A Realistic Lightweight Anonymous Authentication Protocol for Securing Real-time Application Data Access in Wireless Sensor Networks

Prosanta Gope, Tzonelih Hwang

***Abstract* — User authentication in wireless sensor networks (WSNs) is a critical security issue due to their unattended and hostile deployment in the field. Since the sensor nodes are equipped with limited computing power, storage, and communication modules, authenticating remote users in such resource-constrained environment is a paramount security concern. Till now, impressive efforts have been made for designing authentication schemes with user anonymity by using only the lightweight cryptographic primitives such as symmetric key encryption/decryption, and hash functions. However, to the best of our knowledge none has succeeded so far. In this article, we take an initial step to shed light on the rationale underlying this prominent issue. In order to do that, here at first we demonstrate that the existing solutions for anonymous user authentication in WSN are impractical. Subsequently, we propose a realistic authentication protocol for WSN, which can ensure various imperative security properties like user anonymity, untraceability, forward/backward secrecy, perfect forward secrecy, etc.**

***Index Terms* — Mutual authentication, User anonymity, Wireless sensor network.**

I. Introduction

With the rapid development of the micro-electromechanical system and wireless network technologies, wireless sensor networks (WSNs) are becoming more and more popular in everyday life as they offer economically viable, real-time monitoring solutions. Wireless sensors can be quickly and easily deployed in hostile environments, and WSNs are now widely used in hostile environments and also in a variety of real-time applications, such as vehicular tracking, habitat monitoring, environment control, military surveillance, healthcare monitoring, wildlife monitoring, and traffic monitoring. Besides, according to [1], WSNs will become intelligent and integral part of daily lives. In many critical applications [2-3], external users are generally interested in accessing real-time information from the sensor nodes. In that case, to enable external users to access the real-time data directly from desired sensor nodes without involving the gateway node (or base station), it is of great concern that critical data is well protected from eavesdropping, malicious modification, unauthorized access and so on. Hence, the user authentication constitutes an essential security mechanism for user to be authenticated first by the sensor nodes before being granted the right to access data. Fig. 1 depicts a way for real-time data access in WSNs [4]. As described in [5-6], this architecture can be useful in various applications such as in a health-care environment, in that case, continuous monitoring of a patient's temperature, heartbeats, etc. will be collected by sensors. Then a legitimate doctor can acquire these data directly from the sensor nodes. Therefore, this approach can bring lots of benefits to both the patients and doctors in several health-care applications such as bio-sensor based health-care system.

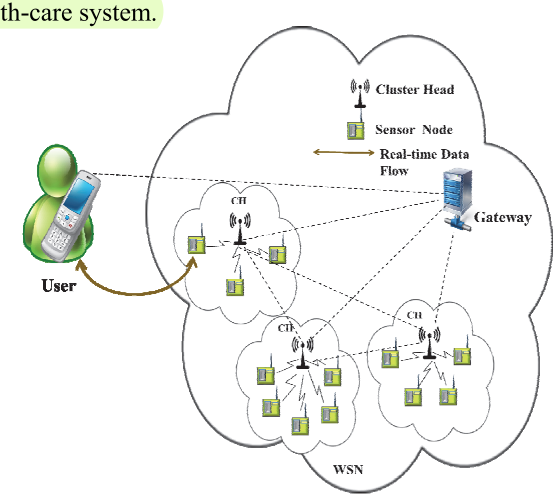


Fig. 1. Real-time Data Access in WSNs

A. *Related* Work

Over the past few years, some interesting anonymous user authentication protocols for wireless sensor networks have been proposed [7-21]. Particularly, in 2007, Wong et al. [7] firstly proposed a hash-based user authentication scheme, which is less complex, lightweight, and dynamic. But some researchers found that it is vulnerable to stolen-verifier, replay, and forgery attacks. Subsequently, in 2009, Das [8] suggested a scheme, which has become a frequently cited literature in this area of password based authentication. Unfortunately, this scheme has some serious security flaws and does not provide mutual authentication and key exchange. A series of schemes [9-11] are subsequently put forward to enhance its security. For example, He et al. [9] proposed a similar protocol as Das.

Manuscript received October 18, 2015; revised March 7, 2016 and April 24, 2016; accepted May 27, 2016.

Author's addresses: Prosanta Gope, iTrust, Centre for Research in Cyber Security of Singapore University of Technology and Design, Singapore

Tzonelih Hwang, Department of Computer Science and Information Engineering, National Cheng Kung University, NO. 1, Ta-Hsueh Rd., Tainan, Taiwan, R.O.C.

Email: prosanta.nitdgp@gmail.com, hwangtl@ismail.csie.ncku.edu.tw

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIE.2016.2585081, IEEE Transactions on Industrial Electronics

Although this scheme enhances the password security but it did not essentially make up for the security flaws. In 2011, Fan et al. [11] noticed that previous two-factor authentication schemes [8-10] for real-time data access in WSNs have various defeats overlooked and they proposed a new privacy preserving scheme, which involves only lightweight cryptographic operations, such as hash functions and exclusive-OR. Hence, it is well-suited to the resource-limited sensor networks and exhibits great potential for practical use. Although their scheme has many merits over existing scheme, but the scheme cannot ensure user anonymity. Besides, it is vulnerable to various practical attacks [11-12]. Meanwhile, Kumar et al. [13] also noticed that previous schemes either ere subject to security defeats or short of essential features like mutual authentication and user anonymity. Hence, they proposed a privacy-preserving two-factor authentication framework for WSNs that can withstand all known attacks and also supports several desired features. Soon after this framework was proposed, Jiang et al. [14] pointed out that the scheme proposed by Kumar et al. cannot resist off-line password guessing attack and also fails to ensure user untraceability and they proposed a scheme to overcome these two drawbacks. However, according to [12], the protocol is vulnerable to de-synchronization attack, where a dishonest sensor node can render the victim user's smartcards completely un-useable by simply altering the last message flow without being detected. In the meantime, Chen et al. [15] proposed a lightweight mutual authentication protocol, but the protocol is also vulnerable to many attacks such as replay attacks, forgery attacks, and bypassing attacks [16]. In 2013, Xue et al. proposed a lightweight temporal-credential-based mutual authentication and key establishment protocol [16]. Unfortunately, Wang et al. [12] demonstrated that the scheme cannot achieve its essential goal of user anonymity and untraceability. Recently, some other interesting authentication protocols for wireless sensor networks have been proposed [17-21]. However, after thorough inspection, we found that the protocols presented in [17-21], suffered from several issues (shown in Section I-B).

B. Problem Statement and Motivation

User anonymity is among the imperative properties of two-factor authentication schemes for WSNs. A more satisfactory property of user anonymity is user un-traceability, which guarantees that the adversary can neither discern who the user is nor can tell apart whether two conversations originate from the same (unknown) user. That is, a scheme accomplishing this advance property can resist the adversary from linking multiple instances of communication generated by the same user and also from tracing a current location, moving history, etc. As a consequence, most schemes that attempts to preserve user privacy aim at fulfilling this stronger notation of user anonymity. Now, to accomplish this property, in the existing state of the art two-factor protocols, such as [1-8], and [10-16] a user needs to encode his/her the original id into a dynamic identity. In that case, when a user requests to a gateway, then none of the requested parameter can help the gateway to realize the identity of the user, where the gateway needs to do more exercise (possibly an exhaustive search operation) or needs to have a backend channel in order to figure out exactly who is the user, which is not relevant at all. In order to justify our point more clearly, here we consider an example, in case of the protocol like [16], where we see that the user sends a request message $\{DID, C_i, PKS, TS_4, TE_i, P_i\}$ to the gateway for authentication, where $DID_i = ID_i \oplus h(TC_i \parallel TS_4), C_i = h(h(ID_i \parallel TS_4) \oplus TC_i),$ $PKS_i = K_i \oplus h(TC_i \parallel TS_4 \parallel "000"),$ $P_i = h(ID_i \parallel TE_i).$ Here, the ID_i represents the user identity, whereas the parameters h and $\oplus$ denote the one-way hash function and Exclusive-OR operations, respectively. On the other hand, the parameters TC_i , TE_i , and TS_4 denote the temporal security credential issued by the gateway to user (which is unique for each user), the expiration time of the temporal security credential TC_i , and timestamp value, respectively. Now, none of the parameter can help the gateway to comprehend that exactly who is the user. In order to know the user's identity either the gateway needs to perform an exhaustive search operation by targeting one of the relation among $DID_i = ID_i \oplus h(TC_i \parallel TS_4),$ or $C_i = h(h(ID_i \parallel TS_4) \oplus TC_i)$, or both the user and the gateway may require to manage a backend channel for that. Unfortunately, the similar problems can also be found in other existing state of the art two-factor anonymous authentication protocols [7-15], and [17-19]. Although some protocols such as [20-21] tried to resolve this issue by using the concept of temporary identity *TID*. In this approach, at the end of the authentication process, both the user and the gateway need to update the *TID*. However, our research found that the protocols presented in [20-21] are vulnerable to DoS attack [24-25]. In that case, when the last response message sent from the gateway is disrupted by the adversary, then, the user cannot update his/her temporary identity and which will cause loss of synchronization between the user and gateway.

Besides, according to a recent study [12], there is no two-factor authentication protocol in WSNs that achieves the precious property of user anonymity. Furthermore, even though there are some schemes [10], [16], and [17-21] which support session key establishment between the user and the sensor node, for the secure real-time data transmission. However, none of the scheme can ensure perfect forward secrecy, which is the basic and important security property for a session key based authentication protocol. In that case, when the secret keying material is compromised by the adversary, then the adversary can learn any previous session key, which is a serious threat against the real-time data security. These are the issues which have been a great inspiration for us to propose a realistic authentication protocol, which can ensure various imperative properties, such as user anonymity, perfect forward secrecy, resiliency of stolen smart card attacks etc. Most importantly, in our proposed scheme a gateway can easily comprehend the user without performing any exhaustive search operation or maintaining any backend channel.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIE.2016.2585081, IEEE Transactions on Industrial Electronics

C. Threat Model

We use the Dovel-Yao threat model [22], in which two communicates interact over the insure channel. Under this model, an adversary can intercept the transmitted messages over insecure public channel. Besides, the adversary has the ability to alter, delete the content of the messages communicated over the insecure public channel. In addition to that, the adversary can acquire all the sensitive information stored in the smart-card by using side-channel attacks [23]. On the other hand, sensors are not equipped with tamper-resistant hardware. As a consequence, if an adversary physically captures a sensor node then the adversary can know all the security credentials stored in the sensor's node memory.

Paper organization. Therefore, the reminder of this article is organized as follows. In Section II, we present our realistic anonymous user authentication protocol for WSN environment. Security analysis of the proposed scheme is given in Section III. A relevant discussion based on the performance of the proposed scheme is given in Section IV. Finally, concluding remark is given in Section VI. The abbreviations and cryptographic functions used in this article are defined in the TABLE I.

TABLE I
NOTATIONS AND CRYPTOGRAPHIC FUNCTIONS

Symbol	Definition
U	User
GW	Gateway
Sn	Sensor
ID_U	Identity of the user
AID_U	One-time-alias identity of the user
SID	Shadow identity of user
ID_G	Secrect Identity of the gateway
ω	Secret key of the gateway
Sn_{id}	Identity of the sensor node
PSW_U	Password of the user
N_u	Random number generated by the user
SK	Session key between Sn and U
K_{ug}	Shared key between U and GW
K_{em}	Shared emergency key between U and GW
K_{gs}	Secret Key shared between the GW and Sn
Ts_{ug}	Transaction sequence number (maintain both U and GW)
$h(.)$	One-way hash function
$\oplus$	Exclusive-OR operation
$\|\|$	Concatenation operation

II. PROPOSED SCHEME

In this section, we will describe our proposed realistic anonymous authentication scheme in detail. Our proposed scheme consists of four phases. **In Phase I**, a gateway (GW) issues a smart card to an intended user through secure channel, this phase is called registration phase. The next phase of our proposed scheme (**Phase II**) is the anonymous authentication and key exchange phase, where both the user (U) and a sensor node (Sn) can establish a session key between them. So that, they can ensure secure real-time data transmission. **In Phase III**, U can renew his/her password. Therefore, this phase is denoted as the password renewal phase. **In Phase IV**, a gateway (GW) can dynamically deploy some fresh sensor nodes. Therefore, this phase is denoted as dynamic node addition phase. So, the design objectives of our proposed scheme are as follows:

- To achieve mutual authentication by preserving the feature of user anonymity;
- To achieve untraceability;
- To achieve perfect forward secrecy (PFS);
- To defeat forgery attack, known session key attack along with the forward/backward secrecy support;
- To reduce computation and communication cost;
- To offer a realistic and expeditious solution;

*A. **Phase I:** Registration Phase*

Our registration phase is the underlying foundation for the construction of the proposed authentication scheme. It consists of the following steps.

Step R1: A new user (U) submits his/her identity ID_U to the gateway (GW) through a secure channel.

Step R2: After receiving the request from user, the gateway generates a random number n_g of 128-bit and then computes secret key $K_{ug} = h\left(ID_U \| n_g\right) \oplus ID_G$. Subsequently, GW also generates a set of unlinkable shadow-IDs $SID = \{sid_1, sid_2, \ldots\}$, where for each $sid_j \in SID$, GW computes $sid_j = h(ID_U \| r_j \| K_{ug})$. Then GW also generates a set of emergency keys $K_{em} = \{k_{em_1}, k_{em_2}, \ldots.\}$, each corresponds to a particular $sid_j \in SID$, where for each $k_{em_j} \in K_{em}$, GW computes $k_{em_j} = h(ID_U \| sid_j \| r_j^{'})$. Here, the parameters r_j, $r_j^{'}$ denote the random numbers of 128-bit used for deriving the shadow-ID sid_j and the corresponding emergency key k_{em_j} respectively. Hereafter, the gateway generates a transaction sequence number Ts_{ug}, which is basically a sequence number of 64-bit. This sequence number is computed based on the number of requests (m) handled by the GW, including the present request of the current user. Where, for each request of any user, the value of the request parameter m will be incremented by one and then the system (GW) sets $Ts_{ug} = m$ and subsequently sends the encoded Ts_{ug} to the user by keeping a copy in its database, in which GW can see the most recent Ts_{ug} for each user. The concept of sequence number is mainly used to speed up the authentication process as well as to prevent any replay attempt from any adversary, where by seeing the Ts_{ug} and comparing it with the stored value of its database, the gateway can comprehend that exactly who is the user and based on

Ts_{ug} , the gateway can even decide that whether the user request (during authentication) is valid or not. Precisely, during the execution of the anonymous authentication and key exchange phase (Phase II), if the Ts_{ug} provided by the user does not match with the stored value of the GW's database. Then, the GW will immediately terminate the connection. In that case, user will be asked to use his/her one of the unused $sid_j \in SID$ and its corresponding emergency key $k_{em_j} \in K_{em}$. Once a pair of shadow-ID sid_j and the emergency key k_{em_j} is used up, then the pair of $\left(sid_j, k_{em_j}\right)$ must be deleted from the list of (SID and K_{em}) by both the U and GW.

Step R3: Now, the GW personalizes a smart card with $\left\{K_{ug},\left(SID,K_{em}\right),Ts_{ug},h(.)\right\}$ and issues it to U through the secure channel; and then the GW uses its secret id and the secret key ω (stored in secure ROM-BIOS of the GW) to encode $\left\{ID_U,K_{ug}, \text{and} K_{em}\right\}$ i.e. $ID_U^{\#}=ID_U \oplus h\left(ID_G \| \omega \| Ts_{ug}\right)$, $K_{ug}^{\#}=K_{ug} \oplus h\left(ID_G \| ID_U \| \omega\right)$, $K_{em}^{\#}=K_{em} \oplus h\left(ID_G \| ID_U \| \omega\right)$, and then stores a copy of $ID_U^{\#}, K_{ug}^{\#},\left(SID, K_{em}^{\#}\right)$, and Ts_{ug} in its own database for further communication.

Step R4: After receiving the smart card, U chooses a password PSW_U and then computes $K_{ug}^* = K_{ug} \oplus h\left(h\left(ID_U\right) \oplus h\left(PSW_U\right)\right)$, $f_U^* = h\left(h\left(K_{ug}\right) \oplus h\left(PSW_U\right) \oplus h\left(ID_U\right)\right)$, $SID^* = SID \oplus h\left(h\left(ID_U\right) \oplus h\left(PSW_U\right)\right)$ $K_{em}^* = K_{em} \oplus h\left(h\left(ID_U\right) \oplus h\left(PSW_U\right)\right)$. Finally, U replaces K_{ug} with K_{ug}^* , SID with SID^* , and K_{em} with K_{em}^* so that, the smart card contains, $\left\{K_{ug}^*, f_U^*,\left(SID^*, K_{em}^*\right), Ts_{ug}, h(.)\right\}$.

B. ***Phase II:*** *Anonymous Authentication and Key Exchange Phase*

This phase achieves goal of authentication among the user, gateway, and the sensor node. Besides, at the end of the execution of this phase, a session key is established between the user and the sensor node. This phase of the proposed scheme consists of the following steps:

Step 1 $\mathbf{M}_{\mathbf{A_1}}$: U → GW: $\left\{AID_U, N_x, Ts_{ug} \ (if \ req.), Sn_{id}, V_1\right\}$.

The User, who wants to acquire real-time data from a sensor node Sn_{id} , inserts his/her smart card terminal, and enters his/her identity ID_U and password PSW_U . The smart card computes $K_{ug} = K_{ug}^* \oplus h(h(ID_U) \oplus h(PSW_U))$, $f_U = h(h(K_{ug}) \oplus h(PSW_U) \oplus h(ID_U))$ and subsequently checks the condition whether $f_U = f_U^*$. If it holds then the smart card ensures that the user successfully passes the verification process. Otherwise, this phase terminates immediately.

Now, after the successful verification, the smart card generates a random number N_u and derives the one-time alias identity $AID_U = h\left(ID_U \| K_{ug} \| N_u \| Ts_{ug}\right)$, $N_x = K_{ug} \oplus N_u$ and $V_1 = h\left(AID_U \| K_{ug} \| N_x \| Sn_{id}\right)$. Finally, the user forms a request message $\mathrm{M}_{\mathrm{A}_1}$ and then sends it to gateway. Here, Ts_{ug} denotes the most recent transaction sequence number received from GW. Note that, in case of loss of synchronization, the user needs to choose one of the unused pair of $\left(sid_j^*, k_{em}^*\right)$ from $\left(SID^*, K_{em}^*\right)$ and then submits his/her identity ID_U and password PSW_U and computes $sid_j = sid_j^* \oplus h\left(ID_U \| PSW_U\right), k_{em_j} = k_{em_j}^* \oplus h\left(ID_U \| PSW_U\right)$. Subsequently, assigns the sid_j as AID_U i.e. $AID_U = sid_j$ and then assigns k_{em_j} as K_{ug} . In that case, the user need not to send any transaction sequence number Ts_{ug} in $\mathrm{M}_{\mathrm{A}_1}$.

Step 2 $\mathbf{M}_{\mathbf{A_2}}$: GW → Sn: $\left\{AID_U, SK', T, V_2\right\}$.

Upon receiving the request message from user, the gateway at first checks whether the transaction sequence number Ts_{ug} is valid or not. In that case, since the GW maintains the most recent transaction sequence number for each user. Hence, when the GW finds Ts_{ug} in its database then it selects that tuple and uses its secret id ID_G and the secret key ω to decode the original identity ID_U and K_{ug} of the user. In this way, the GW can figure out that exactly who is the user. If, the gateway cannot find the Ts_{ug} provided by user, in its database, then it immediately terminates the connection. Otherwise, the gateway checks that whether V_1 is equal to $h\left(AID_U \| K_{ug} \| N_x \| Sn_{id}\right)$ or not. If so, then the gateway at first derives $N_u = K_{ug} \oplus N_x$, and then verifies the one-time alias identity AID_U Otherwise, the gateway terminates the connection. Now, if the verification of AID_U is successful, then the gateway randomly generates a session key SK and a timestamp T. Subsequently, the gateway computes $SK' = h(K_{gs}) \oplus SK$, $V_2 = h(AID_U \| SK' \| T \| K_{gs})$ and forms a message $\mathrm{M}_{\mathrm{A}_2}$ and sends it to the sensor node Sn_{id} that the user wants to interact with.

Step 3 $\mathbf{M}_{\mathbf{A_3}}$: Sn → GW: $\left\{T', Sn_{id}, V_3\right\}$.

After receiving the message $\mathrm{M}_{\mathrm{A}_2}$, the sensor node at first checks the timestamp T and the message V_2. If both of them are valid then the sensor node derives $SK = h(K_{gs}) \oplus SK'$ and subsequently generates a timestamp T' and computes $V_3 = h\left(SK \| K_{gs} \| Sn_{id} \| T'\right)$. Hereafter, the sensor node Sn_{id} forms a response message $\mathrm{M}_{\mathrm{A}_3}$ and sends $\{T', Sn_{id}, V_3\}$ to the gateway. Finally, the sensor node derives $K_{gs_{new}} = h\left(K_{gs} \| Sn_{id}\right)$ and updates its shared secret key with $K_{gs} = K_{gs_{new}}$. In case of loss of synchronization between the sensor node Sn_{id} and GW, which can be comprehended if the sensor node repeatedly fails to verify the message V_2 in $\mathrm{M}_{\mathrm{A}_2}$.

Then, the sensor node Sn_{id} needs to ask the GW for the new secret shared key i.e. $K_{gs_{new}}$, which will be securely sent to the sensor node.

Step 4 $\mathbf{M}_{\mathbf{A_4}}$: GW $\rightarrow$ U: $\{SK'', V_4, Ts, x(if\ \ req.)\}$.

Now, after receiving the response message $\mathrm{M}_{\mathrm{A}_3}$, the gateway at first checks the timestamp T' and subsequently computes V_3 and checks whether it is equal to $h(SK \,||\, K_{gs} \,||\, Sn_{id} \,||\, T')$ or not. If so, then the gateway checks the latest value of the transaction sequence parameter m and increments it by $m \leftarrow m+1$ then stores $Ts_{ug_{new}} = m$ and computes $Ts = h(K_{ug} \,||\, ID_U \,||\, N_u) \oplus Ts_{ug_{new}}, V_4 = h(SK'' \,||\, N_u \,||\, Ts \,||\, K_{ug})$. Then, the gateway forms a response message $\mathrm{M}_{\mathrm{A}4}$ and sends it to the user. Finally, the gateway computes $K_{ug_{new}} = h(K_{ug} \,||\, ID_U \,||\, Ts_{ug_{new}})$, $K_{gs_{new}} = h(K_{gs} \,||\, Sn_{id})$ and updates its database with $K_{ug_{new}}$, $K_{gs_{new}}$ and $Ts_{ug_{new}}$.

Upon receiving the response message $\mathrm{M}_{\mathrm{A}_4}$, the user's smart card terminal computes $h(SK'' \,||\, N_u \,||\, Ts \,||\, K_{ug})$ and verifies whether it is equals to V_4 or not. If so, then the smart card derives $Ts_{ug_{new}} = h(K_{ug} \,||\, ID_U \,||\, N_u) \oplus Ts$, $K_{ug_{new}} = h(K_{ug} \,||\, ID_U \,||\, Ts_{ug_{new}})$ and stores $Ts_{ug} = Ts_{ug_{new}}, K_{ug} = K_{ug_{new}}$ for further communication. Otherwise, the user needs to initiate a new request with an unused pair of shadow identity and emergency key.

Note that, in case if the gateway cannot find any Ts_{ug} in $\mathrm{M}_{\mathrm{A}_1}$, then the system (GW), will validate the AID_U first, where the system will try to recognize the sid_j in AID_U by comparing with the entries in its database. If GW can find sid_j in its database, then it selects the tuple corresponds to sid_j and retrieve k_{em_j} associated with sid_j and subsequently validates the other request parameter such as V_1 and then proceeds for further computation and at the end, it randomly generates a new shared key i.e. $K_{ug_{new}}$ and encodes it by using the emergency key k_{em_j} (used on that particular transaction) and the real identity of the user ID_U, i.e. $x = K_{ug_{new}} \oplus h(ID_U \,||\, k_{em_j})$ and sends x with other response parameters in $\mathrm{M}_{\mathrm{A}_4}$. In that case, the response parameter V_4 will be computed in the following way, i.e. $V_4 = h(SK'' \,||\, N_u \,||\, Ts \,||\, x) \oplus k_{em_j}$. If the system cannot recognize the sid_j in AID_U then it immediately terminates the connection and requests the user to try with a valid unused pair of (sid_j, k_{em_j}). The details of the authentication and key establishment phase are also depicted in Fig. 2.

C. *Phase III: Password Renewal Phase*

In this scheme, a user can freely change his/her password on the smart card without any help of the gateway. When the user wants to renew a password, he/she needs to insert his identity ID_U, old password PSW_U and the new password PSW_U^* to the smart card. Thereafter, the smart card will retrieve $K_{ug} = K_{ug}^* \oplus h(h(ID_U) \oplus h(PSW_U)), SID = SID^* \oplus h(h(ID_U)) \oplus h(PSW_U)$, $K_{em} = K_{em}^* \oplus h(h(ID_U) \oplus h(PSW_U))$, and then derive $K_{ug}^{**} = K_{ug} \oplus h(h(ID_U) \,||\, h(PSW_U^*))$, $SID^{**} = SID^* \oplus h(h(ID_U) \,||\, h(PSW_U^*))$, $K_{em}^{**} = K_{em} \oplus h(h(ID_U) \,||\, h(PSW_U^*))$. Finally, the device will replace K_{ug}^* with K_{ug}^{**}, SID^* with SID^{**} and K_{em}^* with K_{em}^{**} and subsequently stores them for further communication.

D. *Phase IV: Dynamic Node Addition Phase*

Considering a new sensor node Sn_i^{new} is required to be deployed in an existing wireless sensor network. In this context, the gateway at first randomly generates a unique identity $Sn_{id_i}^{new}$ and a key $K_{gs_i}^{new}$ for Sn_i^{new}. Then, the gateway loads the $Sn_{id_i}^{new}$, $K_{gs_i}^{new}$ in the memory of Sn_i^{new} prior to its deployment. Hereafter, the gateway encodes $K_{gs_i}^{new}$ with its id and secret key i.e. $K_{gs_i}^{new*} = K_{gs_i}^{new} \oplus h(ID_G \,||\, \omega \,||\, Sn_{id_i}^{new})$ and stores both the $Sn_{id_i}^{new}$, $K_{gs_i}^{new*}$ in its own database for further communication and subsequently informs the user U_j so that, he/she can access the real-time data from the newly deployed sensor node.

III. SECURITY ANALYSIS

In this section, we will demonstrate that our proposed scheme holds several imperative security properties under the defined threat model, which are indeed essential for securing real-time application data access in wireless sensor networks.

A. *Accomplishment of the User Anonymity and Untraceability*

As we mentioned before that both the shadow identity with emergency key pair and one-time-alias identity with transaction sequence number can resolve the issues like user anonymity and untraceability. There is not direct relationship

between the aliases. Besides, it can also be noticed that during the execution of our authentication protocol none of the parameter in the request message $\mathrm{M}_{\mathrm{A}_1}$ is allowed to be sent twice. This approach of the proposed scheme is quite effective for privacy against eavesdropper (PAE) [26-27] to achieve along with the features of user anonymity and untraceability.

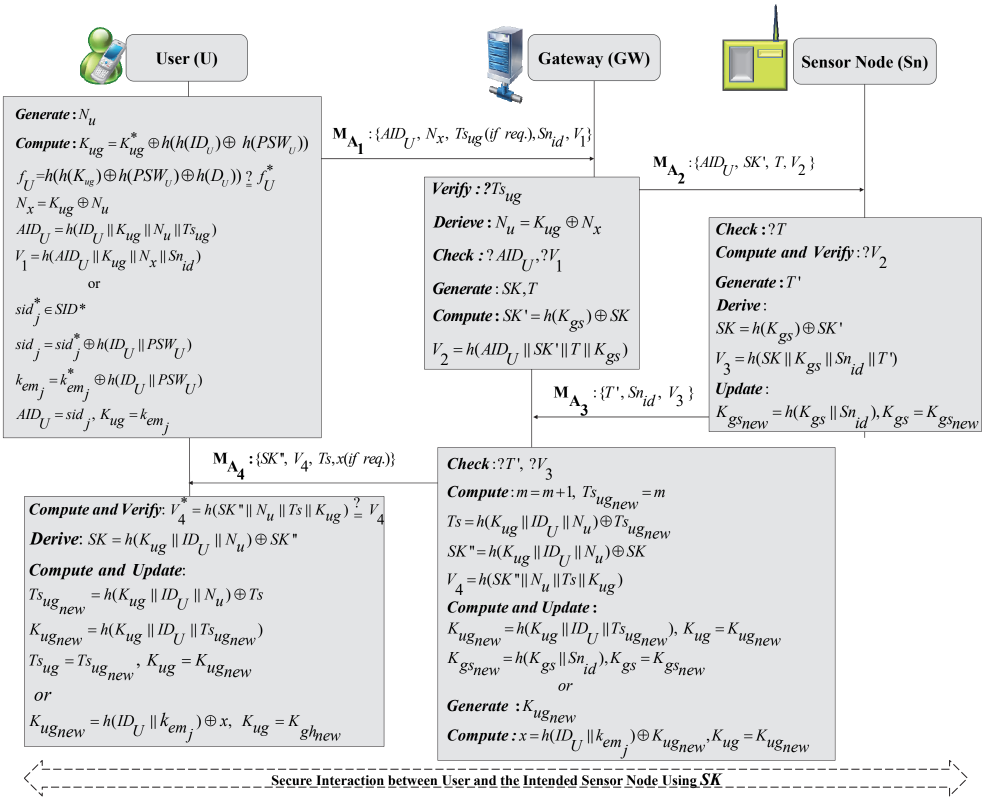

User (U)
Gateway (GW)
Sensor Node (Sn)
Generate : N_u
Compute : $K_{ug} = K_{ug}^* \oplus h(h(ID_U) \oplus h(PSW_U))$
$f_U = h(h(K_{ug}) \oplus h(PSW_U) \oplus h(D_U)) \overset{?}{=} f_U^*$
$N_x = K_{ug} \oplus N_u$
$AID_U = h(ID_U \| K_{ug} \| N_u \| Ts_{ug})$
$V_1 = h(AID_U \| K_{ug} \| N_x \| Sn_{id})$
or
$sid_j^* \in SID^*$
$sid_j = sid_j^* \oplus h(ID_U \| PSW_U)$
$k_{em_j} = k_{em_j}^* \oplus h(ID_U \| PSW_U)$
$AID_U = sid_j,\ K_{ug} = k_{em_j}$
$\mathbf{M_{A_1}} : \{AID_U,\ N_x,\ Ts_{ug}(if\ req.), Sn_{id},\ V_1\}$
Verify : $?Ts_{ug}$
Derieve : $N_u = K_{ug} \oplus N_x$
Check : $?\ AID_U, ?V_1$
Generate : SK, T
Compute : $SK' = h(K_{gs}) \oplus SK$
$V_2 = h(AID_U \| SK' \| T \| K_{gs})$
$\mathbf{M_{A_2}} : \{AID_U,\ SK',\ T, V_2\}$
Check : $?T$
Compute and Verify : $?V_2$
Generate : T'
Derive :
$SK = h(K_{gs}) \oplus SK'$
$V_3 = h(SK \| K_{gs} \| Sn_{id} \| T')$
Update :
$K_{gs_{new}} = h(K_{gs} \| Sn_{id}), K_{gs} = K_{gs_{new}}$
$\mathbf{M_{A_3}} : \{T', Sn_{id},\ V_3\}$
Check : $?T',\ ?V_3$
Compute : $m = m+1,\ Ts_{ug_{new}} = m$
$Ts = h(K_{ug} \| ID_U \| N_u) \oplus Ts_{ug_{new}}$
$SK'' = h(K_{ug} \| ID_U \| N_u) \oplus SK$
$V_4 = h(SK'' \| N_u \| Ts \| K_{ug})$
Compute and Update :
$K_{ug_{new}} = h(K_{ug} \| ID_U \| Ts_{ug_{new}}),\ K_{ug} = K_{ug_{new}}$
$K_{gs_{new}} = h(K_{gs} \| Sn_{id}), K_{gs} = K_{gs_{new}}$
or
Generate : $K_{ug_{new}}$
Compute : $x = h(ID_U \| k_{em_j}) \oplus K_{ug_{new}}, K_{ug} = K_{ug_{new}}$
$\mathbf{M_{A_4}} : \{SK'',\ V_4,\ Ts, x(if\ req.)\}$
Compute and Verify: $V_4^* = h(SK'' \| N_u \| Ts \| K_{ug}) \overset{?}{=} V_4$
Derive: $SK = h(K_{ug} \| ID_U \| N_u) \oplus SK''$
Compute and Update:
$Ts_{ug_{new}} = h(K_{ug} \| ID_U \| N_u) \oplus Ts$
$K_{ug_{new}} = h(K_{ug} \| ID_U \| Ts_{ug_{new}})$
$Ts_{ug} = Ts_{ug_{new}},\ K_{ug} = K_{ug_{new}}$
or
$K_{ug_{new}} = h(ID_U \| k_{em_j}) \oplus x,\ K_{ug} = K_{gh_{new}}$
Secure Interaction between User and the Intended Sensor Node Using SK

Fig. 2 Realistic Anonymous Authentication and Key Exchange Protocol

B. Accomplishment of the Perfect Forward Secrecy (PFS)

According to W. Diffe et al. [28], a protocol provides perfect forward secrecy (PFS) can resist an adversary from learning any previous session key, especially when the secret keying material is compromised by the adversary. Now, in our proposed scheme, after completion of each transaction both the user and gateway, and similarly, both the sensor node and the gateway needs to update their shared key K_{ug} with $K_{ug_{new}} = h(K_{ug} \| ID_U \| Ts_{ug_{new}})$, and K_{gs} with $K_{gs_{new}} = h(K_{gs} \| Sn_{id})$, respectively. Now, if the gateway is compromised by the adversary then he/she can manage $K_{ug_{new}}$, and $K_{gs_{new}}$. However, since the hash function is one-way, so that the adversary cannot acquire K_{ug} from $K_{ug_{new}}$ and K_{gs} from the $K_{gs_{new}}$. In this way, the protocol guarantees the security of any previous session key and eventually achieves perfect forward secrecy (PFS).

C. Resilience against Node Capture Attack

According to [18], the resilience against node captured attack of a user authentication in WSN can be measured by estimating the fraction of total secure communications that are compromised by a capture of n sensor nodes. In other words, we need to find out the effect of n sensor nodes being compromised on the rest of network. Conceive, Sn_i is a compromised sensor node and from that the adversary can know the secret key K_{gs_i} and even the session key SK_{ij}, established between the compromised node Sn_i and a legitimate user U_j. Now, in our proposed scheme, since sensor node shares a distinct secret key with the gateway and each sensor node establishes a distinct session key with a user, which is different from all other session keys in the network. As a consequence, the adversary can respond with the false data to a legal user only with from the captured sensor nodes. Other non-captured sensor nodes have the ability to communicate with the legitimate user by sending the real-time data with higher secrecy. Therefore, in our proposed scheme compromise of a sensor node does not lead to compromise other secure communication between the user and the non-compromised sensor nodes in the network.

D. Security Assurance in Case of Lost Smart Card

Usually, if the user's smart card is lost or an attacker steals the user's smart card then the attacker can easily get all the secret parameters stored [19] in it and thereafter can use it for illegal purposes. However, in our proposed scheme if the smart card is lost or stolen, the attacker cannot obtain the user's identity ID_U and password PSW_U. Besides, without knowing these parameters the attackers cannot compute

$K_{ug} = K^{*}_{ug} \oplus h(h(ID_U) \oplus h(PSW_U))$, $f_U = h(h(K_{ug}) \oplus h(PSW_U) \oplus h(D_U))$, where after proper verification only i.e. $f_U \neq f^{*}_U$, the smartcard can compute the following $AID_U = h(ID_U \| K_{ug} \| N_u \| Ts_{ug})$ or $sid_j = sid^{*}_j \oplus h(h(ID_U) \oplus h(PSW_U))$, $k_{em_j} = k^{*}_{em_j} \oplus h(h(ID_U) \oplus h(PSW_U))$, these are essential to convince the gateway (Details are shown in **Section II**) Furthermore, in our proposed scheme since each user has unique security credentials, so, if we consider that the security credentials of a user are compromised then that will not affect the security of the whole system. Only the victim user will be affected, which can also be resolved by informing the gateway to block services of the lost smart card and asking for a new card with fresh security credentials.

F. Key Compromise Impersonation Attack

Assume that an adversary has gotten the server's secret key ω and manage to obtain the encoded security parameters from the server's database. Based on this strong assumption, this type of attack is still infeasible. The adversary cannot compute $ID_U = ID^{\#}_U \oplus h(ID_G \| \omega \| Ts_{ug})$ as she has no knowledge about the secret identity of the gateway ID_G. It will also resist the adversary to compute $K_{ug} = K^{\#}_{ug} \oplus h(ID_G \| ID_U \| \omega)$. Therefore, the adversary cannot impersonate as a legal user.

IV. PERFORMANCE ANALYSIS AND COMPARISONS

The purpose of the proposed scheme is to resolve several security issues existing in the two-factor based authentication environment of the wireless sensor network and simultaneously to offer a realistic solution which can also guarantee reasonable computational overhead. In this section, we compare our proposed scheme with recently proposed scheme with user anonymity [10], [16], [18], [20] and [21] to manifest the advantages of our propose scheme. We also demonstrate that our propose scheme is well suitable for tiny powered sensor devices. Now, in order to analyze the performance of our proposed scheme especially on the security front, our proposed scheme has been compared with five state of the art protocols [10], [16], [18], [20] and [21] in WSN (shown in Table II). From Table II, it is clear that the proposed scheme can resist various security threats existing in two-factor based user authentication environment in WSN. In contrast, the protocols presented in [10], [16], [18], [20] and [21] are unable to ensure some of the imperative security properties. Even though protocols presented in [20], and [21] can ensure user anonymity property without performing any exhaustive search operation or without having any backend channel, however, these schemes are vulnerable to DoS attack (as shown **in Section I-B**). In addition to that, none of the existing scheme can ensure the perfect forward secrecy, which is highly imperative for any session-key based authentication protocol [26].

Now, in order to analyze the performance of the proposed scheme based on computational overhead, here we simulate the cryptographic operations used in the proposed scheme and the schemes presented in [10], [16], [18], [20] and [21] using the modular sensor board MSB-430 with the TI MSP430 micro controller and the temperature and relative humidity sensor Sensirion SHT11. Now, based on the simulation outcomes, Yeh et al.'s scheme causes significantly higher computational overhead in tiny powered sensor node as compared to others, where each modular exponential operation in ECC-160 algorithm takes 1.2 *Ws* energy, and 11.69 ms execution time. On the other hand, in the protocol presented in [18], where a sensor node need to perform symmetric key encryption/decryption that causes 0.72 *Ws* (128-bit AES-CBC) and 4.62 ms of execution time.

Now, in our proposed scheme and the protocols presented in [10], [16], [18], [20], and [21] a sensor node needs to perform hash operation. For the security of these protocol we consider that the hash function used in our proposed scheme and [10], [16], [18], [20], and [21] are secure one-way non-collision function like SHA-256 [29], and hence, we simulate SHA-256. In that case, each hash operation (SHA-256) takes 0.27 *Ws* and for that the execution time requires 1.06 ms. Table III shows the overall computational overhead of the proposed scheme and the scheme presented in [10], [16], [18], [20], and [21], where our proposed scheme requires only $19 * t_{Hash}$ operations. From that, it can be easily stated that, the computational overhead of the proposed scheme is significantly less the Yeh et al.'s scheme and even less from others. On the other hand, Table IV shows that the computational overhead of the sensor node during authentication process in our proposed scheme is much less than the others. Besides, if we consider the communication overhead of both the proposed scheme and the schemes presented in [10], [16], [18], [20], and [21], especially in terms of the length of the message send by each sensor node during the authentication process. Assuming that, the length of the identity of the sensor node is 128-bit, the length of the each hash value is also 128-bit, and the length of the timestamp value is 24-bit. Then, the length of the message sends by each sensor node in our proposed scheme and [18] is 35-byte. Whereas, the length of the message sends by each sensor node in [10], [16], [20], and [21] is 51-byte, which is significantly higher than both the proposed scheme and the scheme presented in [18], where for average transmission cost for each byte of data in sensor node is 1.2 *Ws*. Conclusively, performance of the proposed scheme in terms of the security, computational overhead, and communication overhead, is better than the other's. Besides, it should be noted that, in our proposed scheme using transaction sequence number a gateway can easily distinguish each user's request from others without performing any exhaustive search operation or any backend channel support. That makes our proposed scheme more realistic than the other existing state of the art protocols.

Table II. Performance Benchmarking Based on Security Properties

Scheme	SP1	SP2	SP3	SP4	SP5	SP6	SP7	SP8
Yeh et al. [10]	No	No	No	No	Yes	No	No	No
Xue et al.[16]	No	Yes	No	No	No	No	No	No
Das [18]	No	Yes	Yes	Yes	Yes	Yes	No	NA
Jiang et al. [20]	Yes	Yes	No	No	No	No	No	No
Das [21]	Yes	Yes	Yes	Yes	Yes	Yes	No	No
Ours	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

SP: Security Property; **NA:** Not Applicable**; SP1:** User Anonymity Without Performing Exhaustive Search Operation; **SP2:** Robustness Against Replay Attack; **SP3:** Privacy Against Eavesdroppers (PAE); **SP4:** Supporting of Dynamic Node Addition; **SP5:** Robustness Against Insider Attack; **SP6:** Robustness Against Lost Smart Card Problem; **SP7:** Perfect Forward Secrecy; **SP8:** Resilience Against DoS Attack;

TABLE III. Performance Benchmarking Based on Overall Computational Cost

Scheme	User	Gateway	Sensor Node
Yeh et al. [10]	$2t_{Exp} + t_{Hash}$	$4t_{Exp} + 4t_{Hash}$	$2t_{Exp} + 3t_{Hash}$
Xue et al.[16]	$7t_{Hash}$	$10t_{Hash}$	$5t_{Hash}$
Das [18]	$10t_{Hash}$	$t_{Sym} + 2t_{Hash}$	$t_{Sym} + 3t_{Hash}$
Jiang et al. [20]	$7t_{Hash}$	$10t_{Hash}$	$5t_{Hash}$
Das [21]	$11t_{Hash}$	$11t_{Hash}$	$6t_{Hash}$
Ours	$7t_{Hash}$	$9t_{Hash}$	$3t_{Hash}$

t_{Hash} : Execution time of a one-way hash function; t_{Exp} : Execution time of a modular exponential operation Using ECC;

t_{Sym} : Execution time of a symmetric key operation AES-CBC;

TABLE IV. Performance Benchmarking Based on Computational and Communicational Cost of the Sensor node

Scheme	Computational Cost	Execution Time	Communication Cost
Yeh et al. [10]	3.21 Ws	26.56 ms	51 byte
Xue et al.[16]	1.35 Ws	5.3 ms	51 byte
Das [18]	1.53 Ws	7.8 ms	35 byte
Jiang et al. [20]	1.35 Ws	5.3 ms	51 byte
Das [21]	1.62 Ws	6.36 ms	51 byte
Ours	0.81 Ws	3.18 ms	35 byte

t_{Hash} : Execution time of a one-way hash function ≈ 1.06 ms with 0.27 Ws (Energy Consumption);

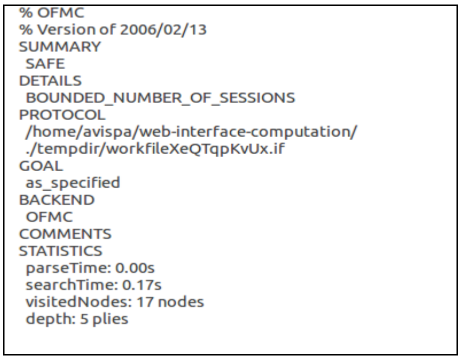
t_{Exp} : Execution time of a modular exponential operation Using ECC ≈ 11.69 ms with 1.2 Ws(Energy Consumption);

t_{Sym} : Execution time of a symmetric key operation Using AES-CBC ≈ 4.62 ms with 0.72 Ws(Energy Consumption);

N.B. Each byte of data transmission by the sennor node causes 1.2 Ws ;

A. Simulation for Formal Security Verification Using AVISPA Tool

In this subsection, we evaluate our scheme for the formal security verification using the AVISPA tool [21], [30]. It is used for automated validation of Internet security sensitive protocols and applications. It consists of four back-ends and abstraction-based methods that are integrated through the high level protocol specific language, known as HLPSL. The formal security verification and the results of our proposed scheme using OFMC backend are shown in Fig. 3. Besides, the implementation details of our proposed scheme HLPSL are provided in the supplementary material. Fig, 3 demonstrates that proposed scheme is secure and can be useful in WSN environment.



```
% OFMC
% Version of 2006/02/13
SUMMARY
 SAFE
DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
 /home/avispa/web-interface-computation/
 ./tempdir/workfileXeQTqpKvUx.if
GOAL
 as_specified
BACKEND
 OFMC
COMMENTS
STATISTICS
 parseTime: 0.00s
 searchTime: 0.17s
 visitedNodes: 17 nodes
 depth: 5 plies
```

Fig. 3. The result of the analysis using OFMC backend of our proposed scheme

V. CONCLUSION

In this article, we have proposed a realistic anonymous user authentication in wireless sensor networks. In comparison with existing schemes, our proposed scheme not only provides more security features and high security level, but at the same time also has low costs of communication and computation. Accordingly, our proposed scheme is suitable for the scenario that the legitimate user is allowed to access sensor data from any specific sensor node in the environment of resource-constrained wireless sensor network.

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. S. Subramaniam, E. Cayirci, "Survey on sensor network," *IEEE Communication Magazine*, vol.40, pp. 112-114, August 2002.

[2] O. Gnawali, K.-Y. Jang, J. Paek, M. Vieira, R. Govindan, B. Greenstein, A. Joki, D. Estrin, E. Kohler, "The tenet architecture for tiered sensor networks," *in: Proc. SenSys* 2006, ACM, pp. 153–166, October 2006.

[3] D. Yang, S. Misra, X. Fang, G. Xue, J. Zhang, "Two-tiered constrained relay node placement in wireless sensor networks: computational complexity and efficient approximations," *IEEE Trans. Mobile Computing*. vol. 11 no. 8 pp. 1399–1411, August 2012.

[4] P. Gope, T. Hwang, "BSN-Care: A Secure IoT-based Modern Healthcare System Using Body Sensor Network," *IEEE Sensors Journal*, Vol. 16 (5), pp. 1368 – 1376, March 2016.

[5] T. Nguyen, A. Al-Saffar, and E-N Huh, "A dynamic ID-based authentication scheme," Proceedings of the Sixth International Conference on Networked Computing and Advanced Information Management (NCM), pp. 248-253, August 2010.

[6] P. Gope, T. Hwang, "Untraceable Sensor Movement in Distributed IoT Infrastructure," *IEEE Sensors Journal*, vol. 15 (9), pp. 5340 – 5348, June 2015.

[7] K. H. M. Wong, Z. Yuan, C. Jiannong, and W. Sheng Wei, "A dynamic user authentication scheme for wireless sensor networks," *in Proceedings of the IEEE International Conference on Sensor Networks*, Ubiquitous, and Trustworthy Computing, pp. 244-251, Taiwan, June, 2006.

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

[9] D. He, Y. Gao, S. Chan, C. Chen, J. Bu. "An enhanced two-factor user authentication scheme in wireless sensor networks," *AdHoc & Sensor Wireless Networks* vol. 10 no. 4, February 2010.

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

[11] R. Fan, D. He, X. Pan, L. Ping, "An efficient and dos-resistant user authentication scheme for two-tiered wireless sensor networks," *Journal of Zhejinag Univ*.-Science vol. 12 no.7 pp. 550–560, May 2011.

[12] D. Wang, P. Wang, "On the anonymity of two-factor authentication schemes for wireless sensor networks: attacks, principle, and solutions," *Computer Networks,* vol. 73, pp. 41-57, November, 2014.

[13] P. Kumar, A.J. Choudhury, M. Sain, S.M. Lee, H.J. Lee, "RUASN: a robust user authentication framework for wireless sensor networks," *Sensors,* vol. 11 no. 5, pp.5020–5046, May 2011.

[14] Q. Jiang, Z. Ma, J.F. Ma, G. Li, "Security enhancement of robust user authentication framework for wireless sensor networks," *China Communication*, vol. 9, no. 10, pp. 103–111, August 2012.

[15] TH Chen, WK. Shih, "A robust authentication protocol for wireless sensor networks," *ETRI Journal*, vol. 32, no. 5, pp. 704-712, October 2010.

[16] K. Xue, C. Ma, P. Hong, R. Ding, "A temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks," *Journal of Network Computer Applications*, vol. 36 no. 1, pp. 316–323, January 2013.

[17] C. T. Li, C. Y. Weng, C. C. Lee, "An advanced temporal credential-based security scheme with mutual authentication and key agreement for wireless sensor networks," *Sensors*, vol. 13, pp. 9589–9603, July 2013.

[18] A. K. Das, "A Secure and Efficient User Anonymity-Preserving Three-Factor Authentication Protocol for Large-Scale Distributed Wireless Sensor Networks," *Wireless Personal Communications,* vol. 82 no. 3: pp. 1377-1404, January 2015.

[19] A. K. Das, "A secure and effective biometric-based user authentication scheme for wireless sensor networks using smart card and fuzzy extractor," *International Journal of Communication Systems* DOI: 10.1002/dac.2933, January 2015.

[20] Q. Jiang, J. Ma, X. Lu, Y. Tian, "An efficient two-factor user authentication scheme with unlinkability for wireless sensor networks," *Peer-to-Peer Networking and Applications*, June 2014.

[21] A. K. Das, A secure and robust temporal credential-based three-factor user authentication scheme for wireless sensor networks. *Peer-to-Peer networking and Applications* DOI: 10.1007/s12083-014-0324-9, June 2014.

[22] D. Dolev, A Yao, "On the security of public key protocols," *IEEE Transaction on Information Theory*, vol. 29 no. 2, pp. 198-208, March 1983.

[23] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis", in Proc. CRYPTO'99, LNCS 1666, pp. 388-397, Springer-Verlag, September 1999.

[24] C. H. Wang and C. Y. Lin, "An efficient delegation-based roaming payment, protocol against denial of service attacks," in Proc. 2011 *International Conference on Electronics, Communications and Control*, pp. 4136–4140, September 2011.

[25] P. Gope, T. Hwang, "A Realistic Lightweight Authentication Protocol Preserving Strong Anonymity for Securing RFID System," Computers & Security, Vol. 55, pp. 271–280, 2015.

[26] P. Gope, T. Hwang, "Lightweight and Energy Efficient Mutual Authentication and Key Agreement Scheme with User Anonymity for Secure Communication in Global Mobility Networks," *IEEE Systems Journal*, DOI: 10.1109/JSYST.2015.2416396, November 2015.

[27] P. Gope, T. Hwang, "An Efficient Mutual Authentication and Key Agreement Scheme Preserving Strong Anonymity of the Mobile User in Global Mobility Networks," *Journal of Network and Computer Applications*, Vol. 62, pp. 1-8, January 2016.

[28] W. Diffie, P. C. van Oorshot, and M. J. Wiener, "Authentication and Authenticated Key Exchanges," *Designs, Codes and Cryptography*, vol. 2, Kluwer Academic Publishers, pp. 107-125, June 1992.

[29] D. R. Stinson, "Universal Hashing and Authentication Codes," *Design Codes and Cryptography*, vol. 4 no. 4, pp. 369-380, July 1994.

[30] AVISPA Automated Validation of Internet Security Protocols and Applications. http://www.avispa-project.org/.

Prosanta Gope received his M. Tech degree in Computer Science and Engineering from National Institute of Technology (NIT), Durgapur, India, in 2009. In 2015, he has earned his PhD degree in Computer Science and Information Engineering, from National Cheng Kung University (NCKU), Tainan, Taiwan. Currently he is associated with the iTrust, Centre for Research in Cyber Security of Singapore University of Technology and Design. His research interests include lightweight authentication, authenticated encryption, access control system, security in mobile communication and cloud computing

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIE.2016.2585081, IEEE Transactions on Industrial Electronics

Tzonelih Hwang received the M.S. and Ph.D. degrees in Computer Science from the University of Southwestern Louisiana, USA, in 1988. He is currently a Distinguished Professor in the department of Computer Science and Information Engineering, National Cheng Kung University (NCKU), Tainan, Taiwan. His research interests include network and information security, access control systems, error control codes, security in mobile communication and quantum cryptography.