International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8, Issue-1, May 2019

A USB- Bluetooth Two Factor Mutual Authentication Security Protocol for Wireless Sensor Networks

Preethi Elizabeth Thomas, Sumitra Binu

Abstract*: Wireless sensor networks are easy to deploy, effective, and can monitor unattended environments. As the data transmitted through these networks is highly sensitive, the security of the networks is important and strong authentication measures must be in place. Authentication is done by means of a security protocol, wherein a user is authenticated through certain factors such as a smartcard or a password, and several mathematical calculations such as hashing, and XOR operations. Several previously proposed authentication protocols and their flaws are discussed in this paper. We propose a new two factor mutual authentication protocol using a USB-Bluetooth token as the second factor, to overcome the security flaws seen in previous schemes. We also provide security analysis as well as Scyther results in support of the proposed protocol. The proposed protocol can be used across various fields such as healthcare, agriculture, traffic monitoring etc.

***Index Terms*: Bluetooth, Key agreement, Two factor mutual authentication, USB, Wireless sensor networks, WSN.**

I. INTRODUCTION

A Wireless Sensor Network (WSN) can be defined as a number of sensors equally distributed in an area. The sensors rely on wireless connectivity and network formation in order to constantly monitor the environment they are in and send the collected data to a central location. Each network is composed of a large number of 'nodes' and each node has a number of sensors. The flow of data is bidirectional- that is, data can be sent and received by the nodes. The main characteristics of a wireless sensor network are its ability to cope with node-failures and constraints regarding power consumption, its ability to be deployed on a large scale, its ease of use, and its resistance to harsh environmental conditions among others. WSNs find their application in various areas such as the military, agriculture industry, traffic monitoring etc [1]. They are used by the military for battleground surveillance. In healthcare, WSNs can be used to monitor patients. WSNs can be used for monitoring safety in areas like coal mines and nuclear power plants where certain factors such as temperature, air purity, and toxicity of materials can lead to catastrophe if left unmonitored. Miners are prone to respiratory illness and/or death due to increased risk of exposure to explosive or toxic gases. The use of WSNs that constantly monitor the air purity and composition in mines could prevent disaster. In nuclear power plants, WSNs could be set up to monitor levels of radioactivity and alert authorities when they cross permissible limits.

The data transmitted by wireless sensor networks is highly sensitive and it is key that this information is not compromised. Hence, in order to promote the widespread use of wireless sensor networks, they must be secure and reliable. Concerns with WSNs include unauthorised access to the data in transmission, password guessing and man in the middle attacks. Authentication measures must be put in place so that only authorised personnel are able to transmit and receive data. A user may be authenticated using one of three methods - an alphanumeric phrase known to them, such as a password, an object they possess, such as a smart card, or a characteristic unique to them, such as a fingerprint. Password based authentication mechanisms are the most common as they are both cost effective and easy to implement. The limitation with such a scheme, however, is that it requires a password verifier table. This causes the scheme to be susceptible to password verifier attacks. If a password is guessed, the user can be locked out of the system and/or impersonated by the attacker, thereby compromising the network. In some schemes, a smart card is used as a second authenticating factor. However, such schemes are susceptible to security attacks such as password guessing or user impersonation, using data extracted from lost or stolen smart cards. This paper proposes a USB-Bluetooth token as the second authenticating factor that works in connected as well as disconnected states. In cases where Bluetooth connectivity is an issue, the USB can be directly plugged into the system. The advantage of the proposed token over a smart card is that a smart card requires a card reader whereas most systems come with inbuilt USB slots. This protocol provides mutual authentication between the user, sensor nodes and the gateway node. The rest of the paper is organised in the following manner. Section 2 is the literature review, Section 3 introduces the proposed protocol and its phases. Section 4 is the security analysis of the proposed protocol, Section 5 demonstrates the security analysis done using Scyther tool. Section 6 concludes the paper and is followed by the references.

II. LITERATURE REVIEW

Wong et al [2] proposed an authentication system for wireless sensor networks that used one-way hash functions and XOR operations.

Revised Manuscript Received on 30 May 2019.
***** Correspondence Author
Preethi Elizabeth Thomas*, Department of Computer Science, CHRIST (Deemed to be University), Bangalore, Karnataka 560029, India.
Dr. Sumitra Binu, Department of Computer Science, CHRIST (Deemed to be University), Bangalore, Karnataka 560029, India.

It was the first of the security initiatives but was susceptible to stolen verifier attack.

Das [3] analysed Wong's scheme, discussed its flaws and introduced a smart card based two factor mutual authentication scheme which did not require a verifier table. This addressed the issues of having multiple logged in users with the same login ID, password guessing, impersonation and stolen verifier attacks. However, this system was susceptible to smart card attacks, sensor impersonation, node capture attacks, gateway node bypassing, parallel session attacks, and privileged insider attack. It also did not allow users to change or update their passwords.

Chen and Shih [4] further improved Das' scheme when they identified its susceptibility to parallel session attacks as well as the lack of mutual authentication between the gateway node and the sensor node (SN). However, Chen-Shih's scheme was susceptible to stolen smart card attacks and the authenticity of the sensor nodes could not be guaranteed as the authors assumed that the GWN could verify SN before deploying it. Vaidya [6] later discovered that in hostile or unattended environments, sensor nodes could be compromised and controlled or impersonated by attackers.

Khan and Algathbar [5] modified Das' scheme to withstand gateway node bypassing and privileged-insider attack and to also allow users to change/update their passwords. Its security concerns were stolen smart card and sensor node impersonation attacks. There was also a discrepancy in values calculated by the GWN and the smart card.

Vaidya et al. [6] identified vulnerabilities in the aforementioned schemes [2-5] and proposed a two-factor mutual authentication and key agreement scheme. The scheme claimed to be lightweight and efficient in terms of communication and computation to reduce energy use, to have the provision to change/update password and other factors, to provide key agreement for secure data transfer, to provide mutual authentication between user device (UD), GWN and SN to prevent active attacks, and to prevent stolen smart card attacks. However, it was found to be susceptible to gateway node bypassing attacks using secret information from sensor nodes, user impersonation attacks, and gateway node bypassing with users' smart cards.

Kim et al. [7] pointed out the demerits of Vaidya's scheme [6] and proposed a scheme that claimed to resolve issues such as stolen verifier attacks, stolen smart card attacks, replay attacks, parallel session attacks, password guessing, etc. The scheme was efficient in terms of both communication and computational cost. However, it was unable to withstand some attacks such as lost smart card, man-in-the-middle attacks and impersonation attacks [20].

Xue et al. [8] proposed a temporal credential based, lightweight authentication key agreement scheme for WSNs using hash and XOR computations. In this scheme, the gateway node issues temporal credentials to users and sensor nodes with the help of password-based authentication. However, this scheme was later analysed by Li et al. [14] and was found to be susceptible to password guessing attacks, stolen verifier attacks, lost smart card attacks and insider attacks. He et al. [16] deemed the scheme to be ineffective due to design defects and susceptibilities to attacks.

In 2014, Turkanović et al. [9] proposed a lightweight user authentication scheme for WSN that was based purely on hash and XOR computations in an attempt to save both computation and communication resources. The scheme claimed to prevent many attacks, fulfil basic security attributes, and have higher complexity. However, authors [17,18] studied the scheme and stated that this scheme was inapplicable for implementation as there were several security drawbacks such as stolen verifier attacks, sensor node spoofing attacks, cryptographic attacks, and a lack of backward secrecy.

Amin-Biswas [10] developed a modified version of the hash and XOR operations for resource constrained environments. The paper addressed both security and efficiency, and stated that the design had multiple attractive features as the system had many gateway nodes. The drawback of this scheme was that there was leakage of session data.

Yeh et al. [11] revealed that Chen et al.'s [4] protocol faced difficulties in updating users' passwords and was vulnerable to insider attacks. As an improvement of Chen et al.'s protocol, Yeh et al. presented the first user authentication protocol using elliptic curve cryptography (ECC) in WSN environments. Choi et al. [15] brought forward the fact that this scheme did not have perfect forward secrecy, user-sensor key agreement or mutual authentication.

Shi et al. [12] proposed a new smart card-based user authentication protocol using ECC for WSNs. Shi et al.'s protocol performs more efficiently, both in terms of computation and communication costs, and provides better security than Yeh et al.'s [11] protocol. However, their improved scheme is not completely secure and is susceptible to session key attacks, stolen smart card attacks and sensor energy exhausting attacks [15].

Chang et al. [13] proposed a scheme that provided user privacy by using dynamic identities and provided better security functionality than Kim et al.'s [7] scheme. However, the scheme lacked forward secrecy, did not withstand password guessing attacks, and its password updation was inaccurate [14].

Li et al. [15] proposed an advanced temporal credential-based scheme for WSNs as an alternative to Xue et al.'s [10] scheme. However, in this scheme, Lee [19] noted that an adversary could derive the temporal credentials, users' identities, and verification values stored in the GWN's verifier table, as well as expiration time from revealed messages. Therefore, this scheme was susceptible to stolen verifier attacks and user impersonation attacks, and the anonymity of the user was not guaranteed. The adversary could also derive the previous session keys of a user or a sensor node and access all transmitted communication.

III. PROPOSED PROTOCOL

List of symbols used

U_{id} - User ID

PW_i - Users password

α - hashed valued of U_{id}

β - hashed valued of PW_i

g - generator of order p of group Z_p*
p - a 1024 bit prime number
h(.) - a one way hash function
GWN - gateway node
U_{id}' - User ID entered at the time of login
PW_i' - Password entered at time of login
ΔT - pre defined small interval of time
T1 - Timestamp generated by user
T2 - Timestamp of when server receives the login request
x_s - Server's secret key
R1- nonce value generated by user's device
R2 - nonce value generated by GWN
SK - session key
DID_i - dynamic id generated at login to authenticate user and GWN
Sn_i - sensor node i
S_{id} - identity value of sensor node Sn_i stored with the GWN
Sn_{id} - identity value of a sensor node Sn_i stored within it
$SDID_i$ - dynamic id generated for authentication between sensor node and gateway node

A.Initialisation

The GWN maintains a table of the hashed IDs of the registered users (α) along with a status bit that represents the user's logged in status. If the status bit is 1, then the user is logged in. The sensor nodes register with the GWN. The GWN generates a S_{id} for each node. The hashed value of the S_{id} of the node is stored in the GWN and the GWN computes $Sn_{id} = h(S_{id} \| h(x_s))$
And stores it in the corresponding sensor node Sn_i.

B.Registration Phase

UR1. User chooses U_{id} and PW_i.
UR2. $\alpha=h(U_{id})$, $\beta=h(PW_i)$ computed by the user and sent securely to the GWN via a secure communication channel.
UR3. GWN compares the value against a stored list of registered users. If the ID has already been taken, the user is prompted to choose another one.
UR4. The GWN computes the following values

$$C = h(\alpha \| \beta)$$

$$A = g^{h(h(\alpha) \| xs)} \bmod p$$

$$B = g^{h(h(C) \| h(\beta))} \bmod p \oplus A$$

$$D = \log_{\beta} B$$

$$Z = \log_{h(xs)} \alpha$$

UR5. And h(.), D, C, g, p, Z are stored on Bluetooth enabled USB token.
This Bluetooth enabled USB token is sent securely to the user.

C.Login Phase

UL1. User enters Bluetooth enabled USB token into terminal and enters user ID and password U_{id}' and PW_i'
UL2. Bluetooth enabled USB token computes
$\alpha'=h(U_{id}')$, $\beta'=h(PW_i)$ and
$C'=h(\alpha' \| \beta')$
UL3. If C'?=C is true , login proceeds.
UL4. The following values are computed by the Bluetooth enabled USB token
$B' = (\beta')^D$
$A'=B' \oplus g^{h(h(C') \| h(\beta'))} \bmod p$
UL5. Bluetooth enabled USB token generates a random nonce value R1.
UL6. Bluetooth enabled USB token computes
$K_1= h(A') \oplus R1$
$L= \log_{A'} K_1$

UL7. Bluetooth enabled USB token computes
$DID_i = h(h(A') \| R1 \| T1 \| K_1)$
where T1 is the current timestamp of the user's system.
UL8. login request <DID_i, T1, Z, L, g, p> is sent to the GWN through a secure channel.

D.Verification and Authentication Phase

UA1. GWN receives login request <DID_i, T1, Z, L, g, p> from the Ui at the timestamp T2.
UA2. ΔT'=T2 - T1 is computed. If ΔT' < = ΔT , the GWN proceeds with the authentication process.
Otherwise, the authentication process is terminated.
UA3. GWN computes
$\alpha'= (h(x_s))^Z$
$A' = g^{h(h(\alpha') \| xs)} \bmod p$
$K_1 = (A')^L$
$R1 = h(A') \oplus K1$
$DID'=h(h(A')\|R1\| T1\| K_1))$

UA4. DID_i''?= DID_i, then verification complete and the user is authenticated.
UA5. GWN generates random nonce R2.
UA6. GWN computes the session key $SK=h(DID_i'\|R1\|R2)$
UA7. The GWN computes
$V_2= \log_{\alpha'} h(DID_i') \oplus R2$ and
$K_2= \log_{R2} h(\alpha')$

UA8. GWN sends <K_2, V_2> to the Bluetooth enabled USB token
UA9. Bluetooth enabled USB token receives <K_2, V_2> and computes
$R2'= V_2 \oplus \log_{\alpha} h(DID_i)$
$K_2'= \log_{R2} h(\alpha')$

UA10. If $K_2'?=K_2$, then the GWN is authenticated. Otherwise, further communication is terminated.
UA11. USB computes session key
$SK = h(DID_i \| R1 \| R2)$
All further communication is encrypted using the session key SK.

UA12. The GWN updates the status bit of the user corresponding to α' to 1 i.e., logged in status.

E. Mutual Authentication between GWN and sensor nodes

Step 1. GWN requests information from a sensor node Sn_i

Step 2. From its list of registered sensor nodes, the GWN finds the S_{id} of the required sensor node Sn_i and computes the following

$Sn_{id} = h(S_{id} \| h(x_s))$

GWN generates a random nonce Rg.

$S_{curr} = \log_{Rg} SK$,

where SK is the session key shared between the user, nodes and the GWN for the current session.

$K = \log_{Tg} (Sn_{id} \oplus Rg)$

where Tg is the current timestamp of the GWN.

$SDID_i = h(Sn_{id} \| Tg \| Rg)$

Step 3. GWN sends $< SDID_i, K, Tg, S_{curr}>$ to Sn_i

Step 4. Sensor node Sn_i receives $<SDID_i, K, Tg, Scurr>$ from the GWN and computes

$Rg = (Tg)^{(Snid \oplus K)}$

$SK = (Rg)^{Scurr}$

$SDID_i' = h(Sn_{id} \| Tg \| Rg)$

Step 5. If $SDID_i' ? = SDID_i$, then the GWN is authenticated. Otherwise, communication between the GWN and the sensor node Sn_i is terminated.

Step 6. Sn_i then computes

$Q = \log_{SDIDi'} (Rs)$

$P = h(Sn_{id} \| Ts \| Rs)$

Step 7. Node sends <P, Q, Ts> to the GWN

Step 8. GWN receives <P, Q, Ts> and computes

$Rs = (SDIDi')^{Q}$

$P' = h(Sn_{id} \| Ts \| Rs)$

If P'?= P , then the sensor node is authenticated. Otherwise, communication between the GWN and the sensor node Sni is terminated.

F. Password Change Phase

UP1. User enters Bluetooth enabled USB token and enters U_{id}' and PW_i'.

UP2. The following values are computed by the Bluetooth enabled USB token

$\alpha' = h(U_{id}')$, $\beta' = h(PW_i)$

$C' = h(\alpha' \| \beta')$

UP3. If C'?=C, the user is prompted to enter new password PW_i*.

Otherwise, the password change phase is terminated.

UP4. The following are computed by the Bluetooth enabled USB token

$\beta^* = h(PW_i^*)$

$C^* = h(\alpha \| \beta^*)$

$B^* = g^{h(h(C^*) \| h(\beta^*))} \bmod p \oplus A$

$D^* = \log_{\beta^*} B^*$

UP5. D*, C* replace the values of D and C on the Bluetooth enabled USB token.

IV. SECURITY ANALYSIS

A. User Anonymity

If an attacker AK intercepts the login request, < DIDi, T1, Z, L, g, p> of Ui, AK would be unable to extract the Uid as it is protected by hashing and is not stored in plaintext.

If AK manages to acquire or access Ui's Bluetooth enabled USB token and extracts all information stored in it, i.e., < h(.), D, C, g, p, Z >, AK is still unable to know the user's identity as none of the parameters contain Uid in plaintext. All values are protected by one-way hash functions. Thus, the user is anonymous in this scheme.

B. User Untraceability

Traceability of a user means that an attacker can distinguish a user in different login sessions. An attacker AK cannot keep track of Ui from the login request, <DIDi, T1, Z, L, g, p> sent to the GWN as the DIDi and T1 values are never constant; they change in every session. Hence the user is not traceable in this scheme.

C. Replay attack

An attacker AK can intercept and replay the login request, $<DID_i, T1, Z, L, g, p>$ of U_i to the GWN. However, the GWN would not be able to authenticate this message as it would fail the freshness test of the timestamp T1. Also, the nonce values generated are unique. They are only generated once and cannot be duplicated by the user and hence the replay attack will fail.

If AK replaces the original message with $<DID_i$,Te,C',K1,V1>, where Te is the timestamp generated by the attacker, then this message would pass the freshness test but would fail the next verification step as T1 is used in the calculation of DID_i. Therefore, when the GWN computes DID_i' using the Te value, DID_i' will not match DID_i and the replay attack will fail.

D. Password Guessing attack

In general, two types of password guessing attacks are possible. The first type is an online attack where an attacker AK intercepts the messages between the user and the GWN and is able to derive the value of the password from the intercepted values, either through mathematical calculations or dictionary attacks. The second type of attack happens offline. In this type of attack, AK has the user's device and having extracted the values stored on it, is able to derive the user's password through either mathematical calculations or dictionary attacks. In this scheme, if an attacker AK receives the login request $<DID_i$, T1, Z, L, g, p> of the user U_i from an insecure channel, AK can neither guess nor calculate PW_i of U_i from this login message as all parameters are hashed. None of the values that are being passed contain PW_i in plaintext. Although dictionary attacks are technically possible on C, they will not occur as C is not being passed directly. Therefore, online password guessing attack is prevented in this scheme. If AK manages to acquire or access the user U_i's Bluetooth enabled USB token and extract all the values from it,

AK is unable to get the U_{id} and PW_i of U_i from the extracted values as the U_{id} and PW_i on the Bluetooth enabled USB token are protected by a one-way hash function h(.) and not stored in plaintext.

Dictionary attacks will fail here as well since the value of C is a hashed concatenation of the user's hashed ID and hashed password, i.e. C = h (α || β). Hence, offline password guessing attack is also prevented.

E. Stolen/lost Bluetooth enabled USB token

If an attacker AK manages to acquire or access the Bluetooth enabled USB token of Ui and extracts the values stored in it, < h(.), D, C, g, p, Z >, AK is unable to login using the users credentials as he does not have access to the U_{id} and PW_i of the user U_i .Without these values, during the login phase, the Bluetooth enabled USB token cannot calculate the values $\alpha'=h(U_{id}')$, $\beta'=h(PW_i)$ and $C'=h(\alpha' || \beta')$. Therefore, C' !=C, where C is the value stored on the card, and the login process will be terminated. Hence, stolen USB attack is prevented in this scheme.

F. Stolen verifier attack

In the proposed protocol, the password of the user is not being stored in the GWN. Therefore, if an attacker AK extracts values from the GWN, AK is only able to access the hashed U_{id} and is unable to use this value to impersonate a legitimate user. If AK has also managed to acquire a legitimate user's USB token and extracts the values stored on it, AK is still unable to impersonate the user with those values and the hashed U_{id} of the user, and hence will not have access to the network.

G. Privileged insider attack

A privileged insider attack occurs when an insider with access to the system is able to impersonate a user using values stored in the server. If a user's password is stored in the server, the insider can use their access to obtain this password and login using the user's account. In this scheme, during the registration phase, U_{id} and PW_i are being transmitted to the GWN in the form of the hashed values α and β where $\alpha=h(U_{id})$ and $\beta=h(PW_i)$. Hashing is a one way function and hence the insider cannot retrieve the password from β. The password is not being stored in the GWN and hence the insider cannot access the password from the GWN. Therefore, privileged insider attack is prevented in this scheme.

H. Man in the middle/ Eavesdropping attacks

If an attacker AK has been eavesdropping on the communication between the user's device and the GWN and has intercepted the login request <DID_i, T1, Z, L, g, p> from the user U_i and sends this to the GWN as his login request, the GWN would not be able to authenticate AK as the login request would fail the freshness test of the timestamp T1.

AK is also unable to gain any useful information about the user U_i from the login request as the U_{id} and PW_i cannot be extracted from the values in the login request.

I. Gateway node bypassing

If an attacker AK attempts to gain access to the network by bypassing the GWN, AK's attempts would fail as access to the network is only granted to the user U_i once the GWN has authenticated U_i. This authentication requires the GWN's secret key x_s, which is only known to the GWN. Therefore, gateway node bypassing attack is not possible in our scheme.

J. Gateway node impersonation

An attacker AK cannot gain information about the user by impersonating the GWN. During the verification and authentication phase, the GWN computes the value $A = g^{h(h(\alpha') || xs)} \bmod p$ which requires the GWN's secret key x_s . This value is not available to AK and hence AK will be unable to compute the value DID_i' required for the comparison $DID_i'?= DID_i$. Without the correct value of DID_i', the mutual authentication between the GWN and the user will fail, and further communication will be stopped. Hence, gateway node impersonation attack is prevented in the proposed scheme.

K. Session key agreement

Once a user has logged in after being authenticated, all further communication between the user and the network is encrypted using a symmetric session key generated each session. The user, GWN and sensor nodes all share this session key for the duration of the session. This key is unique to each session and cannot be duplicated. The session key is calculated independently on the client and server side so an attacker cannot gain access to the key by intercepting messages between the user and the GWN. Since the session key encrypts all communication between the user and the GWN, an attacker will be unable to decrypt the messages by eavesdropping.

L. Many logged in users with same id

If multiple users were able to log in with the same ID, there would be confusion during communication as the GWN would not be able to ensure that the information being transmitted was being sent to the intended recipient. Hence in the proposed scheme, when a user is registering with the network, the hashed U_{id} they choose is compared against a table of previously existing users to ensure that their U_{id} is unique. This prevents having multiple users with the same ID. The GWN maintains a status bit column which states whether a particular user is logged in or not. Therefore, once a user has logged in, a parallel session cannot be started by an attacker with the same credentials. Hence, this scheme prevents against having many logged in users with the same ID.

M. Parallel session attack

The GWN maintains a status bit column which indicates whether a particular user is logged in or not. Therefore, once a user has logged in, a parallel session cannot be started with the same credentials. Hence parallel session attack fails.

N. Sensor node impersonation attack

Each sensor node Sn has a unique value $Sn_{id} = h(S_{id} || h (x_s))$ stored in it. If an attacker AK intercepts the message <$SDID_i$, K, Tg, S_{curr}> sent from the GWN to the sensor node and attempts to replay the GWN's message to the sensor node, the attack will fail as the message will fail the freshness test of the timestamp Tg. If AK attempts to duplicate the message with the timestamp of AK's system, the sensor node will not authenticate AK as a legitimate user as the value of $SDID_i$ is dependent on the timestamp Tg. AK is also unable to obtain the sensor node's identity Sn_{id} and cannot generate a value that can be authenticated by the sensor node Sn_i without it.

If AK intercepts the message <P, Q, Ts> from Sn to the GWN and tries to replay the message, the attack will fail the freshness test of the timestamp Ts. AK is also unable to duplicate this message because, if AK replaces the timestamp with his own, the authentication will fail as P is computed using timestamp Ts. AK also cannot replicate the message, as the value of the sensor node's identity Snid is unavailable to AK without which P cannot be authenticated.

V. SCYTHER RESULTS AND ANALYSIS

The implementation of the proposed protocol is done by using Scyther tool. Scyther is a tool for the formal, automatic verification of security protocols. Through Scyther, the protocol can be simulated and the validity of its claims such as secrecy of a value, niagree, and nisynch can be analysed. It is assumed that an attacker or intruder can eavesdrop on the network.

Secrecy: The claims made are that the user's credentials, the generator p, prime number p, server's secret key x_s, C and the nonces remain secret.

Niagree (Non Injective Agreement) : This claim claims that the values of the variables sent and received are agreed upon by both the sender and the receiver.

Nisynch (Synchronisation) : This claims that the runs have the same content and execute in the correct order.

The three phases of the protocol have been simulated separately and the results shown below. In all phases, I is the initiator and R is the responder.

Fig. 1 shows the results of the registration phase. In this phase, the user is the initiator and the GWN is the responder. It is seen that U_{id}, PW_i, g, p, and C remain secret, and the claims of niagree and nisynch are verified. There are no attacks within bounds.

Scyther results : verify

Claim				Status	Comments
registration	I	registration,i8	Secret uid	Ok	No attacks within bounds.
		registration,i2	Secret pwi	Ok	No attacks within bounds.
		registration,i4	Secret g	Ok	No attacks within bounds.
		registration,i5	Secret p	Ok	No attacks within bounds.
		registration,i6	Secret h(h(uid),h(pwi))	Ok	No attacks within bounds.
		registration,I6	Niagree	Ok	No attacks within bounds.
		registration,I7	Nisynch	Ok	No attacks within bounds.
	R	registration,r8	Secret uid	Ok	No attacks within bounds.
		registration,r2	Secret pwi	Ok	No attacks within bounds.
		registration,r4	Secret g	Ok	No attacks within bounds.
		registration,r5	Secret p	Ok	No attacks within bounds.
		registration,r6	Secret h(h(uid),h(pwi))	Ok	No attacks within bounds.
		registration,R6	Niagree	Ok	No attacks within bounds.
		registration,R7	Nisynch	Ok	No attacks within bounds.

Done.

Figure 1. Scyther results of the Registration Phase.

Figure 2 shows the results of the login, verification, and authentication phase. In this phase, the user is the initiator and the GWN is the responder. It is seen that U_{id}, PW_i, g, p, and C and the nonces remain secret, and niagree and nisynch claims are verified. There are no attacks within bounds.

Scyther results : verify

Claim				Status	Comments
login	I	login,i7	Secret uid	Ok	No attacks within bounds.
		login,i2	Secret pwi	Ok	No attacks within bounds.
		login,i3	Secret R2	Ok	No attacks within bounds.
		login,i4	Secret g	Ok	No attacks within bounds.
		login,i5	Secret p	Ok	No attacks within bounds.
		login,i6	Secret h(h(uid),h(pwi))	Ok	No attacks within bounds.
		login,I7	Niagree	Ok	No attacks within bounds.
		login,I8	Nisynch	Ok	No attacks within bounds.
	R	login,r10	Secret uid	Ok	No attacks within bounds.
		login,r2	Secret pwi	Ok	No attacks within bounds.
		login,r3	Secret R1	Ok	No attacks within bounds.
		login,r4	Secret g	Ok	No attacks within bounds.
		login,r5	Secret p	Ok	No attacks within bounds.
		login,R3	Secret h(h(uid),h(pwi))	Ok	No attacks within bounds.
		login,R7	Niagree	Ok	No attacks within bounds.
		login,R8	Nisynch	Ok	No attacks within bounds.

Done.

Figure 2. Scyther results of the Login and Verification Phase.

Figure 3 shows the results of the sensor node-GWN mutual authentication phase. In this phase, the GWN is the initiator and the sensor node is the responder. It is seen that S_{id} and x_s remain secret, and niagree and nisynch claims are verified. There are no attacks within bounds.

Scyther results : verify

Claim				Status	Comments
sensorauth	I	sensorauth,i2	Secret h(Sid,h(xs))	Ok	No attacks within bounds.
		sensorauth,i3	Secret xs	Ok	No attacks within bounds.
		sensorauth,I3	Niagree	Ok	No attacks within bounds.
		sensorauth,I4	Nisynch	Ok	No attacks within bounds.
	R	sensorauth,r2	Secret h(Sid,h(xs))	Ok	No attacks within bounds.
		sensorauth,r3	Secret xs	Ok	No attacks within bounds.
		sensorauth,R3	Niagree	Ok	No attacks within bounds.
		sensorauth,R4	Nisynch	Ok	No attacks within bounds.

Done.

Figure 3. Scyther results of the Sensor-GWN mutual authentication phase.

VI. CONCLUSION

In this paper, we have proposed a security protocol using a Bluetooth enabled USB token that allows the user to use it in both connected and disconnected states. Our scheme implements two factor mutual authentication between the user and the GWN as well as mutual authentication between the GWN and the sensor nodes.

The advantages of this protocol are that it maintains user anonymity and untraceability and prevents attacks such as stolen verifier or UD, password guessing, man in the middle and user or sensor node impersonation attacks. However, this scheme does not have log file protection and does not prevent node capture attacks. These will be added in future enhancements made to the protocol.

The proposed scheme can be applied in healthcare to monitor patients, for military battleground surveillance, and in the agriculture industry to monitor crops. They can also be used for monitoring safety; in mines to monitor air purity and in nuclear power plants to monitor radioactivity levels. In conclusion, this paper achieves protection against a large number of attacks and any defects will be addressed in future enhancements.

REFERENCES

1. "Wireless sensor network", En.wikipedia.org, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Wireless_sensor_network. [Accessed: 14 May 2019].
2. K. Wong, Y. Zheng, J. Cao, and S. Wang, "A dynamic user authentication scheme for wireless sensor networks," *in Proc. IEEE International Conf. Sensor Networks, Ubiquitous, Trustworthy Computing, IEEE Computer Society*, Taichung, Taiwan, pp. 244-251, 2006.
3. M. L. Das, "Two-factor user authentication in wireless sensor networks," *IEEE Transactions on Wireless Communications*, Vol. 8, issue 3, pp. 1086–1090, 2009.
4. T.H. Chen, W.K. Shih, "A Robust Mutual Authentication Protocol for Wireless Sensor Networks," *ETRI J*, Vol.32, No. 5, pp. 704–712, 2010.
5. M. K. Khan, K. Alghathbar, "Cryptanalysis and security improvements of two-factor user authentication in wireless sensor networks," *Sensors*, 10(3):pp. 2450–2459, 2010.
6. B. Vaidya, D. Makrakis, and H. Mouftah, "Two-factor mutual authentication with key agreement in wireless sensor networks," *Security Comm. Networks*, Vol. 9, Issue 2, pp. 171–183, 2016, doi: 10.1002/sec.517.
7. J. Kim, D. Lee, W. Jeon, Y. Lee , and D. Won, "Security Analysis and Improvements of Two-Factor Mutual Authentication with Key Agreement in Wireless Sensor Networks," *Sensors*, Vol. 14, No. 12, pp. 6443–6462, Apr. 2014.
8. K.P. Xue, C.S.Ma, P.L. Hong, and R. Ding, "A temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks," *Journal of Network and Computer Applications*, Vol. 36, Issue 1, pp. 316–323, 2013.
9. M. Turkanović, B. Brumen, and M. Hölbl, "A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion," *Ad Hoc Networks*, Vol. 20, pp. 96–112, 2014.
10. R. Amin, G.P. Biswas, "A secure lightweight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks," *Ad Hoc Networks*, Vol. 20, pp. 1–23, 2015.
11. H.L. Yeh, T.H. Chen, P.C. Liu, T.H. Kim, and H.W. Wei, "A secured authentication protocol for wireless sensor networks using elliptic curves cryptography," *Sensors*, Vol. 11, pp. 4767–4779, 2011.
12. W. Shi, P. Gong, "A new user authentication protocol for wireless sensor networks using elliptic curves cryptography," *International Journal of Distributed Sensor Networks*, Vol. 9, Issue 4, 2013.
13. C.T. Li, C.Y. Weng, and C.C. Lee, "An advanced temporal credential-based security scheme with mutual authentication and key agreement for wireless sensor networks," *Sensors* 2013, Vol. 13, pp. 9589–9603.
14. Y. Park, Y. Park, "Three-Factor User Authentication and Key Agreement Using Elliptic Curve Cryptosystem in Wireless Sensor Networks", *Sensors*, Vol. 16, No. 12, pp. 2123, 2016. Available: 10.3390/s16122123.
15. Y. Choi, D. Lee, J. Kim, J. Jung, J. Nam, and D. Won, "Security Enhanced User Authentication Protocol for Wireless Sensor Networks Using Elliptic Curves Cryptography," *Sensors*, Vol. 14, No. 12, pp. 10081–10106, Jun. 2014.
16. D.B. He, N. Kumar, and N. Chilamkurti, "A secure temporal-credential-based mutual authentication and key agreement scheme with pseudo identity for wireless sensor networks," *Information Sciences*, Vol. 321, pp. 263–277, 2015.
17. C.C. Chang, H.D. Le, "A provably secure, efficient, and flexible authentication scheme for ad hoc wireless sensor networks," *IEEE Transactions on Wireless Communications*, Vol. 15, Issue 1, pp. 357–366, 2016.
18. M.S. Farash, M. Turkanović, S.Kumari, M. Hölbl, "An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the Internet of Things environment," *Ad Hoc Networks*, Vol. 36, pp. 152–176, 2016.
19. T.F. Lee, "Efficient and Secure Temporal Credential-Based Authenticated Key Agreement Using Extended Chaotic Maps for Wireless Sensor Networks," *Sensors*, Vol. 15, pp. 14960–14980, 2015.
20. I.P. Chang, T.F. Lee, T.H. Lin, C.M. Liu, "Enhanced two-factor authentication and key agreement using dynamic identities in wireless sensor networks," *Sensors* , Vol. 15, pp. 29841–29854, 2015.
21. C. Cremers, *Scyther Semantics and Verification of Security Protocols*, PhD dissertation: Eindhoven University of Technology, 2006.
22. L. Viganò, "Automated security protocol analysis with the AVISPA tool," *Electronic Notes in Theoretical Computer Science*, Vol. 155, pp. 61-86, 2006.
23. B. Blanchet, B. Smyth, *ProVerif 2.00: Automatic Cryptographic Protocol Verifier, User Manual and Tutorial*, Available at: http://prosecco.gforge.inria.fr/personal/bblanche/proverif/manual.pdf [Accessed: 14 May 2019].

AUTHORS PROFILE

Preethi Elizabeth Thomas is a Post Graduate Scholar at CHRIST (Deemed to be University), Bangalore, Karnataka 560029, India. She has done her BSc. in Computer Science, Mathematics and Statistics. Her research interests are Security in Wireless Sensor Networks, Information Security and Cryptography.

Dr. Sumitra Binu has 15 years of teaching and research experience. She has completed her PhD in Cloud Security. Her research interests include Security Analysis of Authentication Protocols, Security in Sensor Networks, Cryptography, and Block Chain Technology. Dr. Binu has many journal publications, conference presentations, and book chapters to her credit. She is a life time member of Computer Society of India(CSI).