Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

RESEARCH ARTICLE

Security and Key Management in IoT Based Wireless Sensor Networks: An Authentication protocol using Symmetric Key

Anwar Ghani[1] | Khwaja Mansoor[2] | Shahid Mehmood[1] | Shehzad Ashraf Chaudhry*[3] | Arif Ur Rahman[4] | Malik Najmus Saqib[5]

[1]Department of Computer Science & Software Engineering, International Islamic University, Islamabad, Pakistan
[2]College of Signals, National University of Science & Technology, Islamabad, Pakistan
[3]Department of Computer Engineering, Faculty of Engineering and Architecture, , Istanbul Gelisim University , Istanbul, Turkey
[4]Department of Computer Science, Bahria University, Islamabad, Pakistan
[5]Department of Cyber Security, College of Computer Science and Engineering, University of Jeddah, Jeddah, Saudi Arabia

Correspondence
*Shehzad Ashraf Chaudhry, Department of Computer Engineering, Faculty of Engineering and Architecture,Istanbul Gelisim University Istanbul, Turkey. Email: ashraf.shehzad.ch@gmail.com

Summary

Wireless sensor networks consist of hundreds of miniature sensor nodes to sense various events in the surrounding environment and report back to the base station. Sensor networks are at the base of IoT and smart computing applications where a function is performed as a result of sensed event or information. However, in resource-limited Wireless Sensor Network authenticating a remote user is a vital security concern. Recently researchers put forth various authentication protocols to address different security issues. Gope et al. presented a protocol claiming resistance against known attacks. A thorough analysis of their protocol shows that it is vulnerable to user traceability, stolen verifier, and DoS attacks. In this article, an enhanced symmetric key-based authentication protocol for IoT based WSN has been presented. The proposed protocol has the ability to counter user traceability, stolen verifier, and DoS attacks. Furthermore, the proposed protocol has been simulated and verified using Proverif and BAN logic. The proposed protocol has the same communication cost, as the baseline protocol, however, in computation cost it has 52.63% efficiency as compared to the baseline protocol.

KEYWORDS:
WSN, IoT, authentication protocol, key agreement, symmetric encryption

1 | INTRODUCTION

Sensor Networks (WSN) are application specific and consists of a large number of sensor nodes deployed in harsh environments to monitor critical events. Sensor nodes are deployed randomly to monitor an area of interest[1,2]. The features of WSNs such as; small size, wireless architecture, ease of deployment and ubiquitous nature makes them an attractive platform for various applications in health, education, business, military and sports[3,4,5,6]. With the advancement in wireless technologies, WSNs are getting more attention lately due to the ease of deployment and low cost. WSN is at the backbone of the Internet of Things(IoT) that provides connectivity among objects of daily use. It make it a crucial component of the modern smart systems where authentication is even more crucial. It links with the wireless network by using interface by the (RFID), sensors and two-dimensional codes on objects[7,8].

The widespread used and resource constrained nature makes WSN an attractive target for attackers and malicious users. An attacker can disrupt operation in emergency situations or interfere with patients' data in healthcare application which may threaten human life, or gain illegal access to some business for monitory benefits[9,10]. In such cases security of WSN becomes

crucial. However, securing WSN is a challenging task due to the resource-limited nodes. Developing a sophisticated and complex security protocol for WSNs is not feasible. Such protocols may drain the power source of sensors faster, leading to energy wastage and shorter network life.

Sensor nodes are gathering real-time data directly from the environment, there are chances that the data may be exposed to unauthorized use by malicious users. Therefore, it is essential to cope with this problem for the sake of protection and unauthorized use of sensitive data. To achieve pool proof security, many techniques have been proposed[4,5,7], however, a realistic and efficient technique that can provide a user with perfect security is still a challenge. Recently, Gope et al. proposed an authentication protocol for IoT based WSN using symmetric key primitives. They claimed the protocol to be secure against known attacks. However, due to static identity and storage of verifier table, their scheme is insecure against traceability, DoS, stolen verifier attacks.

This article presents an enhanced symmetric key-based authentication protocol for IoT based WSN with the ability to counter all known attacks in addition to user traceability, stolen verifier, and DoS attacks. The proposed protocol alternated the use of static identity with pseudo dynamic identity; moreover, it has removed the verifier table to avoid stolen verifier attack. The proposed protocol has been tested for possible security lapses and key leakage using simulation software ProVerif and using BAN logic. The general contributions of this article are:

- Cryptanalyzed the Gope et al.[11] protocol for possible security loopholes and weaknesses and found that it is vulnerable to user traceability, stolen verifier, and DoS attacks.
- Designed an enhanced authentication protocol that resists all known attacks including those found during the cryptanalysis of Gope et al.[11] protocol.
- Analyzed the proposed protocol both formally and informally for any security loopholes and lapses using automated tool and BAN logic.
- Analyzed the proposed protocol for computation overhead and compared it with the existing state-of-the-art protocols to determined its computational efficiency.
- Analyzed how heavy the proposed protocol is in terms of communication? How much data is exchanged during one transaction of the proposed protocol? And comparatively analyzed it with the existing protocols.

The remainder of this article has been organized in seven sections, where section 2 presents a brief description of the related literature, section 3 presents a detailed review of the Gope et al.[11] proposal whereas section 4 details its cryptanalysis. Section 5 contains a detailed presentation about the proposed protocol, section 6 presents the security analysis of the proposed protocol. Security and performance analysis of the proposed protocol has been presented in section 7 whereas section 8 concludes the article.

2 | RELATED WORK

The role and importance of the security and authentication is growing every day with the growth of technology specially with the emergence of IoT[12,13,14]. However, modern technology faces serious issues due to problems in security and authentication. To solve the authentication issues in WSNs, several user authentication protocols have been proposed[15,16,17,18,19,20,21,22,23,24,25,26,27,28,29]. In 2007, Wong et al.[15] presented a hash-based user authentication protocol having low complexity, lightweight, and dynamic. However, it has been found that the protocol has a weak defense against stolen-verifier, replay, and forgery attacks. A password-based authentication protocol was presented by Das[16] in 2009, however, it lacks mutual authentication in the key exchange. To improve security and provide anonymity, He et al.[17], presented a related protocol as Das et al., improving password security, but failed to cope with security flaws.

In 2011, Fan et al.[19] observed that two-factor authentication schemes[16,17,18] for real-time data access in WSNs have many defects and presented a new privacy-preserving scheme based on lightweight cryptographic operations, like hash and exclusive OR. Due to lightweight nature, the protocol is suitable for WSN, but unfortunately, it does not provide user anonymity and other security requirements[19,20]. Meanwhile, Kumar et al.[15] presented a two-factor authentication protocol to preserve privacy in WSNs. This structure has capabilities to cater all known attacks and deficiencies. Later, Jiang et al.[19] observed that[21] protocol

is insecure against offline password guessing attack and suffers from user traceability. Therefore, they proposed a new protocol to address the two drawbacks observed in Kumar et al.[15].

Similarly, the protocol in Wang and Wang[30] is exposed to de-synchronization attack, where a compromised sensor node can judge the suffered user smartcard. The smartcard is completely ineffective by simply changing the flow of the previous message without any detection. Chen and Shih[31] proposed their lightweight mutual authentication protocol, however, like previous protocol, their proposal also has deficiencies to address replay attack, forgery attack, and bypassing attack[24]. In 2013, Xue et al.[24] proposed a lightweight temporal-credential-based mutual authentication and key establishment protocol.

Protocols presented in [28,29] introduced the concept of TID, but later in[32,33] it was observed that the protocols presented in[27,29] are vulnerable against DoS attack. The last response message sent by gateway if intercepted by an adversary then the user cannot update his/her TID. In this case, synchronization between user and gateway is lost. Gope et al.[11] later proposed a new lightweight symmetric key-based authentication protocol for WSN addressing issues like user-anonymity, perfect forward/backward secrecy and stolen smart card attacks. However, the scheme is insecure against User Traceability, Stolen Verifier and DoS attacks. The next section presents review of Gope et al.[11] protocol that from now on will be referred to as the baseline protocol.

3 | REVIEW OF BASELINE PROTOCOL

Baseline protocol for Real-Time Data Access applications in WSN consists of three entities; User, Gateway Node, and Sensor Node. In this protocol, the gateway node always issues a smartcard to the requested user using a secure channel then a session key (Symmetric) has been exchanging between the sensor node and the user. They also proposed password renew, deployment and registration of a new node. The scheme has four phases: Registration, Anonymous Authentication, and Key Exchange Phase, Password Update, New Node Addition. The steps involved in Login and authentication phase are shown in Figure 1

3.1 | Password Update Phase

In the baseline protocol, a user can change his/her password on the smartcard without the intervention of gateway. Whenever a password change is needed, the user only insert his ID_{Ur}, previously used password PSW_{Ur} and a new password PSW^*_{Ur}. After that the smartcard retrieves $K_{UrG} = K^*_{UrG} \oplus h(h(ID_{Ur}) \oplus h(PSW_{Ur}))$, $SID = SID^* \oplus h(h(ID_{Ur})) \oplus h(PSW_{Ur})$, $K_{Em} = K^*_{Em} \oplus h(h(ID_{Ur})\|h(PSW^*_{Ur}))$, and then calculates $K^{**}_{Urg} = K_{Urg} \oplus h(h(ID_{Ur}))\|h(PSW^*_{Ur}$, $SID^{**} = SID^* \oplus h(h(ID_{Ur})\|h(PSW^*_{Ur}))$, $K^{**}_{Em} = K_{Em} \oplus h(h(ID_{Ur})\|h(PSW^*_{Ur}))$. At the end, the device replaces K^*_{Urg} with K^{**}_{Urg}, SID^* with SID^{**} and K_{Em} with K^{**}_{Em} and then stores them for future communication.

3.2 | New Node Addition Phase

In case of a new sensor node ${Sn_1}^{new}$ deployment in existing sensor network, the gateway randomly generates and stores a distinct identifier $Sn^{new}_{id_i}$ with key $K^{new}_{G_{Sn}}$ in the new node's memory before deployment. The identifier and key are loaded into the new node's memory before its deployment. After that the gateway encodes $K^{new}_{G_{Sn_i}}$ with its id and secret key.i.e. $K^{new*}_{GSn_i} = K^{new}_{GSn} \oplus h(ID_{GW}\|K_{GW}\|Sn^{new}_{id_i})$ and stores the values of $Sn^{new}_{id_i}$, $K^{new*}_{GSn_i}$ in its database for future use. Also informs the user U_i so that he/she can access the real time information from the new sensor node.

4 | CRYPTANALYSIS OF BASELINE PROTOCOL

The authors in the baseline protocol presented in[34] claimed that their scheme is secured against various attacks like user anonymity, perfect forward secrecy, and stolen smartcard. However, after a thorough analysis, it has been observed that this protocol has some flaws shown in the cryptanalysis of the baseline protocol in the following subsection.

4.1 | User Traceability

The baseline protocol is vulnerable against user traceability attack. The Message $M_{A_1} = \{A_{ID_U}, N_x, Ts_{UrG}\text{(if required)}, Sn_{id}, Vr_1\}$ is transmitted using a public channel, therefore, an Adversary may capture, alter or

$\mathcal{ID}_{Ur}$	$\mathcal{GW}_T$	$\mathcal{S}_n$
Generate: N_U Compute: $K_{Urg} = K^*_{Urg} \oplus h(h(ID_{Ur}) \oplus h(PSW_{Ur}))$ $f_{Ur} = h(h(K_{Urg}) \oplus h(PSW_{Ur}) \oplus h(ID_{Ur})) \stackrel{?}{=} f^*_{U_r}$ $N_x = K_{Urg} \oplus N_U$ $A_{ID_U} = h(ID_{Ur} \| K_{UrG} \| N_U \| Ts_{UrG})$ $Vr_1 = h(A_{ID_U} \| K_{UrG} \| N_x \| Sn_{id})$ OR $sid^*_j \in SID^*$ $sid_j = sid^*_j \oplus h(ID_{Ur} \| PSW_{Ur})$ $K_{Em_j} = K^*_{Em_j} \oplus h(ID_{Ur} \| PSW_{Ur})$ $A_{ID_U} = sid_j, K_{Urg} = K_{Em_j}$		
$\xrightarrow{M_{A_1}=\{A_{ID_U}, N_x, Ts_{UrG}, Sn_{id}, Vr_1\}}$		
	Verify: $?Ts_{UrG}$ Derive: $N_U = K_{UrG} \oplus N_x$ Check: $?A_{ID_U}, ?Vr_1$ Generate: SK, T Compute: $SK' = h(K_{GSn}) \oplus SK$ Compute:$Vr_2 = h(A_{ID_U} \| SK' \| T \| K_{GSn})$	
	$\xrightarrow{M_{A_2}=\{A_{ID_U}, SK', T, Vr_2\}}$	
		check:$?T$ Compute and verify:$?Vr_2$ Generate: T' Derive: $SK = h(K_{GSn}) \oplus SK'$ $Vr_3 = h(SK \| K_{GSn} \| Sn_{id} \| T')$ Update: $K_{GSn_{new}} = h(K_{GSn} \| Sn_{id})$ $K_{GSn} = K_{GSn_{new}}$
	$\xleftarrow{M_{A_3}=\{T', Sn_{id}, Vr_3\}}$	
	Check:$?T', ?Vr_3$ Compute:$m = m + 1, Ts_{UrG_{new}} = m$ $Ts = h(K_{UrG} \| ID_{Ur} \| N_U) \oplus SK$ $SK = h(K_{UrG} \| ID_{Ur} \| N_{Ur}) \oplus SK_x$ $Vr_4 = h(SK' \| N_U \| Ts \| K_{UrG})$ Compute and update: $K_{UrG_{new}} = h(K_{GSn} \| ID_{Ur} \| Ts_{UrG_{new}}), K_{UrG} = K_{UrG_{new}}$ $K_{GSn_{new}} = h(K_{GSn} \| Sn_{id}), K_{GSn} = K_{GSn_{new}}$ OR Generate:$K_{UrG_{new}}$ Compute:$x = h((ID_{Ur} \| K_{Em_j}) \oplus K_{UrG_{new}}), K_{UrG} = K_{UrG_{new}}$	
$\xleftarrow{M_{A_4}=\{SK^*, Vr_4, Ts, x\}}$		
Compute and Verify: $Vr^*_4 = h(SK^e \| N_U \| Ts \| K_{UrG}) \stackrel{?}{=} Vr_4$ $SK = h(K_{UrG} \| ID_{Ur} \| N_U) \oplus SK^*$ Compute and update: $Ts_{UrG_{new}} = h(K_{UrG} \| ID_{Ur} \| N_U) \oplus Ts$ $K_{UrG_{new}} = h(K_{UrG} \| ID_{Ur} \| Ts_{UrG_{new}})$ $T_{UrG} = Ts_{UrG_{new}}, K_{UrG} = K_{UrG_{new}}$ OR $K_{UrG_{new}} = h(ID_{Ur} \| K_{Em_j}) \oplus x, K_{UrG} = K_{UrG_{new}}$		

FIGURE 1 Gope-Hwang's Proposed Scheme

delete any information. In baseline protocol the transaction sequence number $Ts_{UrG} = m$ where is m is user number, in the message M_{A_1} is transmitted openly. So the attacker may trace the user through $Ts_{UrG} = m$. Hence the protocol is not protected against user traceability.

4.2 | Stolen Verifier Attack

Stolen Verifier Attack occurs when an adversary theft verification data from the server in the current or previous authentication sessions. As the verification data is not well encrypted, The attacker can compromise the authentication process using different credential kept in verifier table of the server. In the baseline scheme, The attacker successful impersonates as a legitimate user from the next authentication session. As there is no encryption applied on the Ts_{UrG} before saving it in the verifier table.

4.3 | Vulnerable to DoS attacks

The Password Update Phase of baseline scheme does not provide verification of the previous password. Although $K_{UrG} = K^*_{UrG} \oplus h(h(ID_{Ur}) \oplus h(PSW_{Ur}))$ is computed in Password Update phase, but it does not provide instant verification. The attacker can use the wrong password again and again in the Password Update phase that may overwhelm the server and may cause Denial of service (DoS) attack.

5 | PROPOSED PROTOCOL

The proposed protocol consists of three main entities; User, Gateway Node, and Sensor Node. Successful authentication process takes place in different phases that include: User Registration, Anonymous Authentication and Key Exchange Phase, Password Update Phase, and New Sensor Node Addition. The proposed protocol consists of three main entities: User, Gateway Node, and Sensor Node. Successful authentication process takes place in the following four phases.

5.1 | User Registration

User registration in the proposed protocol requires the execution of the following four steps in sequence. The steps are also shown in Figure 2

Step 1: New user requests an ID ID_{Ur} from the gateway node GWT using a secure channel.

Step 2: The Gateway generates random number N_G of 128 bit and computes $K_{UrG} = h(ID_{Ur} \| N_G) \oplus ID_{GW}$. The GW_T also computes A_{ID_U} using its secret key K_{GW} as $A_{ID_U = E_{K_{GW}}(ID_{Ur} \| r_u)}$ and stores $(K_{UrG}, ID_{Ur}, A_{ID_U})$ for future correspondence.

Step 3: In this step the gateway personalize the smart-card with $M = \{K_{UrG}, ID_{Ur}, A_{ID_U}\}$ and issues the smart-card to the intended user using secure communication channel.

Step 4: The user after receiving the smart-card, stores information sent by the gateway node $(K_{UrG}, ID_{Ur}, A_{ID_U})$.

5.2 | Anonymous Authentication and Key Exchange Phase

This phase establishes authentication between the intended user, gateway node and the requested sensor node. In this phase, encryption and decryption have been performed at the gateway node GWT. This phase has the following steps. The detail diagram has been presented in in Figure 3

Step 1: $M_{A_1} : U_r \rightarrow GWT : (A_{ID_U}, N_x, T_1, Sn_{id}, Vr_1)$. In case a user needs to access real-time information from a sensor node Sn_{id}, the user will insert his/her smart card in the terminal, provide identity ID_U, and password PSW_{Ur}. After that the smart card computes $N_x = K_{Urg} \oplus N_U$ and $Vr_1 = h(A_{ID_U} \| K_{UrG} \| N_x \| Sn_{id} \| T_1)$. At last the user generates a request message $M_{A_1} = \{A_{ID_U}, N_x, T_1, Sn_{id}, Vr_1\}$ and forward the message to the gateway node for authentication. The same is forwarded to the sensor node by gateway node GWT.

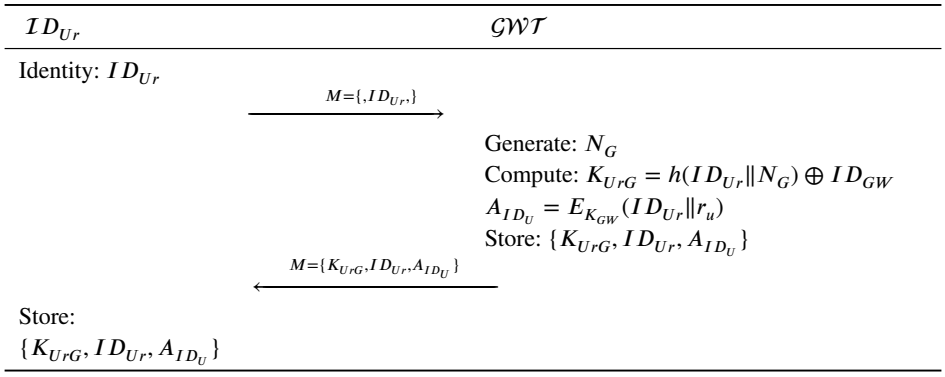

FIGURE 2 Proposed Registration Phase

Step 2: When the message $M_{A_2} : GWT \rightarrow S_n : (A_{ID_U,T_2,Vr_2})$ arrives at the gateway, first it is checked for freshness of transaction number using $T_2 - T_1 \leq \Delta T$. In this case the gateway node maintains the most recent transaction number for each user. The gateway GWT then computes $N_U = K_{UrG} \oplus N_x$ and A_{ID_U} by decrypting $A_{ID_U} = D_{K_{GW}}(ID_U\|r_u)$ and verifies both A_{ID_U} and Vr_1. After that the gateway node computes $Vr_2 = h(A_{ID_U}\|T_2\|K_{GSn})$ and creates a message M_{A_2} and passes it to a sensor node Sn_{id} that the intended user wants to communicate with.

Step 3: After receiving the message $M_{A_3} : Sn_{id} \rightarrow GWT : (T_3, Sn_{id}, Vr_3)$ the sensor node first checks the timestamp T by calculating $T_3 - T_2 \leq \Delta T$ and then checks and verifies the message Vr_2. If the verification is successful then the sensor node calculates and generates a new message Vr_3 as $Vr_3 = h(K_{GSn}\|Sn_{id}\|T_3)$.

Step 4: When the reply message $M_{A_4} : GWT \rightarrow U_r : (Z_G, Vr_4, T_4)$ arrives at the gateway, it checks T_3 and verifies Vr_3 if it is equal to $h(K_{GSn}\|Sn_{id}\|T_3)$. After that the gateway node GWT generates Vr_4 by computing $Vr_4 = h(N_U\|T_3\|K_{UrG})$ and then computes and updates $A_{ID_{U(new)}}$ by calculating $A_{ID_{U(new)}} = E_{K_{GW}}(ID_{Ur}\|r_{u(new)})$ and then calculates $Z_G = A_{ID_{U(new)}} \oplus K_{UrG}$. After this operation the gateway node creates a reply message M_{A_4} and forwards it to the user that contains $M_{A_4} = \{Z_G, Vr_4, T_4\}$.

Step 5: When the reply message received from gateway node GWT to user ID_{Ur} the user verifies the timestamp by calculating $T_5 - T_4 \leq \Delta T$ and then verifies $Vr_4^* = h(N_U\|T_3\|K_{UrG}) \stackrel{?}{=} Vr_4$. After that the user ID_{Ur} updates the values of $A_{ID_{U(new)}} = Z_G \oplus K_{UrG}$ and $A_{ID_U} = A_{ID_{U(new)}}$ in its smartcard for future correspondence.

5.3 | Password Update Phase

In the proposed protocol a user can change his/her password as well as the smart card without the intervention of gateway. Whenever a user needs to change the password, he/she only insert ID_{Ur}, previous password PSW_{Ur} and computes $\{K_{UrG} \stackrel{?}{=} K^*_{UrG} \oplus h(h(ID_{Ur}) \oplus h(PSW_{Ur}))\}$. After verifying the previous K_{UrG} the smartcard requests the user to enter a new password PSW^*_{Ur} to the smart card.

5.4 | New Sensor Node Addition Phase

In case of a new sensor node Sn_1^{new} deployment, the gateway randomly generates a distinct identifier $Sn_{id_i}^{new}$ with key K_{GSn}^{new} and stores it in the new node's memory that is loaded into Sn_1^{new} memory by the gateway node at the time of deployment. After that the gateway encodes $K_{G_{Sn_i}}^{new}$ with its ID_{GW} and secret key K_{GW} i.e. $K_{GSn_i}^{new*} = K_{GSn}^{new} \oplus h(ID_{GW}\|K_{GW}\|Sn_{id_i}^{new})$ and store the values of $Sn_{id_i}^{new}$, $K_{GSn_i}^{new*}$ in its database for future usage and later inform the user U_i so that he/she can access the real time information from the new sensor node.

6 | SECURITY ANALYSIS

The proposed protocol has been analyzed formally and informally. For formal analysis, two methods have been used. 1) A simulation tool proVerif has been used to verify the proposed protocol against a different known attack. 2) BAN logic [35] is used

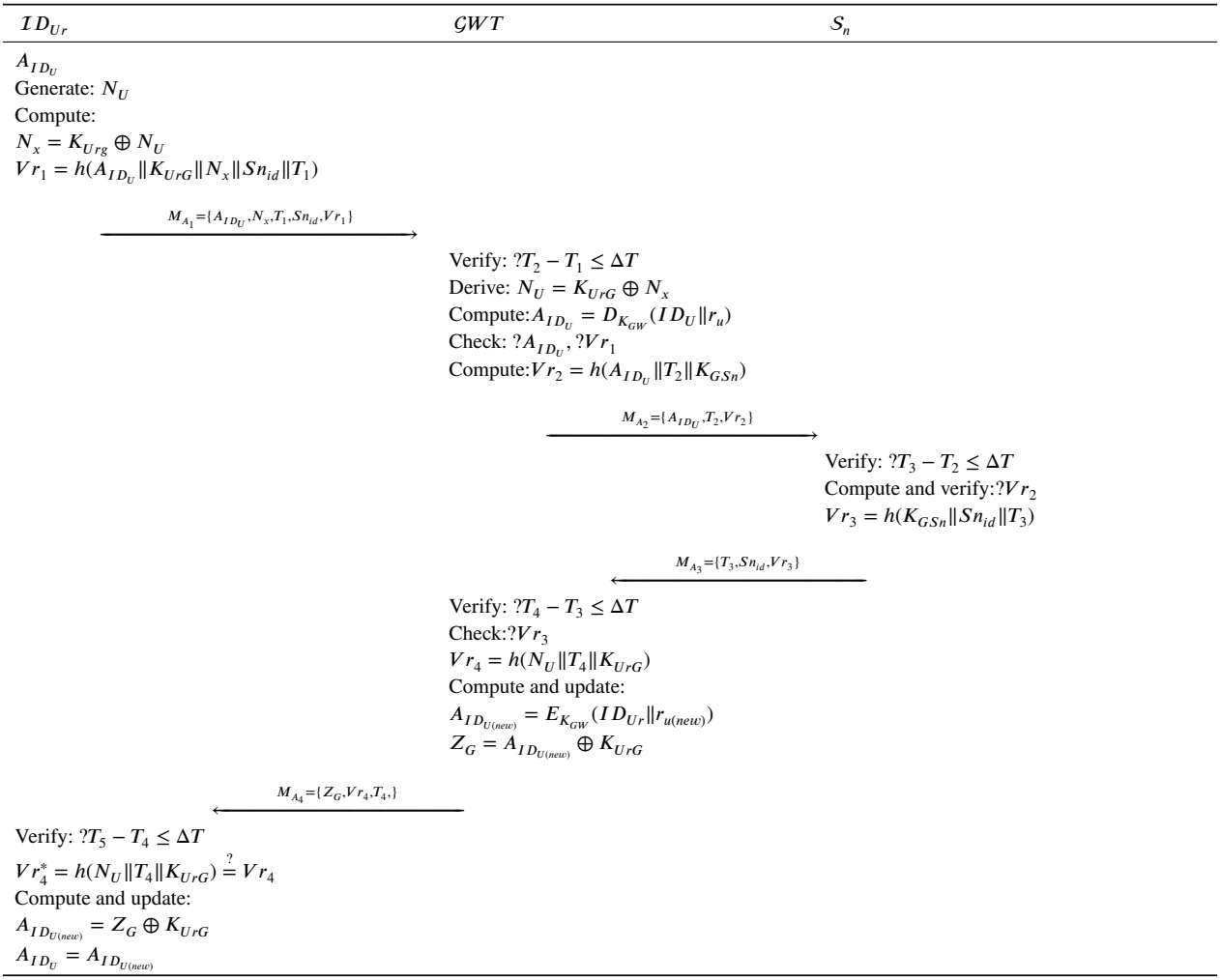

FIGURE 3 Proposed Scheme

to check the freshness and trustworthiness of the key exchanged between the communicating parties. It also checks the key exchange process is resistant to eavesdropping.

Informally, the proposed protocol has been a check against various attacks to find any possible security loopholes found in the baseline and other protocols in the literature.

6.1 | Security Analysis with ProVerif

Security analysis has been performed using the ProVerif simulation tool. ProVerif is a well-known simulation tool designed for verification of security algorithms against know attacks[36].

The proposed protocol uses two types of channels; private channel (ChSec:) and public channel (Chpub:). A secure channel is established between U_r and GWT in registration phase and the public channel between U_r, GWT and S_n for login and authentication phase. ID_{Ur} is the real identity of a user U_r, ID_{GW} is the identity of gateway GWt, and sid_j is the identity of sensor node S_n. All the three participants compute session key SK.

```
(* --------- Channels ----------*)
free ChSec:channel [private]. (*secure channel between UJ,GTW and SJ*)
free ChPub:channel    (*public channel between between UJ,GTW and SJ*)
(*------ Constants and Variables -----*)
free IDur :bitstring.
```

```
free IDs :bitstring.
free GWti:bitstring.
free Kurg : bitstring [private].
```

We have defined four constructors; h, Concat, XOR, and Mult for hash, concatenation, exclusive OR and multiplication.

```
(*========Constructors=======*)
fun h(bitstring):bitstring.
%fun Inverse(bitstring):bitstring.
fun Concat(bitstring,bitstring):bitstring.
fun XOR(bitstring,bitstring):bitstring.
fun Mult(bitstring,bitstring):bitstring.
```

Follows are the proverif code of registration and Login and authentication phases of the proposed scheme as per given in Figure 3

```
(*-----------Registration phase-----------------*)
let  pUser=
out(ChSec,(IDur));
in (ChSec,(xIDur:bitstring,xKurg:bitstring));
(*-----login-authentication---------------------*)
event start_User(IDur);
new Nur:bitstring;
new T1:bitstring;
new fur:bitstring;

let Kurg=XOR(Kurg,h(XOR(h(IDur)),XOR(h(PSWur))) in
if  fur=XOR(h(Kurg),h(XOR(h(IDur)),XOR(h(PSWur))) then
let Nx=XOR(Kurg,Nur) in
let AIDu=h(Concat(IDur,(Kurg,(Nur,T1)))) in
let Vr1=h(Concat(AIDu,(Kurg,(Nx,(Snid,T1)))))in
out(ChPub,(AIDu,Nx,Snid,T1,Vr1));
in (ChPub,(xVr4:bitstring,T3:bitstring,SK:bitstring));
let Vr4=h(Concat(SK,(Nur,(T3,Kurg)))) in
if Vr4=h(Concat(SK,(Nur,(T3,Kurg)))) then
let SKx=XOR(SK,h(Concat(Kurg,(IDur,Nur)))) in
let Kurgnew = h(Concat(Kurg,(IDur,T3))) in
event end_User(IDur)
else
0.
let pGWt=

%(*---------------login-authentication------------*)
event start_GWt(GWti);
new Nur:bitstring;
new Nx:bitstring;
new SKx:bitstring;
new T2:bitstring;
new Kgsn:bitstring;
new Kurg:bitstring;
new Dj:bitstring;
new Ts:bitstring;
new MA1:bitstring;
in (ChPub,(xxAIDu:bitstring,Nx:bitstring,Vr1:bitstring,Snid:bitstring,T1:bitstring));
```

```
let Nur=XOR(Kurg,Nx) in
let SK=XOR(h(Kgsn),SKx) in
let Vr2=h(Concat(AIDu, (SK,(Kgsn,T2)))) in
out(ChPub,(AIDu,SK,Vr2,T2));
in (ChPub,(Vr3:bitstring,Snid:bitstring,T3:bitstring));
let SK=XOR(h(Concat(Kurg,(IDur,T3))),SKx) in
let Vr4=h(Concat(SK, (Nur,(Kurg,T3)))) in
let Kurgnew=h(Concat(Kgsn, (IDur,T3))) in
let Kurgnew=h(Concat(Kgsn, (Snid,T3))) in
out(ChPub,(Vr4,SK,T3));
event end_GWt(GWti)
else
0.
let pSn=

event start_Sn(IDs);
in (ChPub,(SK:bitstring,xVr4:bitstring,T3:bitstring));
if Vr4= h(Concat(SK,( Nur,(T3,Kurg))))
let SKx = XOR(h(Concat(IDur,( Nur,Kurg))),SK) in
new Nx: bitstring;
let Kurgnew= h(Concat(Kurg,( IDur,T3))) in
event end_Sn(IDs)
else
0.
```

The parallel execution of all processes in the new scheme is an under:

```
process ((!pSn) | (!pGWt) | (!pUser) )
```

To verify authentication property the queries are as under:

```
(*-------Queries------*)
%free AIDTinew:bitstring .
query attacker(SK).
query id:bitstring; inj-event(end_User(IDur)) ==> inj-event(start_User(IDur)).
query id:bitstring; inj-event(end_GWt(GWti)) ==> inj-event(start_GWt(GWti)).
query id:bitstring; inj-event(end_Sn(IDs)) ==> inj-event(start_Sn(IDs)).
```

The proposed scheme has the following six events U_r's events(begin/end), GWT events(begin/end) and S_n events(begin/end).

```
(*=====*Events*=====*)
event start_User(bitstring).
event end_User(bitstring).
event start_GWt(bitstring).
event end_GWt(bitstring).
event start_Sn(bitstring).
event end_Sn(bitstring).
```

Results obtained from the proVerif are shown as follows:

```
1-- Query inj-event(end_Sn(IDs[])) ==> inj-event(start_Sn(IDs[]))
Starting query inj-event(end_Sn(IDs[])) ==> inj-event(start_Sn(IDs[]))
RESULT inj-event(end_Sn(IDs[])) ==> inj-event(start_Sn(IDs[])) is true.

2-- Query inj-event(end_GWt(IDGWti[])) ==> inj-event(start_GWt(IDGWti[]))
```

```
Starting query inj-event(end_Sn(IDGWti[])) ==> inj-event(start_IDGWt(IDGWti[]))
RESULT inj-event(end_GWt(IDs[])) ==> inj-event(start_GWt(IDGWti[])) is true.

3-- Query inj-event(end_User(IDur[])) ==> inj-event(start_User(IDur[]))...
Starting query inj-event(end_User(IDur[])) ==> inj-event(start_User(IDur[]))
RESULT inj-event(end_User(IDur[])) ==> inj-event(start_User(IDur[])) is true.

4-- Query not attacker(SK[])
completing...
Starting query not attacker(SK[])
RESULT not attacker(SK[]) is true.
```

Results 1,2 and 3 show that all the three processes are successfully started and terminated. Whereas, result 4 shows that the adversary cannot find the session key SK. Hence proposed protocol preserves correctness, secrecy, and authenticity.

6.2 | Security Analysis with BAN Logic

BAN logic use a set of rules for defining and analyzing protocols used for information exchange[35]. The rules used for the verification of the proposed protocol are shown in Table 1.

TABLE 1 Ban Logic Rules Table

Notation	Description
$P\mid\equiv X$	P believes that X
$P \lhd X$	P sees that X
$P\mid\sim X$	P once said X X
$P \Rightarrow X$	P has total jurisdiction over X
$\#(X)$	X is updated and fresh
(X, Y)	X,Y are components of formula
$\langle X\rangle_Y$	X is combined with Y
$(X)_K$	hash of message X using a key K
$P \leftrightarrow Q$	P and Q share a key for communication
$AIDT_i$	$AIDT_i$ one time key for current session
$\frac{P\mid\equiv P\leftrightarrow Q.p\lhd(X)_K}{P\mid\equiv Q\mid\equiv X}$	Message-Meaning rule
$\frac{P\mid\equiv\#(x)}{P\mid\equiv\#(X,Y)}$	Freshness-conjuncatenation rule
$\frac{P\mid\equiv\#(x),P\mid\equiv\mid\sim X}{P\mid\equiv Q\mid\equiv X}$	Nonce-verification rule
$\frac{P\mid\equiv Q\Rightarrow x,P\mid\equiv Q\mid\equiv X}{P\mid\equiv X}$	Jurisdiction rule

Using BAN Logic the proposed protocol has been verified with the help of eight goals for its accuracy. The goals are listed as follows:

- Goal 1: $GWT\mid\equiv U_r \overset{AID_U}{\longleftrightarrow}$ GWT
- Goal 2: $GWT\mid\equiv U_r\mid\equiv U_r \overset{AID_U}{\longleftrightarrow}$ GWT
- Goal 3: $S_n\mid\equiv GWT \overset{AID_U}{\longleftrightarrow}$ Sn
- Goal 4: $S_n\mid\equiv GWT\mid\equiv GWT \overset{AID_U}{\longleftrightarrow}$ Sn
- Goal 5: $GWT\mid\equiv S_n \overset{AID_U}{\longleftrightarrow}$ GWT
- Goal 6: $GWT\mid\equiv S_n\mid\equiv S_n \overset{AID_U}{\longleftrightarrow}$ GWT
- Goal 7: $U_r\mid\equiv GWT \overset{AID_U}{\longleftrightarrow}$ Ur
- Goal 8: $U_r\mid\equiv GWT\mid\equiv GWT \overset{AID_U}{\longleftrightarrow}$ Ur

Part1: For the proposed protocol, the idealized form is elaborated as under.

M1: $U \rightarrow$GWT: $AID_U, N_x :< N_u >_{K_{urg}}, Sn_{id}, Vr_1, T_1$

M2: $GWT \rightarrow$Sn: $AID_u, Vr_2, , T_2$

M3: $S_n \rightarrow$GWT: Vr_3, Sn_{id}, T_3

M4: $GWT \rightarrow U : Vr_4, Z_G :< AID_U >_{K_{UrG}}, T_4$

Part2: For the proposed solution, the following assumptions are used for analysis.

A1: $U_r| \equiv \#(N_u)(Vr_1)$

A2: $GWT| \equiv (r_u)(Vr_2)$

A3: $S_n| \equiv (AID_U)$

A4: $GWT| \equiv S_n \Rightarrow (AID_U)$

A5: $GWT| \equiv U_r \Rightarrow N_u$

A6: $S_n| \equiv GWT \Rightarrow r_u$

A7: $S_n| \equiv U_r \Rightarrow N_u$

A8: $U_r| \equiv S_n \Rightarrow r_u$

A9: $U_r| \equiv GWT \Rightarrow (AID_U)$

Part 3: The absolute analysis of the proposed protocol is performed on the BAN logic assumptions and the description of BAN logic rules are given below:
M1: $U \rightarrow$GWT: $AID_U, N_x :< N_u >_{K_{UrG}}, Sn_{id}\ T_1$ is timestamp of U
By applying the Ban logic "seeing rule", the following can be obtained

- S_1: $GWT \lhd AID_U, Sn_{id}, N_x :< N_u >_{K_{UrG}}, T1, Vr_1$

Using message-meaning of BAN logic rule and S_1, we get

- S_2: $GWT| \equiv U_r| \sim N_u$

Now using the "Freshness-conjuncatenation" rule and S_2, the following can be accomplished

- S_3: $GWT| \equiv U_r| \equiv N_u$

Using S_3 and the BAN logic "jurisdiction rule", the following is obtained

- S_4:$GWT| \equiv N_u$

Now using session key rule and S_4 of the BAN logic, the following is achieved

- S_5: $GWT| \equiv U_r \overset{AID_U}{\longleftrightarrow}$ GWT **(Goal 1)**

By applying "nonce-verification rule" of the BAN logic, accomplished the defined goal 2

- S_6: $GWT| \equiv U_r| \equiv U \overset{AID_U}{\longleftrightarrow}$ GWT **(Goal 2)**

M2: $GWT \rightarrow$Sn: $AID_U, Vr_2, , T_2$. Where, T_2 is timestamp of GWT
After this using the "seeing rule" of BAN logic, the following is achieved

- S_7: $S_n \lhd AID_U, Vr_2, , T_2$

Then the "message-meaning" rule of BAN logic and S_7 to achieve the following

- S_8: $S_n| \equiv GWT| \sim r_u$

Afterwords using S_8 and "Freshness-conjuncatenation" rule , the following can be obtained

- S_9: $S_n| \equiv GWT| \equiv r_u$

And now using S_9 and the BAN logic rule of "jurisdiction", the following can be achieved

- S_{10}: $S_n| \equiv r_u$

Further, using S_{10} and the session key rule of BAN logic , the following can be achieved

- S_{11}: $S_n| \equiv GWT \overset{AID_U}{\longleftrightarrow}$ Sn **(Goal 3)**

Now using "nonce-verification" rule of the BAN logic and S_{11} to achieve the following

- S_{12}: $S_n| \equiv GWT| \equiv GWT \overset{AID_U}{\longleftrightarrow}$ Sn. **(Goal 4)**

M3: $S_n \rightarrow$GWT: Vr_3, Sn_{id}, T_3 is the timestamp of S_n
Further, using the BAN logic "seeing-rule", the following can be obtained

- S_{13}: $GWT \triangleleft Vr_3, T_3, Sn_{id}$

Now using S_{13} and "message-meaning" rule of the BAN logic, the following can be achieved

- S_{14}: $GWT| \equiv S_n| \sim AID_U$

Then by applying "Freshness-conjuncatenation" rule and S_{14} of the BAN logic, the following can be obtained

- S_{15}: $GWT| \equiv S_n| \equiv AID_U$

Thereafter, by applying the BAN logic "assumption rule" and S_{15} and "jurisdiction rule", the following can be achieved

- S_{16}:$GWT| \equiv AID_U$

Now S_{16} and the BAN logic "session-key" rule can be used to obtain the following

- S_{17}: $GWT| \equiv S_n \overset{AIDT_U}{\longleftrightarrow}$ GWT. **(Goal 5)**

Then the BAN logic "nonce-verification" rule can be used to achieve the following

- S_{18}: $GWT| \equiv S_n| \equiv S_n \overset{AIDT_U}{\longleftrightarrow}$ GWT. **(Goal 6)**

M4: $GWT \rightarrow U : Vr_4, Z_G :< AID_U >_{K_{UrG}}, T_4$ is the timestamp of GWT
Now using the "seeing rule" of BAN logic to achieve the following

- S_{19}v: $U \triangleleft Vr_4, Z_G :< AID_U >_{K_{UrG}}, T_4$

Then by applying S_{19} and "message-meaning" rule of the BAN logic, the following can be achieved

- S_{20}: $U_r| \equiv GWT| \sim AID_U$

Now by combining the BAN logic "Freshness-conjuncatenation" rule with S_{20} the following can be obtained

- S_{21}: $U_r| \equiv GWT| \equiv AID_U$

After applying the BAN logic "jurisdiction rule" with S_{21}, the following is achievable

- S_{22}: $U_r| \equiv AID_U$

After applying the BAN logic "session-key" rule, the following is achieved

- S_{23}: $U_r| \equiv GWT \overset{AID_U}{\longleftrightarrow} U_r$ **(Goal 7)**

After applying BAN logic "nonce-verification" rule, the following can be obtained

- S_{24}: $U_r| \equiv GWT| \equiv GWT \overset{AID_U}{\longleftrightarrow} U_r$ **(Goal 8)**

It clear from the above BAN logic implementation that all the set goals have been achieved. Furthermore, it verifies that U_r, GWT and S_n are mutually authenticated successfully and securely and the session key agreement has been accomplished.

6.3 | Informal Security Analysis

In the informal security analysis, a protocol is analyzed against various known attacks. The proposed protocol is informally analyzed against different attacks. The attacks used in this article are listed as follows:

1. User Anonymity.
2. Resistance to Replay Attack.
3. Resistance to Eavesdropping Attack.
4. Dynamic Node Addition.
5. Resistance against Insider Attacks.
6. Resistance to Stolen Verifier Attack.
7. Forward Secrecy.
8. Resistance to DoS Attack.
9. Resistance to User Traceability

6.3.1 | User Anonymity

In various security applications, if the identity of a user is exposed can lead to severe consequences. User anonymity is a property of authentication protocols where it is desired that the identities of communicating users must not be revealed. For the sake of security, the original communicating user is not identifiable to any adversary during communication over public channels. In the proposed protocol, it has been ensured that the session parameter A_{ID_U} remains fresh for all transactions. The parameter A_{ID_U} is fully encrypted with the random number. In the proposed protocol, a user does not have any direct relation either with the Gateway Node GWT or the Sensor Node S_n. Therefore, no adversary can use the session specific parameter A_{ID_U} to get user information as every time the user communicate with updated parameters. It proves, that the proposed protocol provides user anonymity.

6.3.2 | Resilience Against Replay Attack

A replay attack is a type of security threat where an adversary can intercept the communication and later replays the messages without any modification to the contents of the message. In the proposed protocol, a timestamp ΔT has been used in every message transmitted to ensure message freshness in every session. It enables a receiver of the message to check the values of ΔT and determined whether a message is fresh or an old replayed message. Moreover, in the proposed protocol uses A_{ID_U} that is updated after every session. Therefore, the proposed protocol is secured against a replay attack, as the session parameters are updated after every session. So, an adversary is unable to launch a replay attack, because the parameters required to launch a replay attack are changed after each transaction.

6.3.3 | Resilience Against Eavesdropping attack

Eavesdropping is a type of security threat where hackers or adversaries intercept the communication between two parties without their knowledge. Intercepted communication can be used to find loophole and extract vital information. In the proposed protocol, all the session parameters are fully encrypted as well as updated after each session, so no adversary can launch an attack like eavesdropping because every time the user comes with new encrypted parameters over the public channel.

6.3.4 | Dynamic Node Addition

The proposed protocol is scalable and flexible, so to add a new node in a secure way is very much possible and the process is properly encrypted and secured. In IoT based WSN, the nodes added to the network is a routine, so it has been ensured that this process is secured against node capture attack. The new node immediately updates the symmetric keys and before transmitting its data, the node is encrypted with a random number to keep it secure over public channels.

6.3.5 | Resilience Against Insider Attack

An insider attack is a type of security threat, which is launched or executed on a system by a device or a person with authorized system access. In the proposed protocol, the session parameters are exclusive to one session, therefore, in every new session the parameters are updated and encrypted with random numbers. Consequently, for an insider, it is not possible to launch the insider attack because the insider does not have updated parameters in the next session.

Smartcard is used to stored private information for the purpose of authentication. However, using power analysis information stored in smartcard can be extracted. Therefore, losing a smartcard may result in loss of sensitive information.

6.3.6 | Resilience Against Perfect Forward Secrecy

Perfect Forward Secrecy means even if the secret key is compromised, an adversary cannot learn any previous sessions keys. It is actually protection of past session keys from any future threats. So an adversary cannot use a secret key to compromise any previous sessions. In the proposed protocol, upon a session completion, the session-specific parameters are updated with random numbers. So in worst scenario, if an adversary can learn a session key, is unable to interrupt the communication since the parameters in the next sessions are updated as A_{ID_U} with $A_{ID_{U(new)}} = E_{K_{GW}}(ID_{Ur} \| r_{u(new)})$ after the completion of previous session.

6.3.7 | Resilience Against DoS Attack

In the baseline Gope et al[11] scheme, the DOS attack is possible in the password update phase as during the password update previous password id s not being checked or verified. In the absence of this verification, the previous passwords can be used to send false messages thereby leading to DOS attack. In the proposed protocol, this deficiency has been taken care of by implementing the previous password verification before updating to the new password. Whenever a user needs to change the password, the intended user will insert his/her ID_{Ur}, previously used password PSW_{Ur} and computes $K_{UrG} \stackrel{?}{=} K^*_{UrG} \oplus h(h(ID_{Ur}) \oplus h(PSW_{Ur}))$. After verification of the previous K_{UrG}, the smart card requests the user to enter a new password PSW^*_{Ur} to the smart card. Therefore, the chances of launching a DoS attack has been diminished.

6.3.8 | Resilience Against Stolen Verifier Attack

Stolen verifier attack is a type of security threat where an adversary steals the data used for verification by the server in past or current sessions. In such a situation, if the verification information is stored is not encrypted, then any adversary can get the desired information. In order to avoid stolen verifier attack, the proposed scheme does not make use of verifier on server side rather the server performs authentication based on the information with user (i.e. information stored in smartcard plus user password and identity). When there is no verifier then there are no chances of its theft. Therefore, the proposed scheme is free of any threat from stolen verifier.

6.3.9 | Resilience Against User Traceability

User traceability may not directly lead to the loss of data, however, while communicating a user must not be traceable. Traceability has severe consequences in various applications especially when it comes to tracking someone or some object. For example, if an adversary can intercept a communication containing a specific pattern, then that pattern can be traced without even identification, movement of a person can be traced. The proposed protocol uses A_{ID_U} with proper encryption, and parameters update after completion of one cycle. Hence, an adversary cannot launch user traceability attack.

7 | SECURITY REQUIREMENTS AND PERFORMANCE ANALYSIS

This section presents the performance analysis of the proposed protocol. The analysis has been performed using three different parameters. The protocol has been analyzed for computation complexity using the number of operations and their costs and communication complexity using the number of messages exchanged in one session. However, first, the proposed protocol has been compared with existing state-of-the-art protocols using security requirements as listed below.

A1: User Anonymity.

A2: Replay Attack.

A3: Eavesdrop Attack.

A4: Dynamic Node Addition.

A5: Insider Attack.

A6: Stolen Verifier Attack.

A7: Forward Secrecy.

A8: Denial Of Service Attack.

A9: User Traceability.

The comparative results based on security requirements with the existing protocols[15,16,17], are shown in Table 2. The table clearly shows that the proposed protocol fulfills all the security requirements. In the comparison table, the "YES" means that the

TABLE 2 Security requirements comparison

Protocols	A1	A2	A3	A4	A5	A6	A7	A8	A9
Yeh et al.[7]	NO	NO	NO	NO	YES	NO	NO	NO	NO
Xue et al.[37]	NO	YES	NO	NO	NO	NO	NO	NO	NO
Jiang et al.[38]	YES	YES	NO	NO	NO	NO	NO	NO	NO
Das et al.[39]	YES	YES	YES	YES	YES	NO	NO	NO	NO
Gope et al.[11]	YES	YES	YES	YES	YES	NO	YES	NO	NO
Proposed	YES	YES	Yes	Yes	YES	YES	YES	Yes	YES

protocol provides the requirement while "NO" means that the protocol does not provide the specific requirement. The security analysis shows that the proposed protocol can fulfill all security requirements.

7.1 | Computation Cost Analysis

To ensure efficiency, authentication protocols are analyzed for Computation cost. Computation cost means the number of operations executed in one cycle of the protocol execution. Some operation like concatenation and XOR require negligible execution time and are, therefore, excluded from the computation cost in all protocols. The main focus is to analyze the protocols only for those operations that significantly contribute to the complexity of the protocol like a cryptographic operation that GTW, U_j and Sn need to execute. All the three participants GTW, U_j and Sn are considered in the calculation of the computation cost. A detailed description is shown in Table 3.

TABLE 3 Definition and conversion of various operations units

Notations	Definition and conversion
CC	Computation cost
TH	Computation cost of single hash function
TME	Computation cost of modular exponentiation
TSE	Computation cost of symmetric encryption
TSD	Computation cost of symmetric decryption

For comparative analysis of the proposed protocol, not only the computation cost of the proposed protocol but also other state-of-the-art protocols are computed using the same parameters. The other protocols consider for comparison are Yeh et al. scheme[7],Xue et al. scheme[37],Jiang et al. scheme[38],Das et al. scheme[39] and Gope et al. scheme[11].

Results of the comparative computation cost are presented in Table 4. It can be clearly observed from the table that the total computation cost of Yeh et al. scheme[7] is $8Th + 8TME$, Xue et al. scheme[37] is $22Th$, and that of Jiang et al. scheme[38] is $22Th$. Das et al. scheme[39] are $28Th$ and Gope et al. scheme[11] is $19Th$ However, the total computation cost of the proposed protocol is $5Th + 1TSE + 1TSD$. Furthermore, the proposed protocol does not have any security problems as compared to other schemes[15,16,17].

The computation time of the proposed and related protocols are shown in Table 4 by taking in consideration the timing computed for different cryptographic operations in[40]. The computed time is obtained using a PC with Intel Pentium dual core 2.20 GHz processor (E2200) and a RAM of 2048 MB on Ubuntu 12.04.1, a 32-bit OS. The last row in Table 4 shows the computation time of the protocols in milliseconds (ms). As per the results of Table 4, the proposed scheme takes 0.0207 ms for the authentication purpose and it can be clearly seen that the proposed protocol has superior in terms of computation complexity.

It can be clearly seen from Table 4 that the proposed protocol has clearly superior results in terms of computation complexity. Specifically, it has achieved 52.63% efficiency in comparison to the baseline protocol as depicted in Figure 4.

TABLE 4 Computation Cost Comparison of the Proposed protocol with existing state-of-the-art protocols

Computation Cost	Yeh et al.[7]	Xue et al.[37]	Jiang et al.[38]	Das et al.[39]	Gope et al.[11]	Proposed Scheme
CC_{User}	1Th+2TME	7Th	7Th	11Th	7Th	2Th
CC_{GWT}	4Th+4TME	10Th	10Th	11Th	9Th	2Th+1TSE+1TSD
CC_{Sn}	3Th+2TME	5Th	5Th	6Th	3Th	1Th
CC_{Total}	8Th+8TME	22Th	22Th	28Th	19Th	5Th+1TSE+1TSD
CC_{ms}	30.8184ms	0.0506ms	0.0506ms	0.0644ms	0.0437ms	0.0207ms

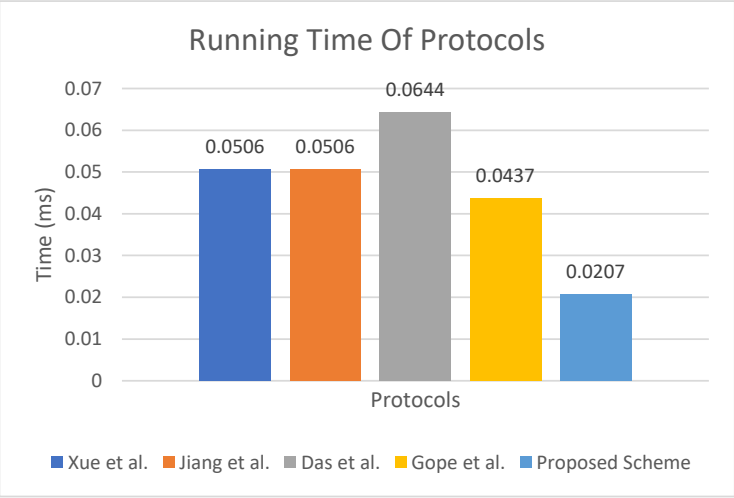

FIGURE 4 Computation time comparison of the proposed protocol with state-of-the-art protocols

7.2 | Communication Cost

Here a detailed description of the communication cost has been presented. The communication cost has been computed in terms of the number of messages exchanged in one transaction of the protocol and the size of each message. For a realistic comparison length of different constructs is considered to be the same for all schemes. For computing the communication cost of the proposed and related schemes, the length of random numbers is considered to be 128-bits, hash based parameters 160-bits, AES parameters 128-bits, and timestamp and sensor identity 32-bit each The message size has then been computed in bytes to compute the bandwidth consumption of proposed as well as the existing protocols. The total communication cost of the proposed protocol in comparison to the existing protocols has been presented in Table 5 in terms of total number of messages exchanged in one transaction of the protocol as well as the total length of all messages in bytes.

TABLE 5 Communication Cost Comparison of the Proposed protocol with existing state-of-the-art protocols

Protocols	**The Communication Overload**
Yeh et al.[7]	3 Messages (186 bytes)
Xue et al.[37]	4 Messages (276 bytes)
Jiang et al.[38]	4 Messages (284 bytes)
Das et al.[39]	4 Messages (264 bytes)
Gope et al.[11]	4 Messages (200 bytes)
Proposed Protocol	4 Messages (168 bytes)

The table shows that in Yeh et al.[7] scheme a total of three (3) messages are exchanged and the total bandwidth consumption of this protocol is 51 bytes. Similarly, in Xue et al.[37] scheme a total of three (03) messages are exchanged where the total bandwidth consumption is 51 bytes. Furthermore, in Jiang et al.[38] scheme a total of three messages are exchanged in one transaction where the total bandwidth cost is 51 bytes. Similarly, in Das et al.[39], a total of 51 bytes are consumed in terms of bandwidth for the exchange of a total of four (04) messages exchanged. Similarly, Gope et al[11] consumes 35 bytes in terms of bandwidth for exchange of four (04) messages whereas the proposed protocol consumes 35 bytes and exchanges a total of four (04) messages in one transaction.

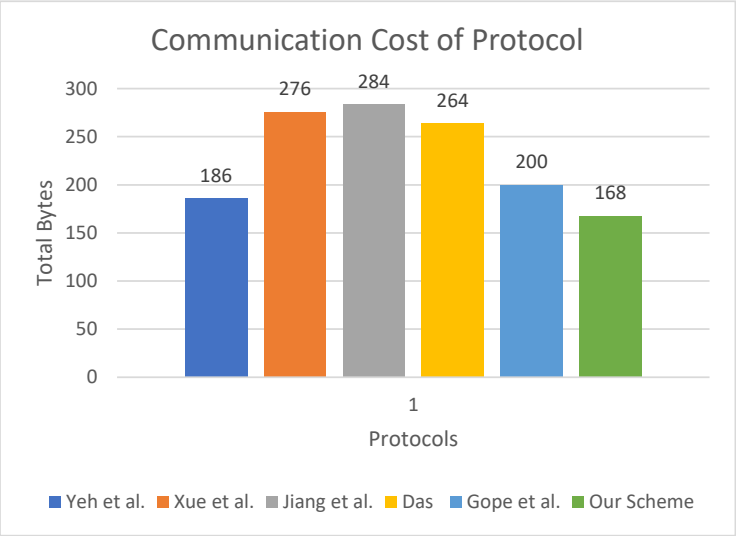

FIGURE 5 Communication cost of the proposed protocol in comparison to existing protocols

The communication cost comparison has been visualized in Figure 5. It can be observed from the figure that the proposed protocol has minimum computation complexity in comparison to all other existing protocol except Gope et al.[11]. Gope et al. have the same communication complexity as the proposed protocol but it is vulnerable to various security threats. Therefore, a protocol being vulnerable to even one security threat is not desirable even if its communication cost is the smallest of all. Furthermore, the proposed protocol has 19.04% efficiency in communication complexity as compared to the rest of the existing protocols.

8 | CONCLUSION

This article scrutinizes some recent authentication protocols designed for WSN and discloses that the proposed baseline protocol (Gope et al.[11]) is vulnerable to user traceability, stolen verifier, and DoS attacks because of that it may face security challenges. To counter the possible security challenges, an enhanced authentication protocol has been presented in this article. To gauge the security strengths and performance of the proposed protocol, it has been verified formally with ProVerif and BAN logic to test its correctness and key freshness. The proposed protocol successfully resists different attacks. It has the same communication cost as the baseline protocol since both exchange the same number of messages in one transaction of the protocol. However, in case of computation cost the proposed protocol achieves 52.63% more efficiency than the baseline protocol with more security and resistance to known attacks.

References

1. Wac K, Bults R, Van Beijnum B, et al. Mobile patient monitoring: the MobiHealth system. In: *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*IEEE. ; 2009: 1238–1241.

2. Athmani S, Bilami A, Boubiche DE. EDAK: An Efficient Dynamic Authentication and Key Management Mechanism for heterogeneous WSNs. *Future Generation Computer Systems* 2019; 92: 789–799.

3. Alotaibi SS. Registration Center Based User Authentication Scheme for Smart E-Governance Applications in Smart Cities. *IEEE Access* 2019; 7: 5819–5833.

4. Dimitriou T, Ioannis K. Security issues in biomedical wireless sensor networks. In: *First International Symposium on Applied Sciences on Biomedical and Communication Technologies*IEEE. ; 2008: 1–5.

5. Desmedt Y, Frankel Y, Yung M. Multi-receiver/multi-sender network security: efficient authenticated multicast/feedback. In: *Eleventh Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM'92*IEEE. ; 1992: 2045–2054.

6. Suhag D, Gaur SS, Mohapatra A. A proposed scheme to achieve node authentication in military applications of wireless sensor network. *Journal of Statistics and Management Systems* 2019; 22(2): 347–362.

7. Yeh HL, Chen TH, Liu PC, Kim TH, Wei HW. A secured authentication protocol for wireless sensor networks using elliptic curves cryptography. *Sensors* 2011; 11(5): 4767–4779.

8. Jan MA, Khan F, Alam M, Usman M. A payload-based mutual authentication scheme for Internet of Things. *Future Generation Computer Systems* 2019; 92: 1028–1039.

9. Li X, Peng J, Obaidat MS, Wu F, Khan MK, Chen C. A Secure Three-Factor User Authentication Protocol With Forward Secrecy for Wireless Medical Sensor Network Systems. *IEEE Systems Journal* 2019.

10. Joshi A, Mohapatra AK. Authentication protocols for wireless body area network with key management approach. *Journal of Discrete Mathematical Sciences and Cryptography* 2019; 22(2): 219-240. doi: 10.1080/09720529.2019.1582869

11. Gope P, Hwang T, others . A Realistic Lightweight Anonymous Authentication Protocol for Securing Real-Time Application Data Access in Wireless Sensor Networks.. *IEEE Trans. Industrial Electronics* 2016; 63(11): 7124–7132.

12. Li X, Niu J, Kumari S, Wu F, Sangaiah AK, Choo KKR. A three-factor anonymous authentication scheme for wireless sensor networks in internet of things environments. *Journal of Network and Computer Applications* 2018; 103: 194 - 204. doi: https://doi.org/10.1016/j.jnca.2017.07.001

13. Li X, Peng J, Niu J, Wu F, Liao J, Choo KR. A Robust and Energy Efficient Authentication Protocol for Industrial Internet of Things. *IEEE Internet of Things Journal* 2018; 5(3): 1606-1615. doi: 10.1109/JIOT.2017.2787800

14. Li X, Niu J, Bhuiyan MZA, Wu F, Karuppiah M, Kumari S. A Robust ECC-Based Provable Secure Authentication Protocol With Privacy Preserving for Industrial Internet of Things. *IEEE Transactions on Industrial Informatics* 2018; 14(8): 3599-3609. doi: 10.1109/TII.2017.2773666

15. Kumar P, Lee SG, Lee HJ. E-SAP: efficient-strong authentication protocol for healthcare applications using wireless medical sensor networks. *Sensors* 2012; 12(2): 1625–1647.

16. He D, Kumar N, Chen J, Lee CC, Chilamkurti N, Yeo SS. Robust anonymous authentication protocol for health-care applications using wireless medical sensor networks. *Multimedia Systems* 2015; 21(1): 49–60.

17. Mir O, Munilla J, Kumari S. Efficient anonymous authentication with key agreement protocol for wireless medical sensor networks. *Peer-to-Peer Networking and Applications* 2015: 1–13.

18. Hayajneh T, Mohd BJ, Imran M, Almashaqbeh G, Vasilakos AV. Secure Authentication for Remote Patient Monitoring with Wireless Medical Sensor Networks. *Sensors* 2016; 16(4): 424.

19. Jiang Q, Lian X, Yang C, Ma J, Tian Y, Yang Y. A bilinear pairing based anonymous authentication scheme in wireless body area networks for mHealth. *Journal of Medical Systems* 2016; 40(11): 231.

20. Amin R, Islam SH, Biswas G, Khan MK, Kumar N. A robust and anonymous patient monitoring system using wireless medical sensor networks. *Future Generation Computer Systems* 2016.

21. Ibrahim MH, Kumari S, Das AK, Wazid M, Odelu V. Secure anonymous mutual authentication for star two-tier wireless body area networks. *Computer methods and programs in biomedicine* 2016; 135: 37–50.

22. Liu J, Zhang L, Sun R. 1-RAAP: An Efficient 1-Round Anonymous Authentication Protocol for Wireless Body Area Networks. *Sensors* 2016; 16(5): 728.

23. Wu L, Zhang Y, Li L, Shen J. Efficient and Anonymous Authentication Scheme for Wireless Body Area Networks. *Journal of medical systems* 2016; 40(6): 1–12.

24. Mohd BJ, Hayajneh T, Shakir MZ, Qaraqe KA, Vasilakos AV. Energy model for light-weight block ciphers for WBAN applications. In: *EAI 4th International Conference on Wireless Mobile Communication and Healthcare (Mobihealth)*IEEE. ; 2014: 1–4.

25. Hayajneh T, Doomun R, Al-Mashaqbeh G, Mohd BJ. An energy-efficient and security aware route selection protocol for wireless sensor networks. *Security and Communication Networks* 2014; 7(11): 2015–2038.

26. Mohd BJ, Hayajneh T, Quttoum AN. Wavelet-transform steganography: Algorithm and hardware implementation. *International Journal of Electronic Security and Digital Forensics* 2013; 5(3-4): 241–256.

27. Liu J, Zhang Z, Chen X, Kwak KS. Certificateless remote anonymous authentication schemes for wirelessbody area networks. *IEEE Transactions on Parallel and Distributed Systems* 2014; 25(2): 332–342.

28. Movassaghi S, Abolhasan M, Lipman J, Smith D, Jamalipour A. Wireless body area networks: A survey. *IEEE Communications Surveys & Tutorials* 2014; 16(3): 1658–1686.

29. Venkatasubramanian KK, Gupta SK. Physiological value-based efficient usable security solutions for body sensor networks. *ACM Transactions on Sensor Networks (TOSN)* 2010; 6(4): 31.

30. Wang D, Wang P. On the anonymity of two-factor authentication schemes for wireless sensor networks: Attacks, principle and solutions. *Computer Networks* 2014; 73: 41–57.

31. Chen TH, Shih WK. A robust mutual authentication protocol for wireless sensor networks. *ETRI journal* 2010; 32(5): 704–712.

32. Li CT, Lee CC, Weng CY, Chen SJ. A Secure Dynamic Identity and Chaotic Maps Based User Authentication and Key Agreement Scheme for e-Healthcare Systems. *Journal of medical systems* 2016; 40(11): 233.

33. Mainetti L, Patrono L, Vilei A. Evolution of wireless sensor networks towards the internet of things: A survey. In: *19th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*IEEE. ; 2011: 1–6.

34. Costa DG, Figuerêdo S, Oliveira G. Cryptography in Wireless Multimedia Sensor Networks: A Survey and Research Directions. *Cryptography* 2017; 1(1): 4.

35. Burrows M, Abadi M, Needham R. A Logic of Authentication. *SIGOPS Oper. Syst. Rev.* 1989; 23(5): 1–13.

36. Li M, Lou W, Ren K. Data security and privacy in wireless body area networks. *IEEE Wireless Communications* 2010; 17(1): 51–58.

37. Xue K, Ma C, Hong P, Ding R. A temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks. *Journal of Network and Computer Applications* 2013; 36(1): 316–323.

38. Jiang Q, Ma J, Lu X, Tian Y. An efficient two-factor user authentication scheme with unlinkability for wireless sensor networks. *Peer-to-peer Networking and Applications* 2015; 8(6): 1070–1081.

39. Das AK. A secure and robust temporal credential-based three-factor user authentication scheme for wireless sensor networks. *Peer-to-peer Networking and Applications* 2016; 9(1): 223–244.

40. Kilinc HH, Yanik T. A survey of SIP authentication and key agreement schemes. *IEEE Communications Surveys & Tutorials* 2013; 16(2): 1005–1023.

How to cite this article: Ghani A, Mansoor K, Mehmood S, Chaudhry SA, Rahman AU, Najmus Saqib M. Security and key management in IoT-based wireless sensor networks: An authentication protocol using symmetric key. Int J Commun Syst. 2019;e4139. https://doi.org/10.1002/dac.4139