Wireless Pers Commun (2017) 94:1881–1898
DOI 10.1007/s11277-016-3717-7

Cryptanalysis and Improvement in User Authentication and Key Agreement Scheme for Wireless Sensor Network

Akansha Singh[1] · Amit K. Awasthi[1] · Karan Singh[2]

Published online: 20 September 2016

Abstract Turkanovic et al. (Ad Hoc Netw 20:96–112, 2014) proposed a user authentication and key agreement scheme based on the notion of the 'internet of things' for wireless sensor network. Authors claimed that their scheme is safe against various attacks. We found that this scheme fails against session key recovery attack. If an attacker has stolen the smartcard, he can easily obtain the session key generated between user and sensor node. In this paper, we shows that the attacker is able to compute the secret parameter K_{GW-U_i}, which is the used by a gateway during communication with others. Now the attacker can modify the first message that was send by the user to the sensor node. Finally, he breaks the complete system. We also provide few other insecurities and vulnerability to many attacks like offline password guessing attack, replay attack and impersonate attack etc. To remedy this, an enhanced scheme is also proposed to remove the flaws of the Turkanovic et al. scheme. The result and performance analysis of our proposed scheme shows that the new enhanced scheme provides high security with low computation, communication and storage overhead.

Keywords WSN · Authentication · Key agreement · Network security

1 Introduction

Nowadays sensors are used in a wide range of applications. Wireless Sensor network (WSN) is composed of one or more gateway nodes and large number of sensor node without any wired connectivity. Sensor nodes collect the data in their surroundings and

✉ Amit K. Awasthi
awasthi.amitk@gmail.com

Akansha Singh
singhakansha1@gmail.com

Karan Singh
karancs12@gmail.com

[1] Gautam Buddha University, Greater Noida, India

[2] Jawaharlal Nehru University, Delhi, India

provide the information about changing environment parameters. As the technology is getting more and more innovative and advanced, IoT (Internet of Things) is being developed and facilitates a remote user to connect the reliable sensor nodes to collect data. User can also request any particular data by passing instructions to the sensor nodes. The main objective behind Internet of Things (IoT) is that everything is accessible and linked in a network. Every transmitted or received message should be authenticated for secure communication. Key agreement schemes are the basic building block for secure communication.

Sensor nodes are embedded with low powered battery cell so that any scheme imposed on WSN must be energy efficient. Sensor nodes are deployed in a hostile environment and exchange of the batteries is infeasible in many cases. Providing secure and authenticated communication is a challenging issue for low powered sensor nodes.

WSN is made of various types of sensor nodes and at least one sink node known as 'Gateway' node (GW). Gateway node plays an important role in WSN. It is more secure and much powerful in terms of processing, computing, communication and efficiency than the other sensor nodes.

In order to deploy an effective security system for WSN various authentication schemes have been suggested [2–9]. Cryptographic authentication can be accomplished in two ways, asymmetric and symmetric authentication. This paper focuses on symmetric authentication protocols. The rest of the paper is organized as follows: In Sect. 2, we have reviewed few cryptographic schemes exist in the literature. Section 3 contains brief review of Turkanovic et al's scheme. Section 4, describes the weaknesses of Turkanovic et al's scheme. In Sect. 5, we propose a new scheme which eliminates the weaknesses of the scheme discussed in previous section. Section 6 describes the password changing phase. In Sects. 7 and 8 we analyse security and evaluate the performance of suggested scheme respectively. Finally, Sect. 9 concludes the work.

2 Related Work

In this section we have reviewed few related schemes exist in the literature. In 2004, Watro et al. [2] proposed a security scheme named as 'TinyPK' based on asymmetric cryptography. Some researchers came to the conclusion that this scheme is prone to various attacks like man in middle attack so that the scheme is unacceptable for deployment. In 2006, Wong et al. [3] proposed a lightweight hash based user authentication scheme based on symmetric encryption. It was later found that this scheme was prone to several attacks i.e. stolen-verifier, replay, and forgery attacks. In 2009, Das [4] made an improvement of Wong et al. scheme and proposed an efficient password based user authentication using gateway node which has become a frequently cited literature in this field of password based authentication. Das scheme based on temporal credential which is released by gateway node after the verification of the user but Das's scheme does not fulfill the need of the mutual authentication and key agreement. Later on some researchers improved the Das Scheme and proposed their own schemes based on the same [5–7] to enhance the security of the original scheme.

In 2010, Khan and Alghathbar [8] also proposed several enhancements in Das scheme. The hash value of the password was used to make the password more secure. And for mutual authentication they brought up a new idea of pre shared keys between gateway and each sensor node. But it gave a new problem of extra storage overhead to GW Nodes. In

2011, Yeh et al. [9] proposed an user authentication and key agreement protocol based on ECC. But beside increasing computational complexity that also require additional storage overhead of public keys other sensor nodes.

More recently Turkanovic et al. [1] proposed a scheme for mutual authentication between the user, the sensor node and the gateway node. Our analysis shows that Turkanovic et al.'s scheme has many security issues. To find solution to these security issues we proposed a novel scheme for the security of network which resolves weaknesses of Turkanovic et al.'s scheme and is more secure, protective and efficient for real application environment. In this paper, we suggested a user authentication protocol based on symmetric key cryptography. We use only hash and XOR functions to provide mutual authentication between user, gateway and sensor node which consume less energy than public key cryptography.

Xue et al. [10] proposed five basic authentication models for WSN. In every model there are four message required implementing mutual authentication. Xue et al. used fourth model in which sensor node receives and sends three messages. Turkanovic et al. used fifth model in which sensor node have to receive and send four message. As we know that sensor node has limited communication energy we use second model. In this model, sensor node will send and receive two messages only in mutual authentication process. When a remote user needs to communicate with sensor node firstly it send a message to gateway node then GW works as a mediator and finally a session key is established in between user and the sensor node. Authentication model of proposed scheme is depicted in Fig. 1. In next section Turkanovic et al's scheme is reviewed.

3 Review of Turkanovic et al.'s Scheme

In this section, we briefly reviewed the Turkanovic et al's user authentication scheme for WSN. Turkanovic et al's protocol involves three participants, namely, the user, the gateway (GW) and the sensor node. There are three phases in Turkanovic et al's scheme:

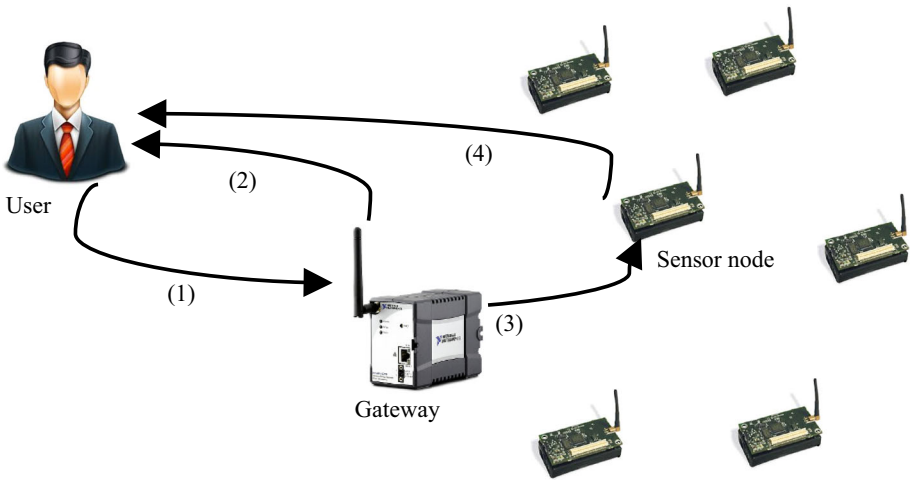

Fig. 1 Authentication model used in proposed scheme

registration, login and authentication followed by password changing phase. Notations used are listed in Table 1.

In pre-deployment phase, sensor node is loaded with its ID_{sj} and a secret password key K_{GW-S_j} shared between sensor node and gateway. Gateway has its own predefined randomly generate password key K_{GW} and stores all shared key K_{GW-S_j} of sensor node. Where $1 \leq j \leq m$ (m is the total number of sensor nodes deployed in the network).

3.1 Registration Phase

There are two registration phase in [1]. First one is between user and gateway and second is between sensor node and gateway.

3.2 Registration Between User and Gateway

User U_i has its ID_i and PW_i then selects a random number r_i. User computes $MP_i = h(r_i||PW_i)$ and $MI_i = h(r_i||ID_i)$ and sent via secure channel to GW. After receiving MP_i and MI_i, GW randomly chooses a secret password key K_{GW-U_i} for ith user. Now GW computes $f_i = h(MI_i||K_{GW})$, $x_i = h(MP_i||K_{GW-U_i})$ and $e_i = f_i \oplus x_i$. The gateway personalizes user (U_i) smartcard with $\{MI_i, e_i, f_i, K_{GW-U_i}\}$. GW stored MI_i and K_{GW-U_i} to its memory. Now the user U_i stores r_i in the smart card. Finally, smartcard has $\{r_i, MI_i, e_i, f_i, K_{GW-U_i}\}$ in its storages.

3.3 Registration Between Sensor Node and Gateway

The sensor node S_j selects a random number r_j and computes $MP_{sj} = h(K_{GW-S_j}||r_j||ID_{sj})$, $MN_{sj} = r_j \oplus K_{GW-S_j}$ and $RMP_j = MP_{sj} \oplus MN_{sj}$. After computing MP_{sj}, MN_{sj} and RMP_j sensor S_j send $\{ID_{sj}, RMP_j, MN_{sj}, T_1\}$ to gateway.T_1 is current timestamp. After receiving the message $\{ID_{sj}, RMP_j, MN_{sj}, T_1\}$ gateway node checks the validity of the timestamp and computes $MP_{sj} = RMP_j \oplus MN_{sj}$. According to received ID_{sj} gateway chooses the K_{GW-S_j}

Table 1 Notations

Symbol	Definition
U_i	ith User
SC	Smart card
S_j	jth Sensor Node
ID_i	ith User's identity
ID_{sj}	jth Sensor node's identity
PW_i	ith User's password
K_{GW}	Secure password known only to Gateway Node
K_{GW-U_i}	Secret password key shared with the user i
K_{GW-S_j}	Secure password shared with the sensor node j
T	Timestamp
SK	Separately computed session key with private information of both user and sensor node
$\oplus, \|\|, h(.)$	XOR, concatenation, a lightweight one way hash function

and computes own version of $r_j^* = MN_{sj} \oplus K_{GW-S_j}$. Also compute $sMP_{sj}^* = h(K_{GW-S_j}||r_j^*||ID_{sj})$ with the help of r_j^*. Now it checks the computed MP_{sj}^* and received MP_{sj} are equal or not. If computed MP_{sj}^* and received MP_{sj} are not equal then GW sends a rejection message to the sensor node S_j. With the use of its secret password key K_{GW} and shared password key K_{GW-S_j}, GW computes $f_{sj} = h\big(ID_{sj}||K_{GW}\big)$ and $x_{sj} = h\Big(\mathrm{MP}_{sj}^*||K_{GW-S_j}\Big)$. Finally GW computes $e_{sj} = f_{sj} \oplus x_{sj}$ and sends $\{e_{sj}, f_{sj}, T_2\}$ via insecure channel to sensor node S_j. T_2 is the current timestamp used by GW. After receiving $\{e_{sj}, f_{sj}, T_2\}$, sensor node S_j checks the validity of the timestamp T_2 and stored e_{sj} and f_{sj} into its memory.

3.4 Login Phase

After completing the registration phase the user U_i can connect to the desired sensor node S_j. For further procedure user U_i has to login first. In login phase, user U_i inserts his smart card in terminal and provides his password PW_i^* as input. With the stored r_i Smart card computes $MP_i^* = h(r_i||PW_i^*)$. With stored K_{GW-U_i}, SC computes own version of $x_i^* = h(MP_i^*||K_{GW-U_i})$ and compare this with original value of $x_i = f_i \oplus e_i$, where f_i and e_i stored in SC. If x_i and x_i^* are not equal then it rejects the login process otherwise further computes $N_i = h(x_i||K_{GW-U_i}||T_1)$. SC chooses a random number q_i and computes $Z_i = q_i \oplus f_i$ by using stored value of f_i. Finally user send an authentication message $\{MI_i, e_i, Z_i, N_i, T_1\}$ to the selected sensor node S_j via unsecure channel.

3.5 Authentication Phase

Login phase is followed by authentication phase. The purpose of this phase is to establish the secret session key between the user and the sensor node. A detail of authentication phase is given below:

- After receiving the authentication message $\{MI_i, e_i, Z_i, N_i, T_1\}$ from user, sensor node first checks the validity of timestamp. After verification with stored value of e_{sj} and f_{sj} it computes $x_{sj} = e_{sj} \oplus f_{sj}$ and $A_j = h\big(K_{GW-S_j}||T_1||T_2\big) \oplus x_{sj}$. Now, sensor node sends $\{MI_i, e_i, N_i, T_1, T_2, ID_{sj}, e_{sj}, A_j\}$ via public channel to gateway.
- Gateway checks the validity of timestamp and after verification computes own version of $f_{sj}^* = h\big(ID_{sj}||K_{GW}\big)$. Using f_{sj}^* and received value of e_{sj} compute $x_{sj}^* = e_{sj} \oplus f_{sj}^*$. Further, it computes original $x_{sj} = A_j \oplus h\big(K_{GW-S_j}||T_1||T_2\big)$ and compares own version of x_{sj}^* with original version of x_{sj}. If both are same then GW successfully authenticated the sensor node S_j, otherwise sends the rejection message to S_j.
- Now GW starts authenticating user U_i. For this purpose GW computes own version of f_i^* using its secret password key K_{GW} and received MI_i by $f_i^* = h(MI_i||K_{GW})$. Then it computes its own version of $x_i^* = e_i \oplus f_i^*$ and $Q_i = h\big(x_i^*||K_{GW-U_i}||T_1\big)$. It compares Q_i and N_i. If both are equal then gateway verified the user otherwise rejects the process.
- Now GW compute $F_{ij} = f_i^* \oplus h\Big(f_{sj}^*||K_{GW-S_j}\Big)$, $H_j = h\Big(f_{sj}^*||K_{GW-S_j}||T_1||T_2||T_3\Big)$ and $S_i = h(Q_i||T_1||T_2||T_3)$.
- GW sends a message $\{F_{ij}, H_j, S_i, T_1, T_2, T_3\}$ to sensor node S_j.
- After receiving this message S_j first checks the validity of timestamp. If validity holds then it compute $h\big(f_{sj}||K_{GW-S_j}||T_1||T_2||T_3\big)$ with stored value of f_{sj} and compares it with

received H_j. If both are equal then S_j authenticated GW otherwise sends a rejection message to GW and user.

- Now sensor node computes $f_i^* = F_{ij} \oplus h(f_{sj}||K_{GW-S_j})$ and $q_i = Z_i \oplus f_i^*$ with previously received Z_i. After that S_j chooses a random number q_j and creates a session key $SK = h(q_i \oplus q_j)$.
- S_j computes $R_{ij} = h(f_i^*||ID_{Sj}||T_1||T_2||T_3||T_4) \oplus q_j$ and sends a message $\{R_{ij}, S_i, T_1, T_2, T_3, T_4\}$ to the user.
- After receiving this message user computes $h(h(e_i \oplus f_i)||T_1)||T_1||T_2||T_3)$ and compare it with S_i. If both are equal then user computes $q_j = R_{ij} \oplus h(f_i||ID_{Sj}||T_1||T_2||T_3||T_4)$ and finally computes the session key $SK = h(q_i \oplus q_j)$.

4 Security Flaws in Turkanovic et al.'s Scheme

In this section we demonstrate that Turkanovic et al.'s scheme is susceptible to various types of attacks. Any information regarding secret key must not be revealed to fulfil the basic requirement of any authentication scheme. Turkanovic et al.'s scheme is highly insecure as the basic requirement is not fulfilled.

4.1 Smart Card Breach Attack

If smart card of the user U_i is stolen or lost then an adversary U_a can extract the secret parameters stored in the smart card by monitoring the power consumption [11]. Secret parameters stored into the smart card are $\{r_i, MI_i, e_i, f_i, K_{GW-U_i}\}$, where K_{GW-U_i} is the secret password chosen by the gateway for user U_i and stored directly into the smart card. An adversary can get this secret password key from smart card. Also an attacker can find f_i as it is also stored directly in smart card and easily can get Z_i which is send by the user U_i to sensor node S_j via insecure channel. XORing of $Z_i \oplus f_i$ will give q_i which is one part of session key. To obtain second part of session key the adversary U_a will try to get the message $\{R_{ij}, S_i, T_1, T_2, T_3, T_4\}$, which is send by the senor node via public channel, and hence compute $q_j = R_{ij} \oplus h(f_i||ID_{Sj}||T_1||T_2||T_3)$. After computing q_j, U_a can get session key $SK = h(q_i \oplus q_j)$. Revealing of session key will break the entire scheme.

4.2 Off-line Password Guessing Attack

Secret parameters stored into the smart card are $\{r_i, MI_i, e_i, f_i, K_{GW-U_i}\}$. XOR of $e_i \& f_i$ will give x_i. After reveling secret password key K_{GW-U_i}, x_i depends only on PW_i since $x_i = h(h(r_i||PW_i)||K_{GW-U_i})$. An adversary U_a can make a guess PW_i^* for password and to compute x_i^*. If $x_i^* = x_i$ holds then the adversary can get the actual password. It showsTurkanovic et al's scheme is not secure against Off-line password guessing attack.

4.3 Replay Attack

In registration phase between sensor node and gateway, sensor node sends the following parameter $\{ID_{sj}, RMP_j, MNS_j, T_1\}$ to gateway. An attacker can capture this message and sends the same message $\{ID_{sj}, RMP_j, MNS_j, T_a\}$ at different time T_a because no parameter RMP_j and MNS_j contain time stamp T_1 that leads to replay attack. Again, GW sends

$\{e_{sj}, f_{sj}, T_2\}$ to sensor node. An attacker can send the same message at different time T_a because time stamp T_2 is not used in any parameter e_{sj} and f_{sj} that leads to replay attack.

4.4 Impersonation Attack

In login phase SC sends $\{MI_i, e_i, Z_i, N_i, T_1\}$ to the selected sensor node S_j. XOR of $e_i \& f_i$ will give x_i and K_{GW-U_i} is directly stored in smart card.

- An adversary U_a at time T_a can compute $N_a = h(x_i || K_{GW-U_i} || T_a)$. Adversary U_a selects a random nonce q_a and compute $sZ_a = q_a \oplus f_i$. To impersonate the user U_i adversary sends $\{MI_i, e_i, Z_a, N_a, T_a\}$ to sensor node S_j via public channel. After receiving this message from U_a, the sensor node S_j checks the validity of time stamp, $|T_a - T_c| < \Delta T$ and after verification use the stored value of e_{sj} and f_{sj} to compute $x_{sj} = e_{sj} \oplus f_{sj}$ then computes $A_j = h\left(K_{GW-S_j} || T_a || T_2\right) \oplus x_{sj}$. Now sensor node S_j sends $\{MI_i, e_i, N_a, T_a, T_2, ID_{sj}, e_{sj}, A_j\}$ via public channel to gateway.
- Gateway checks the validity of time stamp and after verification computes own version of $f_{sj}^* = h\left(ID_{sj} || K_{GW}\right)$. Using f_{sj}^* and received value of e_{sj} it computes $x_{sj}^* = e_{sj} \oplus f_{sj}^*$. Further it computes.
- $x_{sj} = A_j \oplus h\left(K_{GW-S_j} || T_a || T_2\right)$. Compare own version of x_{sj}^* with x_{sj}. If both are same then GW successfully authenticates sensor node S_j .
- For user's authentication, GW computes $f_i^* = h(MI_i || K_{GW})$ using its secret password key K_{GW} and received MI_i. Then it computes $x_i^* = e_i \oplus f_i^*$ and $Q_i = h\left(x_i^* || K_{GW-U_i} || T_a\right)$ and compares Q_i and N_i. If both are equal then gateway verifies the user. Now GW computes $F_{ij} = f_i^* \oplus h\left(f_{sj}^* || K_{GW-S_j}\right), H_j = h\left(f_{sj}^* || K_{GW-S_j} || T_a || T_2 || T_3\right)$ and $S_i = h(Q_i || T_a || T_2 || T_3)$.
- GW Node sends a message $\{F_{ij}, H_j, S_i, T_a, T_2, T_3\}$ to the sensor node S_j.
- After receiving this message S_j first checks the validity of time stamp and then computes $H_j^* = h\left(f_{sj} || K_{GW-S_j} || T_a || T_2 || T_3\right)$ with stored value of f_{sj}. Now sensor node S_j compares it H_j^* with received H_j. If both are equal then S_j authenticates the gateway GW.
- Now sensor node computes $f_i^* = F_{ij} \oplus h\left(f_{sj} || K_{GW-S_j}\right)$ and $q_a = Z_a \oplus f_i^*$ with previously received Z_a. After that S_j chooses a random number q_j and creates a session key $SK = h(q_a \oplus q_j)$, which is not the actual session key and the whole process of authentication and key agreement is failed.

An attacker can modify very first message, that send to sensor by the user which makes this scheme insecure against impersonation attack. Due to this sensor node S_j creates a wrong session key that will destroy the entire process.

4.5 Missing Information

When user receives the last message from sensor S_j, he computes $S_i^* = h(h(e_i \oplus f_i) || T_1) || T_1 || T_2 || T_3)$ and compares it with S_i. Since $S_i = h(Q_i || T_1 || T_2 || T_3)$ where $Q_i = h\left(x_i^* || K_{GW-U_i} || T_1\right).h(h(e_i \oplus f_i) || T_1) || T_1 || T_2 || T_3)$, therefore S_i never match with S_i^*. The verification does not hold and user U_i aborts the authentication phase.

This shows that the scheme is inconsistent and does not complete authenticate process. Next section describes our proposed scheme.

5 Proposed Scheme

In this section, we propose an upgraded authentication scheme that provides high level security and based on four step model as shown in Fig. 1. In our scheme we use 1b model [10]. In this model, sensor node has to receive and send only one message and session key is established without taking gateway node into account. Our scheme resolves all the identified weaknesses of Turkanovic et al.'s scheme and is more robust and efficient for practical application environment. Scheme consists of four phases: registration phase, login phase, authentication and password changing phase.

5.1 Registration Phase

Process of registration phase will start after deployment of sensor nodes in the application area. Registration phase divided into two sub phases. First phase is between user and gateway and second one is between sensor node and the gateway. Figure 2, depicts both phases.

5.1.1 Registration Between User and Gateway

Each user has its identity (ID_i) and secure password (PW_i). User's identity and hash value of user's password is also stored in gateway. Initially gateway chooses a random key K_{GW-U} which is used for communication with user. Gateway also chose another key K_{GW-S}, which is used for communication with sensor nodes. Steps involved in this phase are as follows:

Step 1 User U_i selects a random number r_i and computes $P_i = h(r_i||h(PW_i))$.
Step 2 User generates time stamp T_{s1} and send $\{P_i, ID_i, Ts_1\}$ to gateway via a secure channel.
Step 3 After receiving the message gateway checks the validity of time stamp. If $|Ts_1 - T_c| < ||T$ holds then gateway computes.

- $\alpha_i = h(K_{GW-U}||ID_i)$,
- $b_i = \alpha_i \oplus h(P_i||h(PW_i))$
- $c_i = h(\alpha_i||h(PW_i)||ID_i)$

Step 4 Gateway personalizes smart card with $\{h(.), b_i, c_i, ID_i\}$ and sends to user via secure channel.
Step 5 User adds $d_i = r_i \oplus h(ID_i||PW_i)$ into smart card. Now smart card has the following parameter $\{h(.), b_i, c_i, d_i, ID_i\}$.

5.1.2 Registration Between Sensor Node and Gateway

Each sensor node has its identity (ID_{sj}) and secure password (PW_{sj}). Identity and hash value of password for sensor node S_j is also stored in gateway. Steps involved in this phase are as follows:

Step 1 Sensor node S_j calculates $P_{sj} = h(ID_{sj}||h(PW_{sj})||Ts_2)$ with its ID_{sj} and PW_{sj}.
Step 2 SN sends message containing the parameter $\{P_{sj}, ID_{sj,}Ts_2\}$ to gateway.

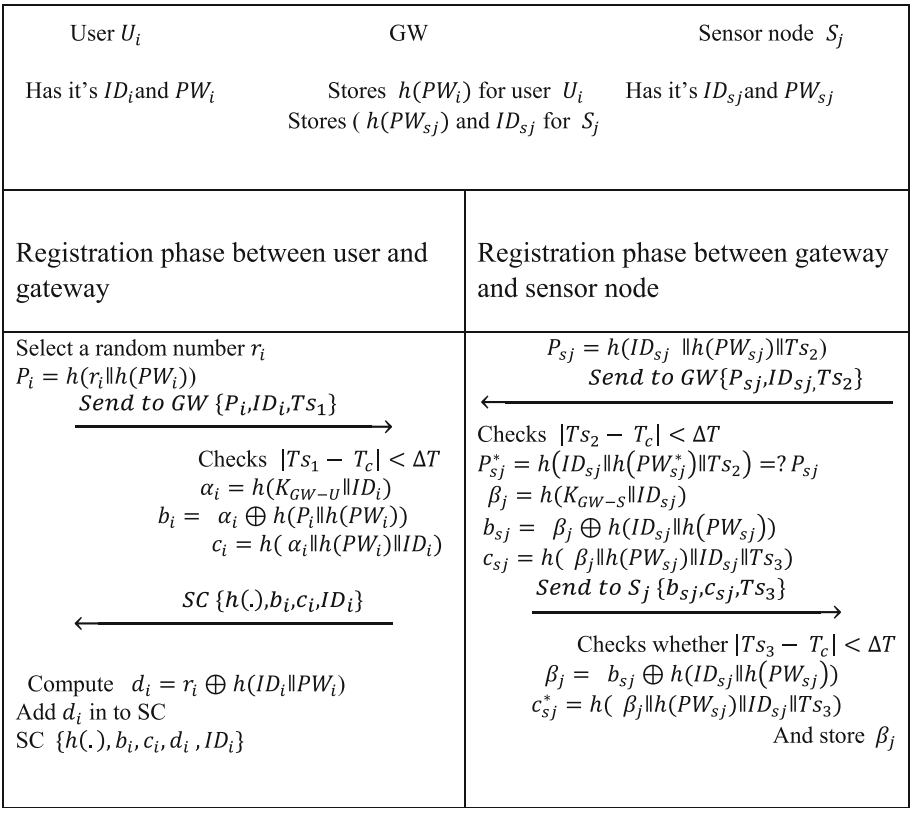

Fig. 2 Registration phase

Step 3 After receiving the message gateway checks the validity of time stamp. If $|Ts_2 - T_c| < \Delta T$ then it proceeds further otherwise send rejection message to sensor node.

Step 4 With the $ID_{sj,}$ gateway chooses its own value of $h\left(PW^*_{sj}\right)$ and computes P^*_{sj}. If P^*_{sj} is not equal to received P_{sj} then send the rejection message to sensor node otherwise perform further steps.

Step 4 With secret key K_{GW-S}, GW computes following values:

- $\beta_j = h\left(K_{GW-S} || ID_{sj}\right)$
- $b_{sj} = \beta_j \oplus h\left(ID_{sj} || h\left(PW_{sj}\right)\right)$
- $c_{sj} = h\left(\beta_j || h\left(PW_{sj}\right) || ID_{sj} || Ts_3\right)$

Step 5 GW sends $\{b_{sj}, c_{sj}, Ts_3\}$ to sensor node via public channel.

Step 6 After receiving the message sensor node checks the validity of time stamp. If $|Ts_3 - T_c| < \Delta T$ then proceeds to next step otherwise sends a rejection message to GW.

Step 7 Sensor node compute $\beta_j = b_{sj} \oplus h\left(ID_{sj} || h\left(PW_{sj}\right)\right)$ and verify $c^*_{sj} = h\left(\beta_j || h\left(PW_{sj}\right) || ID_{sj} || Ts_3\right)$ is equals to c_{sj} then store β_j into its memory otherwise sends a rejection message to GW.

5.2 Login Phase

After successful registration phase, user can connect to a sensor node through the GW node. Figure 3 shows procedure of login phase. Detailed steps are given below:

Step 1 User U_i inserts his/her smart card into terminal and input his ID_i^* and password PW_i^*.
Step 2 Smartcard computes $r_i^* = d_i \oplus h(ID_i^* || PW_i^*)$ with the stored value of d_i. Then it computes $MP_i^* = h(PW_i^*)$ and $P_i = h(r_i^* || MP_i^*)$.
Step 3 Furthermore, Smartcard computes $\alpha_i^* = b_i \oplus h(P_i || MP_i^*)$.
Step 4 He again computes $c_i^* = h(\alpha_i^* || MP_i^* || ID_i^*)$ and checks whether original c_i or computed c_i^* are equal. If they are unequal then login process will be aborted.
Step 5 If the input password was correct then user chooses a random nonce k_i and computes $M_1 = k_i \oplus h(\alpha_i || MP_i)$ and $M_2 = h(\alpha_i || MP_i || k_i || T_1)$.
Step 7 User sends $\{M_1, M_2, ID_i, T_1\}$ to GW via public channel.

5.3 Authentication and Key Agreement Phase

After successful login phase mutual authentication between all parties established in authentication and key agreement phase. There are three steps. First step is for user's legitimacy verification by GW. The second step shows the GW's legitimacy verification by user and the sensor node. Finally in third step, user verifies the legitimacy of sensor node. The target of this phase to generate session key between user and sensor node. This phase is depicted by Fig. 4. Complete phase of authentication and key agreement is as follows

Step 1 When gateway receives a message $\{M_1, M_2, ID_i, T_1\}$ from user U_i, gateway checks the validity of time stamp by computing $|T_1 - T_c| < \Delta T$. If validity holds then further computes next step otherwise sends a rejection message to the user U_i.
Step 2 Using $h(PW_i)$ according to received ID_i gateway computes $k_i^* = M_1 \oplus h(\alpha_i || h(PW_i))$ and then computes its own version of $M_2^* = h(\alpha_i || h(PW_i) || k_i^* || T_1)$ and

User U_i Has it's ID_i and PW_i	GW Stores ID_i , $h(PW_i)$ and K_{GW-U} for user U_i
Inputs ID_i^* and password PW_i^*	
SC: $r_i^* = d_i \oplus h(ID_i^* \| PW_i^*)$	
SC: Compute $MP_i = h(PW_i^*)$	
SC: $P_i = h(r_i^* \| MP_i)$	
SC: $\alpha_i^* = b_i \oplus h(P_i \| MP_i)$	
SC: $c_i^* = h(\alpha_i \| MP_i \| ID_i^*)$	
SC: checks whether $c_i = ?\ c_i^*$	
SC: Select a random nonce k_i	
SC: $M_1 = k_i \oplus h(\alpha_i \| MP_i)$	
SC: $M_2 = h(\alpha_i \| MP_i \| k_i \| T_1)$	
Send to GW $\{M_1, M_2, ID_i, T_1\}$ $\longrightarrow$	

Fig. 3 Login phase

compares it with received M_2. If both are equal then gateway authenticates the user U_i otherwise sends a rejection message to user.

Step 3 After checking the legitimacy of user, gateway computes $\gamma_{ij} = h(\alpha_i||\beta_j||ID_i||ID_{sj})$, $M_3 = \alpha_i \oplus \gamma_{ij}$ and $M_4 = h(\gamma_{ij}||M_3||ID_i||T_2)$ and send $\{M_3, M_4, ID_i, T_2\}$ to gateway where T_2 is the gateway's time stamp.

Step 4 After receiving $\{M_3, M_4, ID_i, T_2\}$, user checks whether $|T_2 - T_c| < \Delta T$ and then computes its own version of $\gamma_{ij} = \alpha_i \oplus M_3$ and $M_4^* = h(\gamma_{ij}||M_3||ID_i||T_2)$ then compare

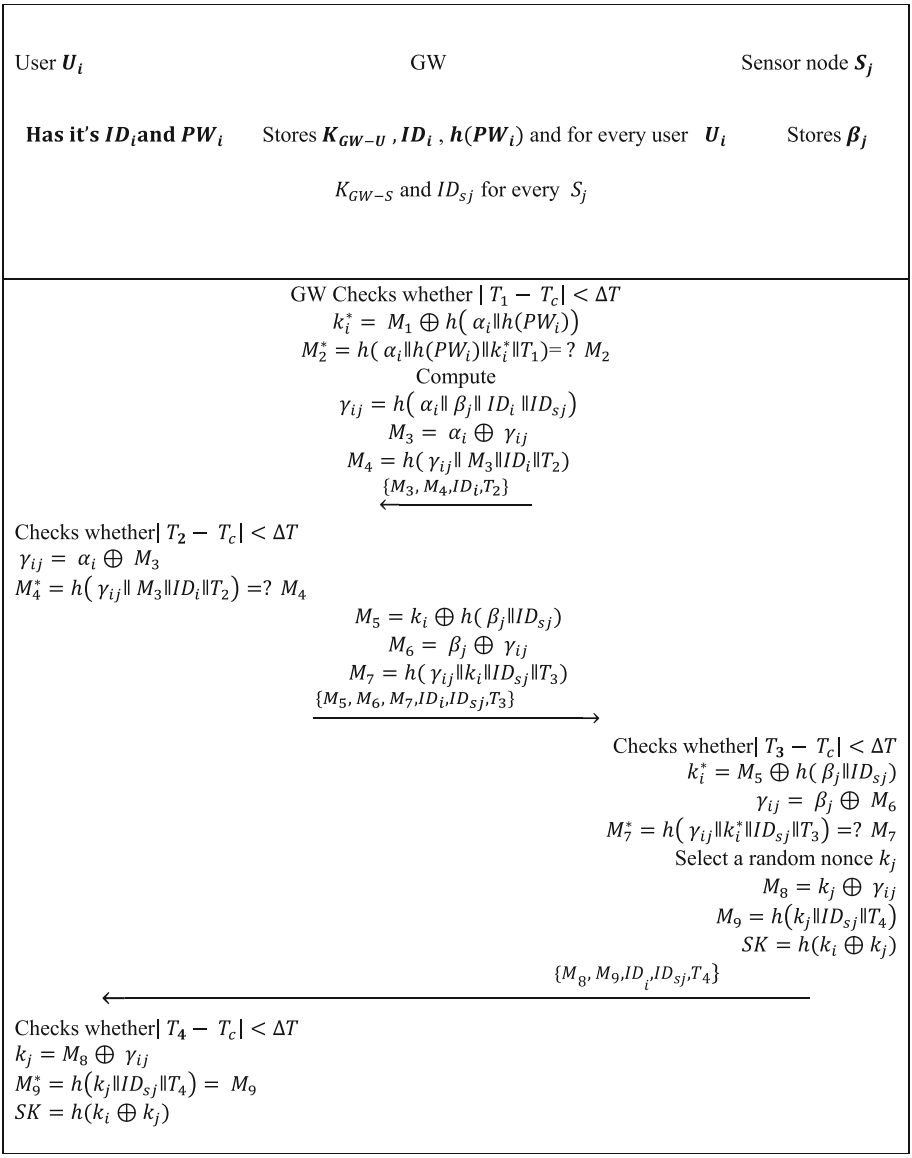

Fig. 4 Authentication phase

this result with received value of M_4. If both are equal then gateway verification by user U_i holds otherwise aborts the process by sending rejection message to GW.

Step 5 After sending a message at time T_2 to user U_i, gateway further compute $M_5 = k_i \oplus h(\beta_j||ID_{sj})$, $M_6 = \beta_j \oplus \gamma_{ij}$ and $M_7 = h(\gamma_{ij}||k_i||ID_{sj}||T_3)$ then sends $\{M_5, M_6, M_7, ID_i, ID_{sj}, T_3\}$ to sensor node S_j.

Step 6 After receiving message from gateway, sensor node checks whether $|T_3 - T_c| < \Delta T$ and then compute its own version of $k_i^* = M_5 \oplus h(\beta_j||ID_{sj})$ by using stored β_j and then compute its own version of $\gamma_{ij} = \beta_j \oplus M_6$ and $M_7^* = h(\gamma_{ij}||k_i^*||ID_{sj}||T_3)$ and compare M_7^* with received M_7. If both values are same then gateway is authenticated by sensor node otherwise sensor node sends a rejection message to gateway.

Step 7 After verification of gateway, sensor node S_j selects a random nonce k_j and compute session key as $SK = h(k_i \oplus k_j)$.

Step 8 Finally sensor node S_j computes $M_8 = k_j \oplus \gamma_{ij}$ and $M_9 = h(k_j||ID_{sj}||T_4)$ then send $\{M_8, M_9, ID_i, ID_{sj}, T_4\}$ to user U_i.

Step 9 After receiving above message from sensor node S_j user checks the validity of time stamp $|T_4 - T_c| < \Delta T$. Checks the legitimacy of sensor node by computing own version of $k_j = M_8 \oplus \gamma_{ij}$ and $M_9^* = h(k_j||ID_{sj}||T_4)$ then compare M_9^* with received M_9. If both are equal then compute session key as $SK = h(k_i \oplus k_j)$ and thus successfully end the authentication phase.

6 Password Changing Phase

This section presents the password changing phase. Details of password changing phase are as follows-

Step 1 User U_i inserts his/her smart card into terminal and input his ID_i^* and his/her old password PW_i^{OLD}.

Step 2 Now SC computes $r_i^* = d_i \oplus h(ID_i^*||PW_i^{OLD})$ with the stored value of d_i then compute $MP_i^* = h(PW_i^{OLD})$ and $P_i = h(r_i^*||MP_i^*)$.

Step 3 Furthermore SC compute $\alpha_i^* = b_i \oplus h(P_i||MP_i^*)$.

Step 4 Compute $c_i^* = h(\alpha_i^*||MP_i^*||ID_i^*)$ and checks whether original c_i or com- puted c_i^* are equal. If they are unequal then password changing process aborted otherwise continued.

Step 5 If the input password was correct then SC invite the user to select his/her new password and then SC compute new version of $P_i^{NEW} = h(r_i||PW_i^{NEW})$. After that compute $b_i^{NEW} = \alpha_i \oplus h(P_i^{NEW}||h(PW_i^{NEW})$ and

$$c_i^{NEW} = h(\alpha_i||h(PW_i^{NEW})||ID_i)$$

and $d_i^{NEW} = r_i \oplus h(ID_i||PW_i^{NEW})$. Finally replaces b_i, c_i, d_i with new values.

7 Security Analysis

In this section we analyse our protocol based on known security attacks. Security analysis of proposed scheme demonstrates that our scheme is safe from various attacks. Details are as follows.

7.1 Mutual Authentication

When a user need to communicate with the sensor node then gateway plays an important role and works as a trusted third party. In our scheme when user U_i sends a message to gateway then gateway verifies the legitimacy of user by checking $M_2^* = M_2$. Then gateway sends a message to the user and sensor node. User checks the legitimacy of the gateway by computing $M_4^* = M_4$ and sensor node verifies the legitimacy of gateway by checking $M_7^* = M_7$. Finally user verifies the sensor node's legitimacy by checking $M_9^* = M_9$. Our scheme executes mutual authentication successfully among the user, gateway and sensor node.

7.2 Key Agreement

The session key is generated at the end of authentication phase. Both parties' user and the sensor node agreed on same session key $SK = h(k_i \oplus k_j)$ where both parties individually contributed to it. k_i is the random nonce choosen by user and k_j is the random nonce choosen by sensor node. Even GW cannot compute the session key generated between user and sensor node. For secret establishment of session key we use hash and XOR operation only over an insecure open network.

7.3 Resist Stolen Smart Card Attacks

We assume if smart card has been lost or stolen from a user then a malicious attacker can get the information stored in smart card. In our scheme smart card has the following parameter $\{h(.), b_i, c_i, d_i, ID_i\}$. To proceed login phase an attacker has to insert his/her password but password is not stored directly into smart card. If he/she uses incorrect password, terminal will not verify the legitimacy of the adversary. So our protocol resists stolen smart card and smart card breach attack.

7.4 Password Protection

In our proposed scheme user password is not directly stored in smart card so adversary cannot get the information regarding password. Only user knows his password. While sending message we use hash value of password so no one can get the password and also in our proposed protocol attacker cannot get K_{GW-U} and K_{GW-S} which are used to compute α_i and β_j for user and sensor node respectively.

7.5 Resist Replay Attacks

An attacker can try to impersonate a message send by user, gateway or sensor node and can cheats by sending out a previous message. Since the message contains the sender's time stamp hence replay attack is unsuccessful for the proposed protocol.

7.6 Resist Gateway Node Bypassing Attack

In our proposed protocol it is difficult to bypass the gateway and cannot send forged message. Without right message any party cannot respond and verified any fake messages.

7.7 Password Updating/Changing

For changing a user's password, an adversary would need to submit his/her smart card. If we assume that an adversary has found the smart card by stealing or finding a lost one. Adversary must have the knowledge of old password to change the password. But it is shown in stolen smart card attack that an adversary would not be able to getting password from the smart card.

7.8 Resist Denial of Service Attack

Denial of service attack can be harmful for resource constrained wireless sensor networks. In our proposed scheme DoS attack is not possible because every time user received a confirmation or rejection message from sensor node. If number of login failures exceeds the predefined value due to bogus login attempts or due to fault of legal user or due to malicious intentions of an adversary; then card reader blocks the card for some specific period at the same time; which saves time, energy and computation resources of the server. Thus, computation exhaustive attacks like DOS attack on the server will be avoided.

7.9 Resist Insider Attack

In the proposed protocol, user U_i does not submit his password in plaintext format to the remote server while he sends hashed value $P_i = h(r_i||h(PW_i))$ to the server in a secure communication channel. Therefore, it is not possible for any privileged insider to guess both the parameters r_i as well as PW_i simultaneously in a polynomial time, which makes him unable to use the secret information of the user for his personal benefit. Hence proposed protocol prevent against the privileged insider attack.

7.10 Resist Offline Password Guessing Attack

An adversary can get the information from login request message $\{M_1, M_2, ID_i, T_1\}$, authentication message and stored security parameters of U_i 's smart card $\{h(.), b_i, c_i, d_i, ID_i\}$.. Then he tries to guess out secret parameters offline $\alpha_i, \beta_j \gamma_{ij}$ and PW_i from his directory. But every time, he has to guess at least two secret unknown parameters correctly at the same time, which is impossible. Thus because of collision resistant property of hash functions, our protocol is secure against offline guessing attacks.

Table 2 shows the comparison of our proposed scheme with existing schemes [1, 2, 5, 10].

8 Performance Evaluation

In this section we examined our scheme on the basis of parameter like communication, computation and storage overhead. We compare our scheme with related schemes and found that our proposed protocol provides more security feature without increasing too much overhead.

Table 2 Comparison of proposed scheme with related scheme in terms of security

Security requirements	Proposed scheme	Turkanovic et al.	Xue et al.	Yeh et al.	Das
Achieve mutual authentication	Yes	Yes	Yes	Yes	No
Key agreement	Yes	Yes	Yes	Yes	No
Resist stolen smart card attack	Yes	No	Yes	No	No
Password protection	Yes	Yes	Yes	Yes	No
Resist replay attack	Yes	No	Yes	No	Yes
Resist gateway node bypassing attack	Yes	Yes	Yes	Yes	No
Password changing	Yes	Yes	–	No	No
Resist Denial of service attack	Yes	Yes	–	–	–
Resist insider attack	Yes	Yes	Yes	Yes	–
Resist off line password guessing attack	Yes	No	–	–	–
Resist impersonate attack	Yes	No	Yes	Yes	–

8.1 Computational Overhead Analysis

We used T_H,T_{ECC} and $T_{\oplus}$ as three computational parameter. Where T_H denotes the time required for hash operation and T_{ECC} denotes the time required for ECC-160 operation in encryption/decryption.$T_{\oplus}$ is the time complexity for XOR operation. We have summarized our result in Table 3, which presents the comparison of our scheme with other related schemes based on the computational overhead of login and key agreement phase only. We compare our protocol with three existing schemes (Turkanovic et al.'s, Xue et al.'s and Yeh et al.). Our analysis shows that proposed scheme has less computation overhead than the others and provide security in all aspects. Yeh et al's protocol is based on ECC, which is more complex in computation and takes more energy than hash operation [12].

8.2 Storage Overhead Analysis

In proposed scheme, stored parameters in smart card are $\{h(.), b_i, c_i, d_i, ID_i\}$. We assume that the length of identity, password, random number and secret parameters are as long as the one way hash function i.e. 128 bits. Time stamps length would be 24 bit. Storage cost for user is 620 bits (5*128). Gateway needs to store identity (ID_i, ID_{sj}) and hash of password for every sensor and user. GW also keeps K_{GW-U} and K_{GW-s}. Storage analysis of sensor node should be considered due to resource constrained environment. In pre deployment phase each sensor node keeps its identity and password. Password can be removed from the memory of sensor node after registration phase and then each sensor

Table 3 Comparison of proposed scheme with related scheme in terms of computational cost

Protocol	User	GW node	Sensor node
Ours	$9T_H + 6T_{\oplus}$	$6T_H + 4T_{\oplus}$	$5T_H + 5T_{\oplus}$
Xue et al.	$10T_H + 6T_{\oplus}$	$13T_H + 6T_{\oplus}$	$6T_H + 4T_{\oplus}$
Turkanovic et al.	$7T_H + 5T_{\oplus}$	$7T_H + 4T_{\oplus}$	$5T_H + 6T_{\oplus}$
Yeh et al.	$T_H + 2T_{ECC}$	$4T_H + 4T_{ECC}$	$3T_H + 2T_{ECC}$

Table 4 Communication cost of sent messages

Message content	From –To	Cost (bytes)
M_1, M_2, ID_i, T_1	User–GW	51
M_3, M_4, ID_i, T_2	GW–User	51
$M_5, M_6, M_7, ID_i, ID_{sj}, T_3$	GW–Sensor node	83
$M_8, M_9, ID_i, ID_{sj}, T_4$	Sensor node –User	67

node need to store its identity and β_j only. After registration phase storage cost for sensor node is 256 bits while in turkanovic et al' scheme storage cost for sensor node is 448 bits. In proposed scheme storage cost of a sensor node is less than the turkanovic et al' scheme.

8.3 Communication Overhead Analysis

Sensor node is the device with low memory, limited computation capacity and low transmission range. The IEEE 802.15.4 [13, 14] standard was specially designed for low cost and low speed communication between devices. According to IEEE 802.15.4 standard, packet frames less than 127 bytes fit to WSN. In our protocol only four message are required for mutual authentication and to establish session key between user and sensor node. Cost of first message sent from user to gateway is 51 bytes. Second message from GW to user has the communication cost of 51 bytes. Third message is sent from GW to sensor node has the communication cost 83 bytes. Communication cost of fourth message from sensor node to user is 67 bytes. As per the standard, threshold value is 127 bytes and it is shown that every message exchanged between the user, GW and sensor node is less than threshold value of 127 bytes.

Communication cost of the four messages used in authentication phase of proposed scheme is given in Table 4.

9 Conclusion

In this paper a user authentication and key agreement scheme is presented. Turkanovic et al. proposed a user authentication and key agreement scheme based on internet of things notion for wireless sensor network. Crypt analysis of their scheme shows the scheme proposed by them is not secure against many types of attacks. We found that their scheme is insecure for practical application and susceptible to offline password guessing attack, replay attack and impersonate attack. Furthermore, an outsider can obtain the secured password shared between the user and gateway that breaks the entire system. To eliminate all security shortcomings, we proposed a new and improved user authentication and key agreement scheme. Security analysis of our scheme shows that it resolves all the described vulnerability of Turkanovic et al.'s scheme and is more protected and competent for practical application surroundings. The performance analysis of our scheme shows that our scheme requires only 1 more hash computation as compared to turkanovic et al's scheme. The additional hash computation is done by user's side.

References

1. Turkanović, M., Brumen, B., & Hölbl, M. (2014). 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, 20*, 96–112.
2. Watro, R., Kong, D., Cuti, S.-F., Gardiner, C., Lynn, C., & Kruus, P. (2004). Tiny PK: Securing sensor networks with public key technology. In *Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks* (pp. 59–64). Washington: ACM.
3. Wong, K. H. M., Zheng, Y., Cao, J., & Wang, S. (2006). A dynamic user authentication scheme for wireless sensor networks. In *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, vol. 1, (SUTC'06)*. (Vol. 01, pp. 244–251). IEEE Computer Society.
4. Das, M. L. (2009). Two-factor user authentication in wireless sensor networks. *IEEE Transactions on Wireless Communications, 8*, 1086–1090.
5. Huang, H. -F., Chang, Y. -F., & Liu, C.-H. (2010). Enhancement of two-factor user authentication in wireless sensor networks. In *Proceedings of the 2010 Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing* (pp. 27–30). IEEE Computer Society.
6. He, D., Gao, Y., Chan, S., Chen, C., & Bu, J. (2010). An enhanced two-factor user authentication scheme in wireless sensor networks. *Ad Hoc and Sensor Wireless Networks, 10*, 361–371.
7. Nyang, D., & Lee, M.-K. (2009). Improvement of Das's two-factor authentication protocol in wireless sensor networks. In *CORD Conference Proceedings, 2009*.
8. Khan, M. K., & Alghathbar, K. (2010). Cryptanalysis and security improvements of 'two-factor user authentication in wireless sensor networks'. *Sensors, 10*, 2450–2459.
9. Yeh, H. L., Chen, T. H., Liu, P. C., Kim, T. H., & Wei, H. W. (2011). A secured authentication protocol for wireless sensor networks using elliptic curves cryptography. *Sensors, 11*, 4767–4779.
10. Xue, K., Ma, C., Hong, P., & Ding, R. (2013). A temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks. *Journal of Network and Computer Applications, 36*, 316–323.
11. Messerges, T. S., Ezzat, A. D., & Robert, H. S. (2002). Examining smart-card security under the threat of power analysis attacks. *IEEE Transactions on Computers*, *51*(5), 541–552.
12. Potlapally, N. R., Ravi, S., Raghunathan, A., & Jha, N. K. (2003). Analyzing the energy consumption of security protocols. In *ISLPED'03, August 25–27, 2003, Seoul, Korea*.
13. Adams, J. T. (2006). An introduction to IEEE STD 802.15.4. In *IEEE Aerospace conference, Big Sky, MT*.
14. Iqbal, M. S., & Al- Raweshidy, H. S. (2013). Performance evaluation of IEEE 802.15.4 standard for low data rate ad hoc wireless sensor networks. In *2013 International Conference on Control, Automation and Information Sciences (ICCAIS)* (pp. 300–304).

Akansha Singh is a research scholar at Gautam Buddha University, India. She pursuing her Ph.D. on the topic "Modelling of Energy Efficient Secure Routing Protocol for Wireless Sensor Networks". She received the M.Sc. degree in mathematics from C.C.S. University, India. Her research interests include Cryptography, Wireless sensor networks, Network security and Network Routing.

Dr. Amit K. Awasthi did his masters (M.Sc. in Mathematics) from Bareilly College, Bareilly (MJP Rohilkhand University) and completed his Doctorate from Dr BR Ambedkar University (Formerly Agra University), India under the supervisions of Prof. Sundar Lal, Vice-chancellor, VBS Purvanchal University, Jaunpur. He has published more than 25 papers in international journal/conference. His current research interests include Cryptography, Information Security, Algorithms, Numerical Methods etc. He is Editorial Board Member and Reviewer of many International Journals. He is a life member of Cryptology Research Society of India, Computer Society of India and Indian Mathematical Society of India.

Dr. Karan Singh received the Engineering degree (Computer Science & Engineering) and M.Tech. (Computer Science & Engineering) from Kamala Nehru Institute of Technology, India. He has also completed his Ph.D. (Computer Science & Engineering) from Motilal Nehru National Institute of Technology India. He worked at Gautam Buddha University, India. Currently, he is working with School of Computer & Systems Sciences, Jawaharlal Nehru University, New Delhi. His primary research interests are in Computer network, Network security, Multicast communication, IoT, and Software define the network. He is reviewer of Springer, Taylor & Francis, Elsevier Journals and IEEE Transactions. He is an Editorial Board Member of Journal of Communications and Network (CN), USA. He published many research papers in refereed journals and good conferences. He had organized the workshops, conference Sessions and trainings. Dr. Singh worked as General Chair of the international conference (Qshine 2013) at Gautam Buddha University, India. Recently he organized a workshop on "PYTHON", short term Course (STC) at JNU and special session in ICGCET 2016 at Denmark. He has been nominated for "Who's who" in World in the year 2008. Dr. Singh has been joined as a Professional member of ACM, New York, CSTA U.S.A, CSI, Secunderabad, India, CRSI, Kolkata, IEEE, USA, IACSIT, Singapore, ICST, IAENG, Hong Kong, ACEEE, India, ISOC, USA and AIRCC.