| r_i'' \| r_h' \| r_j \| T_3)$

← $M_3= <D_6, D_7, T_3>$

Checks if $|T_4 – T_3| \leq \Delta T$
Computes $r_j'=D_6 \oplus h(P_j \| r_h \| T_3)$
Verifies $D_7 ?= h(P_j \| r_i' \| r_h \| r_j' \| T_3)$
Computes $D_8=h(K_i'' \| DID_i \| r_i') \oplus r_h$
$D_9 =h(K_i'' \| DID_i \| r_i' \| r_h) \oplus r_j'$
$D_{10}=E_{XHGWN}(TID_i \| r_h)$
$D_{11}=h(K_i'' \| DID_i \| r_i' \| r_h) \oplus D_{10}$
$D_{12}=h(D_{10} \| X_{HGWN})$
$D_{13}=h(K_i'' \| DID_i \| r_i' \| r_h) \oplus D_{12}$
$D_{14}=h(K_i'' \| DID_i \| D_{10} \| D_{12} \| r_i' \| r_h \| r_j' \| T_4)$

← $M_4 = <D_8, D_9, D_{11}, D_{13}, D_{14}, T_4>$

Checks if $|T_5 – T_4| \leq \Delta T$
Computes $r_h''=D_8 \oplus h(K_i' \| DID_i \| r_i)$
$r_j''=D_9 \oplus h(K_i' \| DID_i \| r_i \| r_h'')$
$D_{10}'=D_{11} \oplus h(K_i' \| DID_i \| r_i \| r_h'')$
$D_{12}'=D_{13} \oplus h(K_i' \| DID_i \| r_i \| r_h'')$
Verifies $D_{14} ?= h(K_i' \| DID_i \| D_{10}' \| D_{12}' \| r_i \| r_h'' \| r_j'' \| T_4)$
Changes DID_i with D_{10}' and K_i' with D_{12}'

Shared session key $SK = h(DID_i \| r_i \| r_j \| r_h \| ID_{SNj})$

Fig. (4). Login phase and authenticated key agreement phase.

3.4.2. Authenticated Key Agreement Phase

After receiving the login message from U_i, $HGWN$ checks whether the requested SN_j is in the registered sensor list or not by checking its database. Only if SN_j is in its database, $HGWN$ executes the authenticated key agreement phase. Otherwise, it forwards the message to the appropriate GWN. The message exchange of login and authenticated key agreement is discussed in Fig. (**4**) and the details of this phase are as follows:

AK1. On receiving the login message $<DID_i, ID_{SNj}, D_1, D_2, T_1>$ at T_2, $HGWN$ checks the freshness of the message as $|T_2 – T_1| \leq \Delta T$. Only if the verification passes, $HGWN$ computes decrypts $TID_i \| n_0=D_{XHGWN}(DID_i)$, computes $K_i''=h(TID_i \| X_{HGWN})$ and retrieves $r_i'=D_1 \oplus h(K_i'' \| DID_i \| ID_{SNj})$. Then, $HGWN$ verifies if D_2 equals to $h(K_i'' \| r_i' \| T_1 \| DID_i \| ID_{SNj} \| TID_i)$. Only if the verification holds, *HGWN* authenticates U_i. Otherwise, the connection is terminated. *HGWN* generates a random number r_h and computes $P_j=h(ID_{SNj} \oplus X_{HGWN})$, $D_3=h(P_j||ID_{SNj}||T_2)\oplus r_h$, $D_4=h(P_j||r_h||T_2)\oplus r_i'$ and $D_5=h(DID_i||P_j||r_i'||r_h||T_2)$. *HGWN* contracts the login message $M_2=<DID_i, D_3, D_4, D_5, T_2>$ and sends it to SN_j.

AK2. On receiving the message at T_3, SN_j checks the freshness of the message as $|T_3 - T_2| \leq \Delta T$. Only if the verification is valid, SN_j extracts $r_h'=D_3\oplus h(P_j||ID_{SNj}||T_2)$, $r_i''=D_4\oplus h(P_j||r_h'||T_2)$, and then verifies if D_5 equals to $h(DID_i||P_j||r_i''||r_h'||T_2)$. If the verification does not hold, the connection is aborted. Otherwise, SN_j generates a random number r_j and computes $D_6=h(P_j||r_h'||T_3)\oplus r_j$ and $D_7=h(P_j||r_i''||r_h'||r_j||T_3)$. Then, SN_j sends the message $M_3=<D_6, D_7, T_3>$ to *HGWN*.

AK3. On receiving the message at T_4, *HGWN* checks $|T_4 - T_3| \leq \Delta T$. If the verification is valid, *HGWN* computes $r_j'=D_6\oplus h(P_j||r_h||T_3)$ and verifies if D_7 equals to $h(P_j||r_i'||r_h||r_j'||T_3)$. If the verification does not hold, the connection is aborted. Otherwise, *HGWN* computes $D_8=h(K_i''||DID_i||r_i')\oplus r_h$, $D_9=h(K_i''||DID_i||r_i'||r_h)\oplus r_j'$, $D_{10}=E_{XHGWN}(TID_i||r_h)$, $D_{11}=h(K_i''||DID_i||r_i'||r_h)\oplus D_{10}$ and $D_{12}=h(K_i''||DID_i||D_{10}||r_i'|| r_h||r_j'||T_4)$ and sends a message $M_4=<D_8, D_9, D_{10}, D_{11}, D_{12}, T_4>$ to U_i.

AK4. On receiving the message at T_5, U_i checks $|T_5 - T^4| \leq \Delta T$. If the verification is valid, U_i computes $r_h''=D_8\oplus h(K_i||DID_i||r_i)$, $r_j''=D_9\oplus h(K_i||DID_i||r_i||r_h'')$ and $D_{10}'=D_{11}\oplus h(K_i||DID_i||r_i||r_h'')$ and then verifies if D_{12} equals to $h(K_i||DID_i||D_{10}'||r_i||r_h''||r_j''||T_4)$. If the verification does holds, U_i changes DID_i with D_{10}'. Hence, it is confirmed that SN_j is authentic. But if not, the connection is aborted. On the success of mutual authentication, a session key $SK=h(DID_i||r_i||r_j||r_h||ID_{SNj})$ is constructed by involved entities in the system.

3.5. Dynamic Node Addition Phase

It may happen that a new sensor node needs to be added over the target field as and when required, after the establishment of WSNs. So, *SA* deploys the new sensor node over the target region by performing the system setup phase in off-line mode. Then after, the newly added sensor node under-goes the sensor node registration phase and introduces the new sensor node into the setup network model.

3.6. Password Change Phase

In smart card based authentication, protocols should be able to address password related attack so that user with valid smart card and personal credentials can initiate the password change phase. Additionally, user should be able to choose and change the password without interaction with *SA* or *HGWN*, which is to provide user-friendly password selection and change. The proposed password change phase requires user can change the password without interaction with the other network entity.

A user U_i with valid credentials and smart card can initiate the password change by inputting ID_i' and PW_i'. *SC* computes $u'=C_i\oplus h(ID_i'||PW_i')$. To resist password related attack, *SC* verifies if V_i equals to $h(ID_i'||PW_i'||u')$. Using this condition, *SC* identifies the correctness of user credentials. If verification holds, *SC* asks for the new password PW_{new} to U_i. On receiving PW_{new}, *SC* computes $RPW_{new}=h(PW_{new}||u')$ and updates $Y_i=Y_i \oplus RPW_i \oplus RPW_{new}$, $C_i=C_i\oplus h(ID_i||PW_i)\oplus h(ID_i'||PW_{new})$ and $V_i= h(ID_i'\oplus PW_{new}\oplus u')$ on *SC*.

4. ANALYSIS

This section provide security analysis and performance analysis. First of all, we provide BAN logic analysis and informal security analysis to show the security and privacy of UAKA. Performance analysis is focused on computational and communicational overheads and provides comparisons of UAKA with the related schemes.

4.1. BAN Logic Analysis

In this section, we provide a formal protocol analysis of our proposed UAKA using the BAN logic method [21]. The BAN logic is used to verify the correctness of the authentication scheme with key agreement. UAKA correctness refers to the communication parties: a legal user U_i, GWN and an accessed sensor node S_j who share a fresh shared session key with each other after the scheme is achieved. The formal analysis of UAKA using BAN logic involves following steps:

(1) Converting original scheme statements to their idealized form.

(2) Determining the assumptions about the initial state of the system.

(3) Representation of the state of the system after executing each statement as logical assertions by attaching logical formulas to each statement.

(4) Application of logical postulates to assumptions and assertions.

The following notations are used in formal security analysis using the BAN logic:

- $Q \mid\equiv X$: Principal Q believes the statement X.
- $\#(X)$: Formula X is fresh.
- $Q \mid\Rightarrow X$: Principal Q has jurisdiction over the statement X.
- $|\xrightarrow{K} Q$: Principal Q has a public key K.
- $Q \triangleleft X$: Principal Q sees the statement X.
- $Q \mid\sim X$: Principal Q once said the statement X.
- (X, Y): Formula X or Y is one part of the formula (X, Y).

- $\langle P \rangle_Q$: Formula *P* combined with the formula *Q*.
- $Q \overset{SK}{\leftrightarrow} R$: Principal *Q* and *R* may use the shared session key, *SK* to communicate among each other. The session key *SK* is good, in that it will never be discovered by any principal except *Q* and *R*.

In addition, the following four BAN logic rules are used to prove that UAKA provides a secure mutual authentication among U_i, *HGWN* and S_j :

Rule 1. **Message-meaning rule**: $\dfrac{R\,|\!\equiv R \overset{Y}{\leftrightarrow} S,\ R \lhd <X>_Y}{R\,|\!\equiv S\,|\!\sim X}$

Rule 2. **Nonce-verification rule**: $\dfrac{R|\!\equiv \#(X),\ R\,|\!\equiv S\,|\!\sim X}{R|\!\equiv S\,|\!\equiv X}$

Rule 3. **Jurisdiction rule**: $\dfrac{R\,|\!\equiv S\,|\!\Rightarrow X,\ R\,|\!\equiv S\,|\!\equiv X}{R|\!\equiv X}$

Rule 4. Freshness-concatenation rule: $\dfrac{R\,|\!\equiv \#(X)}{R|\!\equiv \#(X,Y)}$

In order to show that UAKA provides secure mutual authentication between among U_i, *HGWN* and S_j, we need to achieve the following goals:

Goal 1: $U_i| \equiv (U_i \overset{SK}{\leftrightarrow} S_j)$

Goal 2: $S_j| \equiv (S_j \overset{SK}{\leftrightarrow} U_i)$

Goal 3: $U_i| \equiv S_j| \equiv (S_j \overset{SK}{\leftrightarrow} U_i)$

Goal 4: $S_j| \equiv U_i| \equiv (U_i \overset{SK}{\leftrightarrow} S_j)$

Idealized form: The arrangement of the transmitted messages among U_i, *HGWN* and S_j in UAKA to the idealized forms is as follows:

Message 1. $U_i \rightarrow HGWN$: $<DID_i>_{X_{HGWN}}$, $<ID_{SNj}>$, $<D_1>_{Ki}$, $<D_2>_{Ki}$, $<T_1>$

Message 2. $HGWN \rightarrow S_j$: $<DID_i>_{X_{HGWN}}$, $<D_3>_{Pj}$, $<D_4>_{Pj}$, $<D_5>_{Pj}$, $<T_2>$

Message 3. $S_j \rightarrow HGWN$: $<D_6>_{Pj}$, $<D_7>_{Pj}$, $<T_3>$

Message 4. $HGWN \rightarrow U_i$: $<D_8>_{Ki}$, $<D_9>_{Ki}$, $<D_{11}>_{Ki}$, $<D_{13}>_{Ki}$, $<D_{14}>_{Ki}$, $<T_2>$

Assumptions: The following are the initial assumptions of UAKA:

A1: $U_i| \equiv \#(r_i, T_1)$

A2: $HGWN| \equiv \#(r_h, T_2, T_4)$

A3: $S_j| \equiv \#(r_i, T_3)$

A4: $U_i| \equiv(U_i \overset{(K_i)}{\longleftrightarrow} HGWN)$

A5: $HGWN| \equiv(HGWN \overset{(K_i)}{\longleftrightarrow} U_i)$

A6: $HGWN| \equiv(HGWN \overset{P_j}{\leftrightarrow} S_j)$

A7: $S_j| \equiv(S_j \overset{P_j}{\leftrightarrow} HGWN)$

A8: $U_i\ | \equiv S_j| \Longrightarrow U_i \overset{SK}{\leftrightarrow} S_j$

A9: $S_j| \equiv U_i\ | \Longrightarrow S_j \overset{SK}{\leftrightarrow} U_i$

Proof:

In the following, we prove the test goals in order to show the secure authentication using the BAN logic rules and the assumptions.

Based on Message 1, we could derive:

Step 1. $HGWN \lhd <DID_i>_{X_{HGWN}}$, $<ID_{SNj}>$, $<D_1>_{Ki}$, $<D_2>_{Ki}$, $<T_1>$
According to assumption A4 and the message meaning rule, we could get:

Step 2. $HGWN| \equiv U_i| \sim (<DID_i>_{X_{HGWN}}$, $<ID_{SNj}>$, $<D_1>_{Ki}$, $<D_2>_{Ki}$, $<T_1>)$
According to assumption A1 and the freshness concatenation rule, we could get:

Step 3: $HGWN| \equiv \#(<DID_i>_{X_{HGWN}}$, $<ID_{SNj}>$, $<D_1>_{Ki}$, $<D_2>_{Ki}$, $<T_1>)$
According to Step 2, Step 3 and the nonce verification rule, we could get:

Step 4. $HGWN| \equiv U_i| \equiv(<DID_i>_{X_{HGWN}}$, $<ID_{SNj}>$, $<D_1>_{Ki}$, $<D_2>_{Ki}$, $<T_1>)$
According to Step 4, assumption A4 and the believe rule, we could get:

Step 5. $HGWN| \equiv U_i| \equiv(U_i \overset{(K_i)}{\longleftrightarrow} HGWN)$
According to the jurisdiction rule, we could get:

Step 6. $HGWN| \equiv(HGWN \overset{(K_i)}{\longleftrightarrow} U_i)$
Based on Message 2, we could derive

Step 7. $S_j \lhd <DID_i>_{X_{HGWN}}$, $<D_3>_{Pj}$, $<D_4>_{Pj}$, $<D_5>_{Pj}$, $<T_2>$
According to assumption A7 and the message meaning rule, we could get:

Step 8. $S_j| \equiv HGWN| \sim (<DID_i>_{X_{HGWN}}$, $<D_3>_{Pj}$, $<D_4>_{Pj}$, $<D_5>_{Pj}$, $<T_2>)$
According to assumption A2 and the freshness concatenation rule, we could get:

Step 9: $S_j| \equiv \#(<DID_i>_{X_{HGWN}}$, $<D_3>_{Pj}$, $<D_4>_{Pj}$, $<D_5>_{Pj}$, $<T_2>)$
According to Step 8, Step 9 and the nonce verification rule, we could get:

Step 10. $S_j| \equiv HGWN| \equiv(<DID_i>_{X_{HGWN}}$, $<D_3>_{Pj}$, $<D_4>_{Pj}$, $<D_5>_{Pj}$, $<T_2>)$
According to Step 10, assumption A6 and the believe rule, we could get:

Step 11. $S_j| \equiv HGWN| \equiv(HGWN \overset{P_j}{\leftrightarrow} S_j)$
According to the jurisdiction rule, we could get:

Step 12. $S_j| \equiv(S_j \overset{P_j}{\leftrightarrow} HGWN)$
According to Step 8, Step 9, Step 10 and the nonce verification rule, we could get:

Step 13. $S_j| \equiv U_i| \equiv(U_i \overset{SK}{\leftrightarrow} S_j)$ **(Goal 4)**
According to assumption A8 and the jurisdiction rule, we could get:

Step 14. $S_j| \equiv(S_j \overset{SK}{\leftrightarrow} U_i)$ **(Goal 2)**
Based on Message 3, we could derive

Step 15. $HGWN \lhd <D_6>_{Pj}$, $<D_7>_{Pj}$, $<T_3>$

According to assumption A6 and the message meaning rule, we could get:

Step 16. $HGWN|\equiv S_j|\sim(<D_6>_{Pj}, <D_7>_{Pj}, <T_3>)$
According to assumption A3 and the freshness concatenation rule, we could get:

Step 17: $HGWN|\equiv \#(<D_6>_{Pj}, <D_7>_{Pj}, <T_3>)$
According to Step 16, Step 17 and the nonce verification rule, we could get:

Step 18. $HGWN|\equiv S_j|\equiv(<D_6>_{Pj}, <D_7>_{Pj}, <T_3>)$
According to Step 18, assumption A7 and the believe rule, we could get:

Step 19. $HGWN|\equiv S_j|\equiv(S_j \overset{P_j}{\leftrightarrow} HGWN)$
According to Step 16, Step 17, Step 18 and the nonce verification rule, we could get:

Step 20. $HGWN|\equiv S_j|\equiv(S_j \overset{SK}{\longleftrightarrow} HGWN)$
According to assumption A10 and the jurisdiction rule, we could get:

Step 21. $HGWN|\equiv (HGWN \overset{SK}{\longleftrightarrow} S_j)$
Based on Message 4, we could derive

Step 22. $U_i \lhd <D_8>_{Ki}, <D_9>_{Ki}, <D_{11}>_{Ki}, <D_{13}>_{Ki}, <D_{14}>_{Ki}, <T_2>$
According to assumption A4 and the message meaning rule, we could get:

Step 23. $U_i|\equiv HGWN|\sim(<D_8>_{Ki}, <D_9>_{Ki}, <D_{11}>_{Ki}, <D_{13}>_{Ki}, <D_{14}>_{Ki}, <T_2>)$
According to assumption A2 and the freshness concatenation rule, we could get:

Step 24: $U_i|\equiv\#(<D_8>_{Ki}, <D_9>_{Ki}, <D_{11}>_{Ki}, <D_{13}>_{Ki}, <D_{14}>_{Ki}, <T_2>)$
According to Step 23, Step 24 and the nonce verification rule, we could get:

Step 25. $U_i|\equiv HGWN|\equiv(<D_8>_{Ki}, <D_9>_{Ki}, <D_{11}>_{Ki}, <D_{13}>_{Ki}, <D_{14}>_{Ki}, <T_2>)$
According to Step 25, assumption A5 and the believe rule, we could get:

Step 26. $U_i|\equiv HGWN|\equiv(HGWN \overset{K_i}{\leftrightarrow} U_i)$
According to Step 23, Step 24, Step 25 and the nonce verification rule and the jurisdiction rule, we could get:

Step 27. $U_i|\equiv S_j|\equiv(S_j \overset{SK}{\longleftrightarrow} U_i)$ (**Goal 3**)
According to assumption A8 and the jurisdiction rule, we could get:

Step 28. $U_i|\equiv(U_i \overset{SK}{\longleftrightarrow} S_j)$ (**Goal 1**)

According to Steps 14 and 28, UAKA successfully achieves both goals (**Goals 1 and 2**). Both U_i and S_j believes that they share a common session key $SK=h(DID_i||r_i||r_j||r_h||ID_{SNj})$.

4.2. Informal Security Analysis

Although it is important to provide a formal security proof on any cryptographic protocol, the formal security proof of protocols remains one of the most challenging issues for cryptography research. Until now, a simple, efficient and convincing formal methodology for correctness analysis on security protocols is still an important subject of research and an open problem. Because of these reasons, most protocols have been demonstrated with a simple proof. This section follows the security analysis approaches used in [22]. As shown in Table **1**, the security analysis is focused on verifying the overall security requirements for UAKA, including passive and active attacks, as follows.

Table 1. Comparison of Security Features.

Security Attributes	[12]	[13]	[14]	UAKA
Masquerading attack	Weak	Weak	Weak	Strong
Replay attack	Strong	Strong	Strong	Strong
Trace attack	Weak	Weak	Weak	Strong
Insider attack	Strong	Strong	Strong	Strong
Password guessing attack	Weak	Weak	Weak	Strong
DoS attack	Strong	Strong	Weak	Strong
Anonymity and unlinkability	Not	Not	Not	Provide

Proposition 1. UAKA is secure against $HGWN$ masquerading attack.

Proof: By definition this is the attack in which an attacker pretends to be a legitimate $HGWN$ and plays in between user and sensor node with the assumption that the attacker could obtain any messages transmitted in the previous sessions. In UAKA, the attacker could try to form $M_2=<D_3, D_4, D_5, T_2>$ or $M_4=<D_8, D_9, D_{11}, D_{13}, D_{14}, T_4>$ right after receiving $M_1=<DID_i, ID_{SNj}, D_1, D_2, T_1>$ from U_i for the trial of this attack. However, they are impossible to the attacker in UAKA because they require knowledge of the important secret key, X_{HGWN} of $HGWN$. Again to U_i, the attacker needs to form a correct $M_4=<D_8, D_9, D_{11}, D_{13}, D_{14}, T_4>$, which requires the knowledge of K_i'' where $K_i''=h(D_{12}'||X_{HGWN})$. Without the knowledge of X_{HGWN}, the attacker could not form the proper message M_1. In the other hand, against to SN_j, the attacker needs to form $M_2=<D_3, D_4, D_5, T_2>$, which requires the knowledge of P_j where $P_j=h(ID_{SNj}\oplus X_{HGWN})$. The attacker could not do anything to form the proper message with the same reason for U_i. There is no feasible way the attacker knows X_{HGWN} or P_j. Hence we can confirm that UAKA resists HGWN masquerading attack.

Proposition 2. UAKA is secure against SN_j masquerading attack.

Proof: With the similar definition of the attack on $HGWN$ and the assumption, to masquerade as a legitimate sensor node SN_j, an attacker needs to form a proper response message $M_3=<D_6, D_7, T_3>$ to X_{HGWN}. For the attacker to do this, he (or she) must have the knowledge of $P_j=h(ID_{SNj}\oplus X_{HGWN})$. However, it is not possible in UAKA as the attacker does not have the knowledge of the secret key X_{HGWN} of the involved parties. Thus it will be impossible for him (or her) to compute the message M_3 correctly. Therefore, UAKA can resist the sensor node masquerading attack.

Proposition 3. UAKA is secure against replay attack,

Proof: By definition replay attack is an attack where the attacker captures the previously transmitted messages and uses them during UAKA execution to make the receiver of the message believe that the transmitted message is from a legal entity. In order to justify UAKA resist from the replay attack, we assume that the attacker has captured the previous session messages of UAKA and later tries to transmit the same message to the targeted. In a replay attack, it does not matter if the attacker who intercepted the original message can read or decipher the key. All he (or she) has to do is capture and resent the entire thing — message and key — together. To counter this possibility, UAKA uses random session key, which is a type of code that is only valid for one transaction and cannot be used again. For an example, when preparing message one M_1=<DID_i, ID_{SNj}, D_1, D_2, T_1>, UAKA uses r_i as its random number while in when preparing message two M_2=<D_3, D_4, D_5, T_2>, *HGWN* generates r_h as its random number and SN_j generates r_j as its random number to send a message to *HGWN*. Another preventative measure for this type of attack is using time stamps on all messages as you we can see each and every message has a time stamp as it is sent, this prevents hackers from resenting messages sent longer ago than a certain length of time, thus reducing the window of opportunity for an attacker to eavesdrop, siphon off the message, and resent it. In specific, a time stamp and random number mechanisms are used to guarantee the freshness of each message. Following this we can conclude that UAKA's scheme is strong against replay attack.

Proposition 4. UAKA could withstand trace attack

Proof: Trace attack is an attack where the attacker can distinguish the messages communicated between entities by eavesdropping on a communication. For an attacker to achieve this, he (or she) intercepts two or more messages from two or more different sessions and checks whether they have something in come that can be computed by the attacker. If it happens, the attacker believes that these two messages belong to the same source, either from U_i or *HGWN*. However, the attacker cannot trace U_i, *HGWN*, SN_j after intercepting the communicating messages because UAKA's scheme updates DID_i and K_i apart from that he uses the one-way hash function which is infeasible for an attacker to compute important parameters such as X_{HGWN}.

Proposition 5. UAKA could withstand privileged insider attack

Proof: An insider attack is defined as a malicious attack perpetrated on a network or computer system by a person with the authorized system access. Practically, in UAKA, it is assumed that the *HGWN* is trusted. So, the *HGWN* provides confidentiality to the user's credentials, where leakage of any confidential parameters of the user is not permitted. But, it is observed that due to the presence of an insider, systems can get hacked. Therefore, the user's information such as identity and password should be kept secret such that the insider of the *HGWN* cannot gain control over the user's information. In UAKA, during the user's registration phase, instead of the original ID_i and PW_i we have transmitted the masked identity TID_i=h(ID_i||u) and password RPW_i=h(PW_i||u) to *HGWN*. Hence, extracting the user's password or identity by the insider of the *HGWN* is computationally infeasible due to the non-invertible property of the cryptographic one-way hash function. Therefore, UAKA can resist privileged insider attack.

Proposition 6. UAKA could withstand password guessing attack

Proof: A password guessing attack is an attack that consists of an attacker trying many passwords or pass phrases with the hope of eventually guessing correctly. We suppose that the user U_i's smart card was stolen by an attacker, then the attacker can extract the information stored on the smart card <Y_i, DID_i, $h(\cdot)$, ID_{SNj}, C_i, V_i> by using the method of power analysis, where V_i=h(ID_i||PW_i||u), C_i=u⊕h(ID_i||PW_i) and Y_i=K_i⊕RPW$_i$. The attacker needs to know u, ID_i and PW_i, where this information is known only to the user, and both user ID_i and PW_i are unknown to the attacker because they are well protected by the one way hash function. So, the attacker has no way to guess or exact user ID_i and PW_i at the same time, as it is computationally infeasible to guess the two parameters at the same time and also, due to the non-invertible one-way hash function property. Hence, there is nowhere for an attacker to update the PW_i of the user U_i. Therefore, UAKA is free from the stolen smart card attack.

Proposition 7. UAKA could withstand denial-of-service (DoS) attack diagram.

Proof: DoS attack is an attack in which the perpetrator seeks to make a machine or network resource unavailable to its intended users by temporarily or indefinitely disrupting services of a host connected to the Internet. In UAKA, we have three possibilities where a registered user could encounter DoS. However, UAKA is efficient to resist DoS attack in all scenario as follows: In first situation when a user inputs incorrect credentials (identity or password) unknowingly during login phase, however, the smart card can correctly verify the login credentials using the condition V_i ?= h(ID_i||PW_i||u). This ensures that only with the correct input of user credentials a login message M_1=<DID_i, ID_{SNj}, D_1, D_2, T_1> will be executed. Thus, there will not be occurrence of denial-of-service. Adversary may also try to engage sensors by replaying the messages so that valid user login attempt may deny or delayed, however, the transmitted messages M_2=<DID_i, D_3, D_4, D_5, T_2> includes the time stamp. The sensors verify the freshness of time stamp before professing the requests. This shows that a sensor can efficiently encounter the fake request in UAKA, which shows the security of UAKA against denial of service attack. The third situation is where an adversary can mount an application layer DoS attack, this is a form of DoS attack, where attackers target the application layer of the open systems interconnection (OSI) model. The attack over-exercises specific functions or features of a website with the intention to disable those functions or features. This application-layer attack is different from an entire network attack. However this is not feasible in our scheme since our scheme involves the use of one way-hash function which very difficult for an attacker to compute it.

Proposition 8. UAKA provides anonymity and unlinkability

Proof: Anonymity is a property of network security. An entity in a system has anonymity if no other entity can identify the first entity, nor is there any link back to the first entity that can be used, nor any way to verify that any two anonymous acts are performed by the same entity.

As shown in proposition 6, it is clear from UAKA that an attacker has no way to obtain or guess the identity ID_i of the user U_i as it is protected by the non-invertible cryptographic one way hash function not only that but also the use of pseudo-identity makes the system anonymous thus UAKA provides anonymity and also unlinkability.

4.3. Performance Analysis

In this section, we compare the performance and functionality features of UAKA with the related existing schemes proposed for WSNs. This evaluation gives an insight into the effectiveness of UAKA.

Table 2. Comparison of Communication and Performance Overhead.

Features	[12]	[13]	[14]	UAKA
Total number of messages	4	4/8	4/7	4
Computational overhead	$32T_h$	$20T_h/20T_h$	$29T_h/35T_h$	$2T_{SE}+30T_h$

Table **2** shows the communication overhead required during the login and authenticated key agreement phases between UAKA and the related schemes. The length of message has also effects to the communication overhead. However, the number of messages is much important factor to be compared. UAKA only requires 4 messages in any cases. Thereby, UAKA has a good property in the communication overhead concern compared to the other schemes. However, UAKA has a bit more computational overhead due to provide anonymity and unlinkability, which requires two symmetric key cryptosystem operations. Based on MIRACL library with 32-bit Windows 7 operating systems and Visual C++ 2008, symmetric key cryptosystem operation and hash function require 0.1303 ms and 0.0004 ms if advanced encryption standard and secure hash algorithm 1 are used [23].

5. CONCLUSION

In this paper, we have proposed an unlinkable user authenticated key agreement scheme for multi-gateway WSNs. Security validation of the proposed scheme has analyzed using BAN logic and informal cryptanalysis proofs the resilience of relevant security and privacy attacks. Even if WSNs are resource constrained, they should consider and provide privacy provision to their entities. To solve this requirement, the proposed scheme adopted the symmetric key cryptosystem operation, which requires a bit overhead than the other related schemes. However, it has a good property in communication overhead concern as shown in Table **2**.

ACKNOWLEDGEMENTS

The results in this paper are part of Solomon Kuonga Nkhoma's Master degree thesis. Hyunsung Kim is the corresponding author. This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2017R1D1A1B04032598).

CONFLICT OF INTEREST

The author declares that there are no conflicts of interest.

REFERENCES

[1]. AL-Mousawi AJ, AL-Hassani KA. A survey in wireless sensor network for explosives detection. Computers & Electrical Engineering 2017; in press: https://doi.org/10.1016/j.compeleceng.2017.11.013.

[2]. Romer K, Mattern F. The design space of wireless sensor networks. IEEE Wireless Communications 2004; 11(6): 54-61.

[3]. Atzori L, Lera A, Morabito G. The Internet of Things: A Survey. Computer Networks 2010; 54(15): 2787-2805.

[4]. Song T, Jung J, Kang D, Kim H, Won D. Cryptanalysis of an Authentication Scheme for Multi-Gateway Wireless Sensor Networks. Proceedings of the Twelfth International Conference on Digital Information Management; 2017 Sep 12-14; Kyushu University, Fukuoka, Japan. IEEE 2017.

[5]. Wong, KHM, Zheng Y, Cao J, Wang S. A dynamic user authentication scheme for wireless sensor networks. Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing; 2006 June 5-7; Taichung, Taiwan. IEEE 2006.

[6]. Das ML. Two-factor user authentication in wireless sensor networks. IEEE Transactions on Wireless Communications 2009; 8(3): 1086-1090.

[7]. Khan MK, Alghathbar K. Cryptanalysis and security improvements of two–factor user authentication in wireless sensor networks. Sensors 2010; 10(3): 2450-59.

[8]. Vaidya B, Makrakis D, Mouftah HT. Improved two-factor user authentication in wireless sensor networks. Proceedings on the IEEE 6th International Conference on Wireless and Mobile Computing, Networking and Communications; 2010 Oct. 11-13; ON, Canada. IEEE 2010.

[9]. Deebak BD. Secure and efficient mutual adaptive user authentication scheme for heterogeneous wireless sensor networks using multimedia client–server systems. Wireless Personal Communications 2016; 87(3): 1013-35.

[10]. Das AK, Sharma P, Chatterjee S, Sing JK. A dynamic password-based user authentication scheme for hierarchical wireless sensor networks. Journal of Network and Computer Applications 2012; 35(5): 1646-56.

[11]. Turkanovic M, Brumen B, Hölbl M. A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the internet of things notion. Ad Hoc Networks 2014; 20: 96-112.

[12]. Farash MS, Turkanovic M, Kumari S, Hölbl M. An efficient user authentication and key agreement scheme for heterogeneous wireless sensor net- work tailored for the internet of things environment. Ad Hoc Networks 2016; 36: 152-76.

[13]. Amin R, Biswas G. A secure light weight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks. Ad Hoc Networks 2016; 36: 58-80.

[14]. Srinivas J, Mukhopadhyay S, Mishra D. Secure and efficient user authentication scheme for multi-gateway wireless sensor networks. Ad Hoc Networks 2017; 54: 147-69.

[15]. Xu J, Liu W, Lang F, Zhang Y, Wang C. Distance measurement model based on RSSI in WSN. Wireless Sensor Network 2010; 2: 606-11.

[16]. Lee S, Lee J, Sin H, Yoo S, Lee S, Lee J, Lee Y, Kim S. An energy-efficient distributed unequal clustering protocol for wireless sensor networks. World Academy of Science, International Journal of Electronics and Communication Engineering 2008; 8(12): 2715-9.

[17]. Li C, Ye M, Chen G, Wu J. An energy-efficient unequal clustering mechanism for wireless sensor networks. Proceedings on the IEEE International Conference on Mobile Adhoc and Sensor Systems Conference; 2005 Nov. 7-7; Washington DC, USA. IEEE 2005.

[18]. Kocher P, Jaffe J, Jun B. Differential power analysis. Lecture Notes in Computer Science 1999; 1666: 388-97.

[19]. Messerges TS, Dabbish EA, Sloan RH. Examining smart-card security under the threat of power analysis attacks. IEEE Transactions on Computers 2002; 51(5): 541-52.

[20]. Kerckhoffs A. La cryptographie militaire. Journal des sciences militaires 1883; 9: 161-91.

[21]. Burrows M, Abadi M, Needham R. A logic of authentication. ACM transactions on Computer Systems 1990; 8(1): 18-36.

[22]. Kim H. Freshness-Preserving Non-Interactive Hierarchical Key Agreement Protocol over WHMS. Sensors 2014; 14: 23742-57.

[23]. MIRACL, https://www.miracl.com, accessed at July 10 2018.