Journal of Network and Computer Applications 35 (2012) 1646–1656

Contents lists available at SciVerse ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

A dynamic password-based user authentication scheme for hierarchical wireless sensor networks

Ashok Kumar Das [a,*], Pranay Sharma [a], Santanu Chatterjee [b], Jamuna Kanta Sing [c]

[a] Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad 500 032, India
[b] Research Center Imarat (RCI), Defence Research and Development Organization, Hyderabad 500 069, India
[c] Department of Computer Science and Engineering, Jadavpur University, Kolkata 700 032, India

ARTICLE INFO

Article history:
Received 14 April 2011
Received in revised form
9 October 2011
Accepted 22 March 2012
Available online 3 April 2012

Keywords:
Wireless sensor networks
User authentication
Wireless security
Passwords
Hash function
Smart cards

ABSTRACT

Most queries in wireless sensor network (WSN) applications are issued at the point of the base station or gateway node of the network. However, for critical applications of WSNs there is a great need to access the real-time data inside the WSN from the nodes, because the real-time data may no longer be accessed through the base station only. So, the real-time data can be given access directly to the external users (parties) those who are authorized to access data as and when they demand. The user authentication plays a vital role for this purpose. In this paper, we propose a new password-based user authentication scheme in hierarchical wireless sensor networks. Our proposed scheme achieves better security and efficiency as compared to those for other existing password-based approaches. In addition, our scheme has merit to change dynamically the user's password locally without the help of the base station or gateway node. Furthermore, our scheme supports dynamic nodes addition after the initial deployment of nodes in the existing sensor network.

1. Introduction

In a wireless sensor network (WSN), a large number of sensor nodes are deployed in a target field (also called a deployment field). After deployment of nodes, they form adhoc infrastructure-less wireless network. Nodes communicate with each other using wireless communication within their communication ranges and data are routed to the nearby base station(s) via multi-hop communication path. WSNs have numerous applications in field of military (for example, battlefield surveillance), hospitals and other real-time applications. Often WSNs are easy to deploy in a given area because nodes can be deployed randomly in the field and do not require constant maintenance. A large number of nodes could be dropped on a particular area from truck/plane and there after each node coordinates with their neighboring nodes and together they form a network which is finally linked to a base station. Information gathered from the area of deployment is then passed on to the base station. A survey on WSNs can be found in Akyildiz et al. (2002).

Consider the scenario of battlefield surveillance which is one of the major military applications. A large number of sensor nodes are rapidly deployed in a battlefield via airplanes or trucks. Each individual sensor node monitors conditions and activities in its surroundings after deployment in the battlefield and then reports these sensing observations to the base station via wireless communications through its neighboring sensor nodes. The base station then can conduct a more accurate detection on the activities (for example, possible attacks) of the opposing force after collecting a large number of sensing observations from the sensor nodes. Thus, the appropriate decisions as well as responses can be made quickly in the battlefield. In such an application, nodes in WSN need to transmit critical data immediately. Usually, the information from nodes are gathered periodically in the *BS* and so, the gathered information may not be real-time. If the base station waits for a read cycle, the information gathered from the nodes may not be real-time and as a result, appropriate decisions could not make quickly. As most critical applications of WSNs are real-time based, users are generally interested in accessing real-time information. This is possible if the users (called the external parties) are allowed to access the real-time data directly from the nodes inside WSN and not from base station as and when they demand. In order to get the real-time information from the nodes, the user needs to be first authorized to the nodes as well as the *BS* so that illegal access to nodes do not happen. This paper aims to propose a new user authentication scheme based on traditional passwords of users to provide user access to real-time data by authorizing him/her directly at node level and also making it

* Corresponding author. Tel.: +91 40 6653 1506; fax: +91 40 6653 1413.
E-mail addresses: iitkgp.akdas@gmail.com, ashok.das@iiit.ac.in (A.K. Das), pranaysharma08@gmail.com (P. Sharma), santanu.chatterjee@rcilab.in (S. Chatterjee), jksing@ieee.org (J.K. Sing).

1084-8045/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.jnca.2012.03.011

possible for users to communicate with the nodes in order to have responses to their queries.

1.1. Network model

In hierarchical wireless sensor network (HWSN) (shown in Fig. 1), there is a hierarchy among the nodes based on their capabilities: *base station*, *cluster heads* and *sensor nodes*. Usually, the sensor nodes are inexpensive, limited capabilities and generic wireless devices. Sensor node is equipped with limited battery power, memory size and data processing capability and short radio transmission range. Sensor nodes in a cluster communicate among each other in that cluster and finally communicate with the cluster head (CH). The cluster heads are more resource rich than traditional sensors. They are equipped with high power batteries, larger memory storage, powerful antenna and data processing capabilities. The cluster heads can execute relatively complicated numerical operations than the sensor nodes as well as they have much larger radio transmission range. The cluster heads can communicate with each other directly and relay data between its cluster members and the base station. A base station (BS) or gateway node (GW) is typically a gateway to another network, a powerful data processing/storage center, or also an access point for human interface.

We consider the HWSN model for developing our proposed scheme due to the following reasons (Cheng and Agrawal, 2007). Wireless sensor networks are distributed event-driven systems that differ from traditional wireless networks in several ways, for examples, extremely large network size, severe energy constraints, redundant low-rate data, and many-to-one flows. In many sensing applications, connectivity between all sensor nodes is not always necessary. As a result, data centric mechanisms can be performed to aggregate redundant data in order to reduce the energy consumption and traffic load in wireless sensor networks, and thus HWSN has more operational advantages than the distributed WSN model (DWSN) for wireless sensor networks because of inherent limitations of sensors on power and processing capabilities.

1.2. Our contributions

In this paper, we propose a new password-based user authentication scheme in large-scale hierarchical wireless sensor networks. Our scheme has the following attractive properties:

- It provides better security as compared with the other related schemes, since it supports mutual authentication between the user and the cluster heads, resists denial-of-serive attack, privileged-insider attack, smart card breach attack and node capture attack.
- It supports dynamic node addition after initial deployment of nodes in the network. The proposed scheme does not require to update information for new nodes addition in the user's smart card.
- It supports changing the user's password locally without the help of the *BS*.
- It provides unconditional security against node capture attacks. That is, compromise of a cluster head does not reveal any secret information of other cluster heads and it does not lead to compromise any other secure communication between the user and the non-compromised nodes in the network.
- It establishes a secret session key between the user and a cluster head for future secret communication of the real-time data inside WSN between them using the established session key.

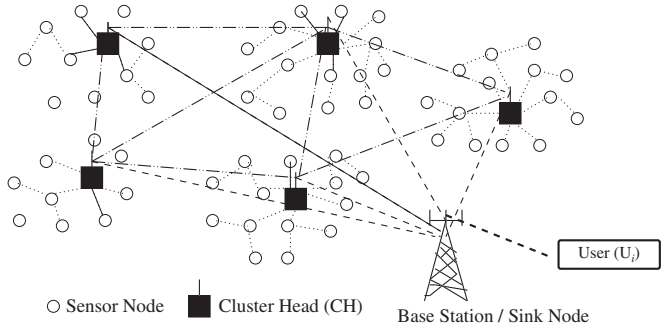

Fig. 1. A hierarchical wireless sensor network (HWSN) architecture.

1.3. Organization of the paper

The rest of this paper is organized as follows. In Section 2, we review the existing related works on user authentication in WSNs. In Section 3, we propose a novel dynamic password-based user authentication scheme in HWSNs. Section 4 analyzes the security properties of our proposed scheme and in Section 5 we compare the performances of our scheme with the existing related schemes. Finally, we conclude the paper in Section 6.

2. Related work

In this section, we discuss in brief the existing related user authentication schemes applicable in resource-constrained WSN environment.

Watro et al. (2004) proposed a user authentication scheme for WSNs based on public-key cryptographic protocols, called TinyPK protocol. TinyPK uses the RSA protocol (Rivest et al., 1978) and Diffie-Hellman protocol (Diffie and Hellman, 1976). However, TinyPK is vulnerable to the following attack, which is described in Das (2009), Yuan et al. (2010). On receiving the user's public key, an attacker can easily encrypt a session key along with other parameters and send the encrypted message to the user. After receiving the encrypted message, the user believes that the message has come from the sensor node. The user thus decrypts the receiving encrypted message using his/her private key and also uses the session key for subsequent operations the attacker intends to perform. Wong et al. (2006) proposed a user authentication scheme based on user's password which uses only the efficient hash function. However, there are security flaws in their scheme (Das, 2009; Yuan et al., 2010). One security flaw is that their scheme does not resist many logged in users with the same login-id threat in which if an attacker possesses a valid user's password, he/she can easily login to the sensor network. Moreover, their protocol suffers from stolen-verifier attack, because both the GW-node and login-node maintain the lookup table of the registered user's secret information.

M.L. Das proposed an efficient scheme (Das, 2009) based on passwords which improves security over Wong et al.'s scheme. The scheme uses timestamp for verification. However, it cannot resist denial-of-service attack and node compromise attack. Another drawback of this scheme is that a user cannot securely and freely change its password.

Later, (Khan and Alghathbar, 2010) showed that M.L. Das's scheme (Das, 2009) is insecure. Khan and Alghathbar showed that M.L. Das's scheme is insecure against GW-node bypassing attack due to the following reason. In M.L. Das's scheme, when the verification occurs successfully and the login request message of a user U_i is accepted, the GW-node (base station) sends an intimation message $\langle DID_i, A_i, T' \rangle$ to some nearest sensor node, say S_n in

order to inform about the successful login of U_i and request S_n to respond the query/data of U_i. In this scheme, DID_i, known as dynamic ID of user U_i, and A_i are calculated as $DID_i = h(ID_i \| PW_i) \oplus h(x_a \| T)$ and $A_i = h(DID_i \| S_n \| x_a \| T')$, where T is the current timestamp of U_i's system, ID_i,PW_i are the identity and password of U_i, x_a is a secret parameter which is known to the GW-node, all sensor nodes and it is also stored in U_i's smart card, and T' is the timestamp of GW-node. The value of x_a is used to ensure S_n that message $\langle DID_i, A_i, T' \rangle$ comes from the legitimate GW-node. However, if the value of x_a is extracted from the smart card of U_i or by capturing the sensor node S_n or any other sensor node in the network (as pointed out in Khan and Alghathbar (2010)), then the user U_i or any attacker can login S_n without going through the GW-node as follows. The attacker or U_i himself/herself computes a fake dynamic identity, say $DID_f = h(ID_f \| PW_f) \oplus h(x_a \| T_f)$, where ID_f,PW_f, and T_f are the fake ID of attacker, a randomly chosen fake password and the current timestamp of attacker's machine. After computing $A_f = h(DID_f \| S_n \| x_a \| T_f)$, the attacker sends the message $\langle DID_f, A_f, T_f \rangle$ to S_n over insecure channel. When S_n receives the message, S_n first validates whether $(T^* - T_f) \leq \triangle T$, where T^* is the current time of S_n. If this is valid, S_n then computes $A'_f = h(DID_f \| S_n \| x_a \| T_f)$ and checks if A'_f is equal to A_f received in the message. If it holds, S_n responds to the attacker's query and hence, being an illegal user of the sensor network, the attacker still enjoys the resources as an authorized user without being a member of the system.

Further, Khan and Alghathbar showed that M.L. Das's scheme is insecure against privileged-insider attack due to the following reason. In M.L. Das's scheme, the user U_i performs registration phase with GW-node by presenting his/her password PW_i in plaintext format. Hence, if the system manager or a privileged-insider of the GW-node knows the password of U_i, he/she may try to impersonate U_i by accessing other servers where U_i could be also a registered user because the user U_i may use the same password to access different applications or servers for his/her convenience of remembering long password and easy-to-use whenever required. The improvement and enhancement of M.L. Das's scheme have been proposed in Huang et al. (2010), Nyang and Lee (2009).

He et al. (2010) proposed an enhanced scheme based on M.L. Das's scheme (Das, 2009). Their scheme keeps the merits of the original protocol and can withstand the security weaknesses such as vulnerabilities to an insider attack and to an impersonation attack. Vaidya et al. (2010) showed that M.L. Das's scheme (Das, 2009) and Khan-Alghathbar's scheme (Khan and Alghathbar, 2010) have flaws and remain vulnerable to various attacks including stolen smart card attacks. In order to overcome security weaknesses of both schemes (Das, 2009; Khan and Alghathbar, 2010), they proposed an improved two-factor user authentication which is resilient to stolen smart card attacks as well as other common type of attacks. Fan et al. (2010) proposed a simple user authentication scheme, which is efficient and Denial-of-Service (DoS) resistant user authentication scheme for two-tiered WSNs. Their scheme can establish a session key between the user and a master node (cluster node) in the sensor network. Chen and Shih (2010) pointed out that M.L. Das's scheme (Das, 2009) fails to achieve mutual authentication. To tackle such a problem, they proposed a robust mutual authentication protocol for WSNs.

Recently, biometric-based user authentication in WSNs has drawn some research attention. A biometric-based user authentication scheme for WSNs has been proposed by Yuan et al. (2010). It uses very similar concept as in M.L. Das's scheme. As in Das (2009), Yuan et al.'s scheme does not offer any protection against denial-of-service attack because the GW-node does not expect any acknowledgement in their protocol. Moreover, the GW-node and the sensor nodes will not know about the message if an attacker blocks it from reaching the nodes. Further, their scheme is not resilient against node compromise attack. However, it supports freely changing password locally by the user without contacting the GW-node in the network as compared to other schemes (Das, 2009; Watro et al., 2004; Wong et al., 2006).

3. The proposed scheme

In this section, we first discuss the threat model used in our protocol. We then give the list of notations used in our proposed scheme. Finally, we describe the different phases related to our scheme.

3.1. Threat model

Due to the hostile environments in the deployment field, nodes can be physically captured by an attacker. We assume that both the sensor nodes as well as cluster heads can be compromised or captured by an attacker. Usually, nodes are not equipped with tamper-resistant hardware due to cost constraints and hence we assume that once a node is captured by an attacker, all the stored sensitive data as well as cryptographic information are revealed to the attacker. However, we assume that in any case, the base station (BS) will not be compromised by an attacker. Finally, as in Das (2009), we make use of the famous Dolev–Yao threat model (Dolev and Yao, 1983) in which two communicating parties (nodes) communicate over an insecure channel. We adopt the similar threat model for WSNs where the channel is insecure and the end-points (users, sensor nodes, cluster heads) cannot in general be trustworthy.

Eschenauer and Gligor proposed a centralized node revocation method (Eschenauer and Gligor, 2002), in which when the base station detects a misbehaving node, it broadcasts a message to revoke that node. A localized mechanism for sensor network node revocation was further proposed by Chan et al. (2003) and in their approach, nodes can revoke their neighbors. The sybil attack in sensor network has been analyzed and described by Newsome et al. (2004). Further, a mechanism for distributed detection of node replication attacks in sensor networks was proposed by in Parno et al. (2005). Zhu et al. proposed an approach (Zhu et al., 2010) that combines deterministic mapping (to reduce communication and storage costs) with randomization (to increase the level of resilience to node compromise). Their approach performs better than Parno et al.'s approach (Parno et al., 2005). We thus assume that the compromised (captured) nodes can be detected and as a result, the base station, cluster head and sensor nodes in each cluster know the ids of the compromised nodes. Consequently, the base station alerts the users with the compromised cluster heads in the network.

3.2. Notations

We use the notations in this paper to describe our proposed scheme in Table 1.

A one-way hash function $h : \{0,1\}^* \rightarrow \{0,1\}^l$ takes an arbitrary-length input $X \in \{0,1\}^*$, and produces a fixed-length (say, l-bits) output $h(X) \in \{0,1\}^l$, called the message digest. The hash function is the fingerprint of a file, a message, or other data blocks, and has the following attributes (Stallings, 2004).

- h can be applied to a data block of all sizes.
- For any given variable X, $h(X)$ is easy to operate, enabling easy implementation in software and hardware.
- The output length of $h(X)$ is fixed.
- Deriving X from the given value $Y = h(X)$ and the given hash function $h(\cdot)$ is computationally infeasible.

Table 1
Notations used in this paper.

Symbol	Description
U_i	User
BS	Base station
S_j	Sensor node
CH_j	Cluster head in the j-th cluster
PW_i	Password of user U_i
ID_i	Identity of user U_i
ID_{CH_j}	Identifier of cluster head CH_j
$h(\cdot)$	A secure one-way hash function
E	Symmetric key encryption algorithm
D	Symmetric key decryption algorithm
X_s	A secret information maintained by the base station
X_A	A secret information shared between user and base station
y	A secret random number only known to user
T	Timestamp
$A\|\|B$	Data A concatenates with data B
$A \oplus B$	XOR operation of A and B

- For any given variable X, finding any $Y \neq X$ so that $h(Y) = h(X)$ is computationally infeasible.
- Finding a pair of inputs (X,Y), $X \neq Y$, so that $h(X) = h(Y)$ is computationally infeasible.

The hash functions have many applications in the field of cryptology and information security, notably in digital signatures, message authentication codes (MACs), and other forms of authentication and thus it becomes the basis of many cryptographic protocols. One of the fundamental properties of hash functions is that the outputs are very sensitive to small perturbations in their inputs. We use $SHA-1$ as the secure hash algorithm (Secure hash standard, 1995). For efficient encryption/decryption, we use the AES algorithm (Advanced Encryption Standard).

3.3. Description of the proposed scheme

In this section, we discuss our proposed user authentication scheme. Our scheme consists of the following phases: pre-deployment phase, post-deployment phase, registration phase, login phase, authentication phase, password change phase and dynamic node addition phase.

We consider a hierarchical wireless sensor network (HWSN) (Das, 2009; Das and Sengupta, 2008) consisting of two types of sensors: a small number of powerful High-end sensors (H-sensors) and a large number of resource-constrained Low-end sensors (L-sensors). The H-sensors can execute relatively complicated numerical operations than the L-sensors. The H-sensors have much larger radio transmission range and also larger storage space than the L-sensor nodes. The L-sensors are extremely resource-constrained. For example, the H-sensors can be PDAs and the L-sensors are the MICA2-DOT motes (Crossbow Technology Inc, 2011). Since MICA2 motes are now obsolete devices, one can use the L-sensors as MICAz/IRIS sensor devices (Crossbow Technology Inc, 2011). Dong and Liu (2007) used the TelosB devices (Crossbow Technology Inc, 2011), which have 1 MB flash memory, as assisting nodes in the network. The assisting node act as cluster heads to facilitate pairwise key establishment between sensor nodes in the network. Further, one can also use the SunSPOT (Sun Small Programmable Object Technology) devices (Sun SPOT, 2011) as cluster heads. The SunSPOT device is built upon the IEEE 802.15.4 standard and it supports the IEEE 802.15.4 MAC layer, on top of which e.g. (ZigBee, 2011) can be built. After deployment of cluster heads and sensor nodes, the sensor nodes communicate among their neighbor sensor nodes in a cluster. The sensor nodes also communicate with the neighbor cluster head in that cluster. The cluster heads communicate among each other and also with the base station. All these communication take place using the IEEE 802.15.4 standard.

The target field is considered as two dimensional and is partitioned into a number m of equal sized disjoint clusters. Each cluster consists of a cluster head CH_j (here it is an H-sensor node) and a number n_i of L-sensor nodes. For our sake of simplicity, we call an L-sensor node as a regular sensor node and an H-sensor node as a cluster head (CH). The number n_i of regular sensor nodes is to be taken in each deployment cluster so that the network connectivity in each cluster is high so that every sensor node can communicate securely among each other and finally with their neighbor cluster head in that cluster. The sensors are to be deployed randomly in a cluster and each cluster head is deployed in that cluster around the center of that cluster. The base station (BS) can be located either in the center or at a corner of the network.

3.3.1. Pre-deployment phase

The (key) setup server (the base station) performs the following steps in offline before deployment of the sensor nodes and cluster heads in a target field (deployment field):

- *Step* 1: The setup server assigns a unique identifier, say ID_{CH_j} to each cluster head CH_j which will be deployed in the target field. For each deployed regular sensor node S_i, the setup server also assigns a unique identifier, say ID_{S_i}.
- *Step* 2: The setup server then selects randomly a unique master key, say MK_{CH_j} for each cluster head CH_j. Note that the master key is shared between the cluster head CH_j and the base station only. Similarly, the setup server also assigns a unique randomly generated master key, say MK_{S_i} for each deployed regular sensor node S_i, which will be shared with the base station only.
- *Step* 3: Finally, the setup server loads the following information into the memory of each cluster head CH_j ($j = 1, 2, \ldots, m$): (i) its own identifier, ID_{CH_j} and (ii) its own master key MK_{CH_j}. Each deployed regular sensor node S_i in the cluster C_j is loaded with the following information: (i) its own identifier, ID_{S_i} and (ii) its own master key MK_{S_i}.

3.3.2. Post-deployment phase

As soon as regular sensor nodes are deployed randomly in their respective clusters, their task is to locate the physical neighbors within their communication ranges. Cluster heads in their own clusters locate their physical neighbors which are the regular sensor nodes. Cluster heads also locate their other cluster heads in their communication ranges in the network.

For secure communication between regular sensor nodes, and between regular sensor nodes and cluster head in a cluster, nodes require to establish pairwise secret keys between them. Since our main goal in this paper is user authentication, so we assume that nodes in a cluster can establish secret keys using some existing efficient and secure key establishment techniques. For example, we use the unconditionally secure key establishment scheme (Das and Sengupta, 2008) for pairwise key establishment between nodes in each cluster and between cluster heads in the network.

After key establishment, sensor nodes can securely communicate with other neighbor sensor nodes and their cluster head in the cluster. Cluster heads can also securely communicate with other neighbor cluster heads and finally to the base station.

3.3.3. Registration phase

When the remote user authentication scheme starts, the user U_i and the base station BS need to perform the following steps:

- *Step* 1: The user U_i selects a random number y, the identifier ID_i and the password PW_i. U_i then computes $RPW_i = h(y\|PW_i)$. U_i provides the computed masked password RPW_i and ID_i to the base station via a secure channel.
- *Step* 2: The *BS* computes $f_i = h(ID_i\|X_s)$, $x = h(RPW_i\|X_A)$, $r_i = h(y\|x)$, and $e_i = f_i \oplus x = h(ID_i\|X_s) \oplus h(RPW_i\|X_A)$. Note that the secret information X_s is only known to the *BS*. The secret information X_A is shared between U_i and the *BS*.
- *Step* 3: The *BS* then selects all m deployed cluster heads in the network, $CH_1, CH_2, \ldots, CH_m$, which will be deployed during the initial deployment phase, and computes the m key-plus-id combinations $\{(K_j, ID_{CH_j}) \mid 1 \le j \le m\}$, where $K_j = E_{MK_{CH_j}}(ID_i\| ID_{CH_j}\|X_s)$.
- *Step* 4: For dynamic cluster head addition phase, assume that another m' cluster heads, CH_{m+1}, CH_{m+2}, …, $CH_{m+m'}$, which will be deployed later after the initial deployment in the network in order to replace some compromised cluster heads, if any, and add some fresh cluster heads along with sensor nodes.
  For this purpose, the *BS* computes another m' key-plus-id combinations $\{(K_{m+j}, ID_{CH_{m+j}}) \mid 1 \le j \le m'\}$, where $K_{m+j} = E_{MK_{CH_{m+j}}}(ID_i\|ID_{CH_{m+j}}\|X_s)$. $ID_{CH_{m+j}}$ is the unique identifier generated by the *BS* for the cluster head CH_{m+j} to be deployed during the dynamic node addition phase and $MK_{CH_{m+j}}$ the unique master key randomly generated by the *BS* for CH_{m+j}, which is shared between it and the *BS*.
- *Step* 5: Finally, the *BS* generates a tamper-proof smart card with the following parameters: (i) ID_i, (ii) y, (iii) X_A, (iv) r_i, (v) e_i, (vi) $h(\cdot)$, and (vi) $m+m'$ key-plus-id combinations $\{(K_j, ID_{CH_j}) \mid 1 \le j \le m+m'\}$.

The value of $m+m'$ is chosen according to memory availability of the smart card. For example, we can store $m+m' = 200$ encrypted keys along with the identifies of the cluster heads in the memory of a smart card. We have only a small number of cluster heads to be deployed for a large-scale wireless sensor network along with a large number of regular sensor nodes. Thus, if each cluster contains 220 sensor nodes, then for a hierarchical sensor network containing 22,000 regular sensor nodes we only require 100 cluster nodes in the network. Thus, it is a practical assumption to store the computed $m=100$ encrypted keys for initial deployment of cluster heads and $m' = 100$ encrypted keys for dynamic cluster heads addition so that $m+m' = 200$ encrypted keys can be stored into the memory of the smart card.

Further, note that the $m+m'$ encrypted keys stored into the memory of the smart card of a user U_i are different from those for another user U_j, because these keys are encrypted using the master keys of cluster heads along with the different identifiers of users, the identifiers of cluster heads and the secret information X_s.

This registration phase is summarized in Fig. 2.

3.3.4. Login phase

If the user U_i wants to access real-time data from the WSN, the user U_i needs to perform the following steps:

- *Step* 1: U_i inserts his/her smart card into the card reader of a specific terminal and provides his/her password PW_i.
- *Step* 2: The smart card then computes the masked password of the user U_i as $RPW'_i = h(y\|PW_i)$. Using the computed masked password, the smart card further computes $x' = h(RPW'_i\|X_A)$ and $r'_i = h(y\|x')$, and then verifies whether $r'_i = r_i$. If this verification does not hold, U_i has entered his/her password incorrectly and the scheme terminates. Otherwise, performs the following steps.
- *Step* 3: Using the system's current timestamp T_1, the smart card computes $N_i = h(x'\|T_1)$.
- *Step* 4: The user U_i selects a cluster head, say CH_j from which the real-time data can be accessed inside WSN. Corresponding to CH_j, the smart card selects the encrypted master key of CH_j, K_j from its memory and computes a ciphertext message $E_{K_j}(ID_i\|ID_{CH_j}\|N_i\|e_i\|T_1)$. Finally, the user sends the message $\langle ID_i\|ID_{CH_j}\| \ E_{K_j}(ID_i\|ID_{CH_j}\|N_i\|e_i\|T_1)\rangle$ to the *BS*, via a public channel.

The login phase is summarized in Fig. 3.

3.3.5. Authentication phase

After receiving the login request message $\langle ID_i\|ID_{CH_j} \|E_{K_j}(ID_i\|ID_{CH_j}\|N_i \ \|e_i\|T_1)\rangle$ from the user U_i, the *BS* performs the following steps in order to authenticate the user U_i:

- *Step* 1: The *BS* computes a key K using the stored master key MK_{CH_j} of the cluster head CH_j as $K = E_{MK_{CH_j}}(ID_i\|ID_{CH_j}\|X_s)$. Using this computed key K, the *BS* decrypts $E_{K_j}(ID_i\|ID_{CH_j} \ \|N_i\|e_i\|T_1)$ and thus, $D_K[E_{K_j}(ID_i\|ID_{CH_j}\|N_i\|e_i\|T_1)] = (ID_i\|ID_{CH_j}\|N_i\|e_i\|T_1)$.
- *Step* 2: The *BS* checks if retrieved ID_i is equal to received ID_i and also if retrieved ID_{CH_j} is equal to received ID_{CH_j}. If these hold, the *BS* further checks if $|T_1 - T_1^*| < \triangle T_1$, where T_1^* is the current system timestamp of the *BS* and $\triangle T_1$ is the expected time interval for the transmission delay. Now, if it holds, the *BS* further computes $X = h(ID_i\|X_s)$, $Y = e_i \oplus X$, and $Z = h(Y\|T_1)$. If $Z = N_i$, then the *BS* accepts U_i's login request and U_i is

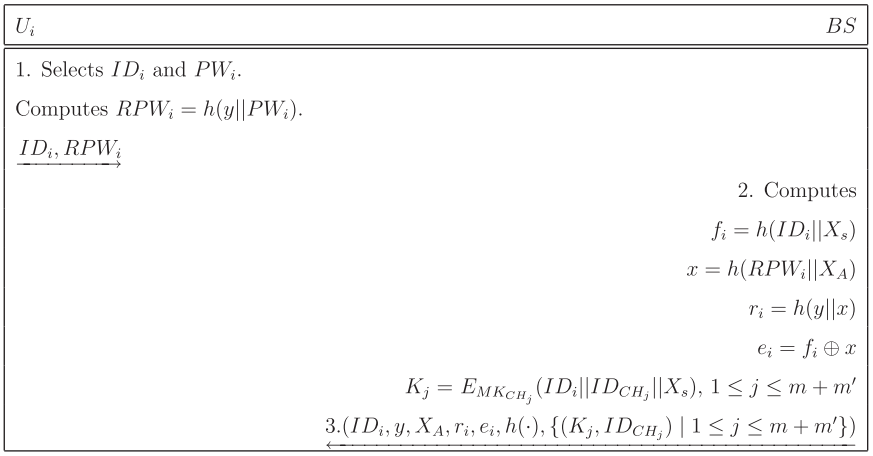

Fig. 2. Registration phase of our proposed scheme.

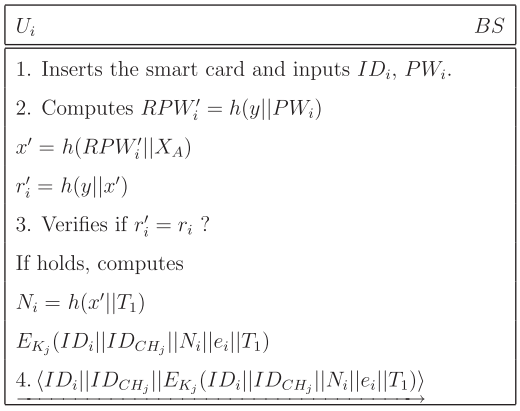

Fig. 3. Login phase of our proposed scheme.

considered as a valid user by the *BS*. Otherwise, the scheme terminates.

- *Step* 3: Using the current system timestamp T_2, the *BS* computes $u = h(Y\|T_2)$ and produces a ciphertext message encrypted using the master key MK_{CH_j} of the cluster head CH_j as $E_{MK_{CH_j}}(ID_i\|ID_{CH_j}\|u\|T_1\|T_2\|X\|e_i)$. The *BS* sends the message $\langle ID_i\|ID_{CH_j} \|E_{MK_{CH_j}}(ID_i \|ID_{CH_j}\|u\|T_1\|T_2\|X\|e_i)\rangle$ to the corresponding cluster head CH_j.
- *Step* 4: After receiving the message in Step 3 from the *BS*, the cluster head CH_j decrypts $E_{MK_{CH_j}}(ID_i \|ID_{CH_j}\|u\|T_1\|T_2\|X\|e_i)$ using its own master key MK_{CH_j} as $D_{MK_{CH_j}}[E_{MK_{CH_j}}(ID_i \|ID_{CH_j}\|u\|T_1\|T_2\|X\|e_i)] = (ID_i\|ID_{CH_j}\|u\|T_1\|T_2\|X\|e_i)$. CH_j then checks if retrieved ID_i is equal to received ID_i and also if retrieved ID_{CH_j} is equal to received ID_{CH_j}. If these hold, CH_j further checks if $|T_2 - T_2^*| < \triangle T_2$, where T_2^* is the current system timestamp of the CH_j and $\triangle T_2$ is the expected time interval for the transmission delay.
  If it holds good, CH_j computes $v = e_i \oplus X = h(RPW_i\|X_A)$, $w = h(v\|T_2) = h(h(RPW_i\|X_A)\|T_2)$. CH_j then checks if $w = u$. If it does not hold, the scheme terminates. Otherwise, if it holds, the user U_i is considered as a valid user and authenticated by CH_j.
  CH_j computes a secret session key SK shared with the user U_i as $SK = h(ID_i\|ID_{CH_j}\|e_i\|T_1)$. Finally, CH_j sends an acknowledgment to the user U_i via other cluster heads and the *BS* and responds to the query of the user U_i.
- *Step* 5: After receiving the acknowledgment from CH_j, the user U_i computes the same secret session key shared with CH_j using its previous system timestamp T_1, ID_i, ID_{CH_j} and e_i as $SK = h(ID_i\|ID_{CH_j}\|e_i\|T_1)$. Thus, both user U_i and cluster head CH_j will communicate securely in future using the derived secret session key SK.

This authentication phase is summarized in Fig. 4.

3.3.6. Password change phase

In this phase, any user U_i can change his/her password freely and completely locally without the help of the *BS*. This phase contains the following steps:

- *Step* 1: U_i inputs his/her smart card into the card reader of a specific terminal and provides his/her identifier ID_i, old password PW_i^{old} as well as new changed password PW_i^{new}. After that the smart card computes the masked old password of the user U_i as $RPW_i^* = h(y\|PW_i^{old})$, $M_1 = h(RPW_i^*\|X_A)$, and $M_2 = h(y\|M_1)$.
- *Step* 2: The smart card then compares the computed M_2 with the stored r_i in its memory. If they do not match, this means that the user U_i has entered his/her old password PW_i^{old} incorrectly and hence, the password change phase terminates immediately.
- *Step* 3: The smart card computes $M_3 = e_i \oplus M_1 = h(ID_i\|X_s)$, $M_4 = h(y\|PW_i^{new})$, $r_i' = h(y\|M_4)$, $M_5 = h(M_4\|X_A)$, $e_i' = M_3 \oplus M_5 = h(ID_i\|X_s) \oplus h(h(y\|PW_i^{new})\|X_A)$.
- *Step* 4: Finally, replace r_i with r_i' and e_i with e_i' into the memory of the smart card.

3.3.7. Dynamic node addition phase

In this phase, we describe the methods for adding new nodes in the existing network. If some sensor nodes or cluster heads are captured by an attacker or some nodes expire due to energy problem, we often require to add some new nodes in order to replace those nodes or some nodes can be later added in the network.

Case 1. Addition of sensor nodes
In this case, if a sensor node S_i is deployed in a cluster C_i, prior to deployment of that node, then the *BS* assigns the unique identifier ID_{S_i} to it and also randomly generated unique master key MK_{S_i}. These information are loaded in the memory of the node S_i.

Case 2. Addition of cluster heads
When a cluster head CH_j needs to be added in a cluster C_i prior to deployment of that cluster head, CH_j, the BS assigns the unique identifier ID_{CH_j} to it and also randomly generated unique master key MK_{CH_j} to it as already done in registration phase (see Section 3.3.3). These information are finally loaded in the memory of the cluster head CH_i.

After deploying sensor nodes in their corresponding cluster along with its cluster head in the deployment field, the *BS* informs the user U_i about the addition of the cluster heads. Thus, it is noted that no other information regarding cluster heads addition is required to store in the user's smart card.

4. Security analysis of the proposed scheme

In this section, for security analysis of our proposed scheme, we use the threat model described in Section 3.1. We now show that our scheme can resist against the following attacks.

4.1. Replay attack

Suppose an attacker intercepts a valid login request message $\langle ID_i\| ID_{CH_j} \|E_{K_j}(ID_i \|ID_{CH_j}\| N_i \|e_i\|T_1)\rangle$ in the login phase and tries to login to the *BS* by replaying the same. The verification of this login request will fail in our scheme by the attacker because he/she has to know N_i, e_i and T_1. However, to retrieve these from the login request message, he/she needs to know the encrypted master key $E_{MK_{CH_j}}(ID_i\|ID_{CH_j}\|X_s)$ of the cluster head CH_j. The attacker is further unable to compute X, Y, Z and u for verification of the login request message. Thus, the proposed scheme can resist replay attack.

4.2. Many logged-in users with the same login-id attack

The proposed scheme can prevent the threat of the many logged-in users with the same login-id. The systems which maintain the password table to verify user login are usually vulnerable to this kind of attack. However, our scheme requires on-card computation for login to the WSN, and once the smart

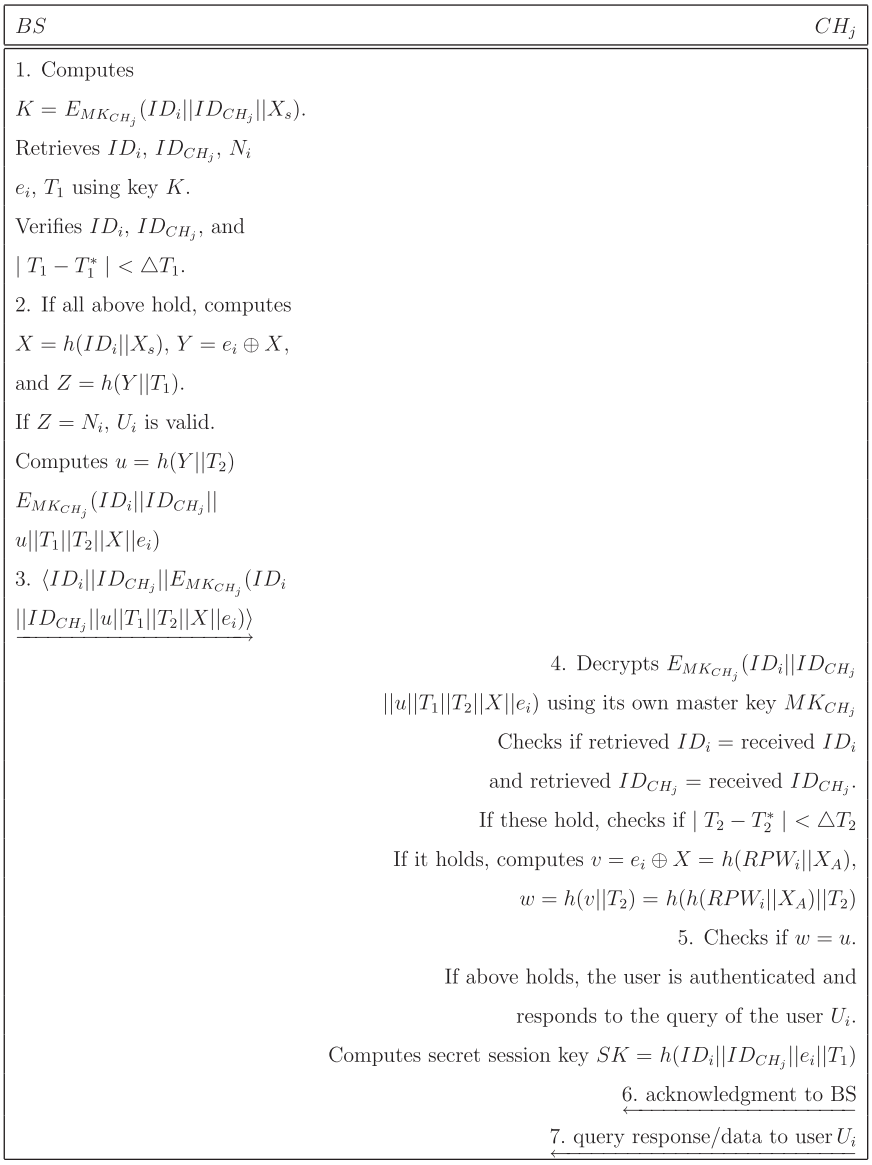

Fig. 4. Authentication phase of our proposed scheme.

card is removed from the system, the login process is aborted. Consider an example of two users U_i and U_j having the same password. The random secret number y is used in computation of their masked passwords. Hence, even if two users have same ids and passwords, they will have different the masked passwords. As a result, even if two users have same password, problem of many logged in users with same login id do not arise in our scheme and our scheme resists this kind of threat.

4.3. Stolen-verifier attack

Our scheme is resilient against stolen-verifier attack because the proposed scheme does not require to store any verifier/password table for verification. It is also noted that the *BS*, cluster heads and sensor nodes do not keep password tables. As a result, in our scheme an attacker cannot steal users' password tables.

4.4. Password guessing attack

In our protocol, the login message $\langle ID_i \| ID_{CH_j} \| E_{K_j}(ID_i \| ID_{CH_j} \| N_i \| e_i \| T_1)\rangle$ is transmitted. The attacker cannot guess the user's password PW_i from $E_{K_j}(ID_i \| ID_{CH_j} \| N_i \| e_i \| T_1)$ because the attacker has to decrypt it using the encrypted master key $K_j = E_{MK_{CH_j}}(ID_i \| ID_{CH_j} \| X_s)$ of the cluster head CH_j and then to retrieve N_i and e_i. Moreover, even if attacker knows the master key MK_{CH_j} of the cluster head CH_j after capturing that CH_j in the network, he/she is still unable to get the encrypted master key $K_j = E_{MK_{CH_j}}(ID_i \| ID_{CH_j} \| X_s)$ of the cluster head CH_j because the secret information X_s is unknown to the attacker (X_s is only known to the *BS*). Thus, computing user's password from N_i and e_i is computationally infeasible due to the one-way characteristic property of the hash function $h(\cdot)$. In this way, our scheme can resist guessing attack.

4.5. Password change attack

Suppose a legal user lost his/her smart card or his/her smart card has been stolen. Suppose the attacker can also breach the information $(ID_i, y, X_A, r_i, e_i, h(\cdot), \{(K_j, ID_{CH_j}) | 1 \le j \le m+m'\})$ which are stored in the smart card. In our scheme, for changing the password the attacker has to pass the old password PW_i^{old} verification. However, it is computationally infeasible to derive the old password PW_i^{old} from r_i and e_i due to the one-way

characteristic property of the hash function $h(\cdot)$ and the secret information X_s only maintained by the *BS*. As a result, for a success in this attack the attacker has to guess the old password before updating the new password chosen by him/her.

4.6. Resilience against node capture attack

We measure the resilience against node capture attack of a user authentication scheme in WSN by estimating the fraction of total secure communications that are compromised by a capture of c nodes *not including* the communication in which the compromised nodes are directly involved, that is, we want to find out the effect of c cluster nodes being compromised on the rest of the network. For example, for any non-compromised cluster head CH_j, we need to find out the probability that the adversary can decrypt the secure communication between CH_j and a user U_i when c cluster heads are already compromised. We denote this probability by $P_e(c)$. Now, if $P_e(c)=0$, we call such an user authentication scheme as *unconditionally secure against node capture attack*.

Due to unattended nature of WSN, nodes can be captured by an attacker. Assume that some cluster heads are captured by the attacker. If the attacker captures a cluster head, he/she knows its master key from its memory. Note that each node (including cluster head) is given prior to deployment in the target field a unique randomly generated master key. Thus, the attacker will be able to compromise the master key of that captured cluster head only. Using that compromised master key the attacker can response with false data to the legitimate user only. However, other non-compromised nodes can still communicate securely with the actual real-time data to the legitimate users. As a result, compromise of that captured cluster head does not lead to compromise any other secure communication between the user and the non-compromised nodes in the network. In this way, our scheme provides unconditional security against node capture attack.

Remark. When a cluster head (CH) is compromised, an attacker then compromises its own master key and session key shared with a user. In addition, secure communication with its neighbor sensor nodes are also compromised. However, other non-compromised sensor nodes in the cluster in which compromised CH is charge of, communicate securely with each other in that cluster because we use the unconditionally secure key management scheme (Das, 2009) for secure communication between nodes in each cluster. In Das (2009), we take 220 sensor nodes in a cluster so that any two neighbor nodes can establish a secure link for their secure communication when each sensor node and cluster head store 200 keys in their key rings. Thus, if we have $m=100$ clusters in the sensor network which are initially deployed, then total sensor nodes can be deployed is $220\times 100=22000$, which constitutes a large-scale WSN.

According to threat model (given in Section 3.1), compromised CHs and sensor nodes in each cluster can be detected and as a result, the BS knows about compromised nodes in the network. The BS then informs the user U_i about compromised CHs and thus U_i will not send any queries in order to retrieve data from the compromised CHs. Therefore, it is necessary to re-deploy new cluster heads or sensor nodes to replace the compromised CHs or sensor nodes in the existing network. We first consider that a sensor node is compromied by an attacker and it is replaced by a new sensor node. In our scheme, during dynamic sensor nodes addition phase (described in Section 3.3.7), a new sensor node is assigned its own identifier and a unique randomly generated master key which are different from the identifiers and master keys of the compromised nodes. After its deployment, it will establish with its neighbor sensor nodes as well as its cluster head (if the cluster head is its neighbor) in the cluster using (Das, 2009). Note that according to Das (2009), each secure link uses a distinct secret pairwise key.

We now consider that a cluster head in a cluster is compromised and that CH will be replaced by a new CH in that cluster. From our dynamic cluster heads addition phase (described in Section 3.3.7), we see that new cluster head is assigned its own identifier and a unique randomly generated master key before its deployment in the existing network, which are different from the identifiers and master keys of the compromised nodes. We further note that the user U_i's smart card contains additional m' key-plus-identifier combinations $\{(K_{m+j},ID_{CH_{m+j}})|1\le j\le m'\}$, and hence we can deploy additional m' cluster heads CH_{m+j} $(1\le j\le m')$ after initial deployment into the existing network. In our scheme, the new deployed cluster head needs to be one of the m' cluster heads CH_{m+j} $(1\le j\le m')$ for which the information are already stored in U_i's smart card. For example, a compromised cluster head CH_j in a cluster will be replaced by a cluster head CH_{m+j}, for some $j\in\{1,2,\ldots,m'\}$. After its deployment in that cluster, it establishes secure communication links with its neighbor sensor nodes in that cluster and also with its neighbor cluster heads in the network using (Das, 2009). As a result, there is no need to update information about addition of CH_{m+j} in the user U_i's smart card. Of course, our scheme tolerates addition of m' (for example, $m'=100$) new cluster heads in the network. In practice, we assume that not all m cluster heads, which were deployed in the initial deployment, are compromised by the attacker, and hence, it is reasonable to believe that some cluster heads could be compromised by the attacker and they will be replaced by new cluster heads whose information are already in user's smart card. Finally, the BS needs to inform U_i about addition of cluster heads in clusters so that U_i can access data from those cluster heads. Thus, our scheme is suitable when a cluster head is added into the existing network after initial deployment of nodes.

4.7. Smart card breach attack

As in Fan et al.'s scheme (Fan et al., 2010), although the smart card is assumed safe and cannot be cracked, however there is a risk of smart card crack. If an attacker/intruder attains a smart card and cracks it, he/she can obtain its stored infomation ID_i, y, X_A, $r_i=h(y\|h(RPW_i\|X_A))$, $e_i=f_i\oplus x\ =h(ID_i\|X_s)\oplus h(RPW_i\|X_A)$, $h(\cdot)$, $\{(K_j,ID_{CH_j})|1\le j\le m+m'\}$, where $K_j=E_{MK_{CH_j}}(ID_i\|ID_{CH_j}\|X_s)$. However, the attacker has no feasible way to know the user U_i's password PW_i from r_i and e_i due to one-way property of the hash function $h(\cdot)$. Since the secret information X_s is only known to the BS, there is no feasible way for the attacker to obtain the master key MK_{CH_j} of the cluster head CH_j from K_j again due to one-way property of the hash function $h(\cdot)$. The attacker needs to guess the user U_i's correct password PW_i in order to pass the password verification in the login phase. Moreover, the computation of N_i in the login phase becomes infeasible due to also one-way property of the hash function $h(\cdot)$. Hence, our scheme prevents from smart card breach attack.

4.8. Denial-of-service attack

In our scheme, the *BS* sends message with expecting acknowledgment from the cluster head requested by the user. The denial-of-service attack is not possible in our scheme because at the end of each user authentication, an acknowledgment is sent to the user U_i via the *BS* which allows the user to duly know that response coming from cluster head is authentic.

4.9. Privileged-insider attack

During the registration phase of the proposed scheme (see in Section 3.3.3), the user U_i does not send his/her password PW_i in plaintext, instead the user U_i sends the hashed password $RPW_i = h(y\|PW_i)$ to the BS. It is computationally infeasible task to retrieve PW_i from RPW_i due to one-way property of the hash function $h(\cdot)$. The system manager or a privileged-insider of the BS does not know the password PW_i of U_i, and he/she is thus unable to impersonate U_i by accessing other servers where U_i could be also a registered user if U_i uses the same password PW_i for his/her convenience of remembering long password and easy-to-use whenever required. Consequently, our scheme is secure against privileged-insider attack.

4.10. Masquerade attack

Consider that an illegal user may try to fabricate fake login request message to cheat the *BS* to convince that it is a legal login request in the login phase. Note that the user sends the message $\langle ID_i\|ID_{CH_j}\|E_{K_j}(ID_i\|ID_{CH_j}\|N_i\|e_i\|T_1)\rangle$ to the *BS*. In order to convince the *BS* that it is a legal remote login request, the illegal user has to decrypt $E_{K_j}(ID_i\|ID_{CH_j}\|N_i\|e_i\|T_1)$ using the key K_j correctly, where $K_j = E_{MK_{CH_j}}(ID_i\|ID_{CH_j}\|X_s)$. Suppose the illegal user captures the cluster head CH_j in the network and gets its master key MK_{CH_j}. Since the secret information X_s is only known to the *BS*, even after knowing the master key MK_{CH_j} of the captured cluster head CH_j, the illegal user cannot still compute the key K_j using the identifiers ID_i and ID_{CH_j}. Thus, our scheme resists this attack.

5. Comparison with related schemes

In this section, we compare the performances of our proposed scheme with the existing related password-based schemes: Watro et al.'s scheme Watro et al. (2004), Wong et al.'s scheme Wong et al. (2006), M.L. Das's scheme Das (2009), Nyang-Lee's scheme Nyang and Lee (2009), Huang et al.'s scheme Huang et al. (2010), He et al.'s scheme He et al. (2010), Vaidya et al.'s scheme Vaidya et al. (2010), Fan et al.'s scheme Fan et al. (2010), and Chen-Shih's scheme Chen and Shih (2010).

In Table 2, we have compared the secrecy level provided by Watro et al. (2004), Wong et al. (2006), Das (2009), Nyang and Lee (2009), Huang et al. (2010), He et al. (2010), Vaidya et al. (2010), Fan et al. (2010) and Chen and Shih (2010) with that provided by our scheme. Password change feature is supported by our scheme, Huang et al. (2010); He et al. (2010); Vaidya et al. (2010), while our scheme, Huang et al. (2010); Vaidya et al. (2010) are also resilient against node capture attack. Our scheme is secure against denial-of-service attack and also provides mutual authentication between the user and the cluster head. Among all the schemes, only our scheme and (Fan et al., 2010) establish a secret session key between the user and cluster head that allows the user to collect data from a particular cluster (data from a particular region of the target field) for a given session. In addition, our scheme prevents smart card breach attack as in Fan et al.'s scheme (Fan et al., 2010) and privileged-insider attack. Considering other existing approaches, our scheme is better in terms of security. Besides these security aspects, our scheme supports addition of new cluster heads and sensor nodes easily in the existing network in order to replace the compromised cluster heads and sensor nodes. The proposed scheme does not require to update extra information about new nodes addition in the user's smart card.

In Table 3, we have compared computational costs in different phases like registration, login and authentication phases of Watro et al. (2004), Wong et al. (2006), Das (2009), Nyang and Lee (2009), Huang et al. (2010), He et al. (2010), Vaidya et al. (2010), Fan et al. (2010), Chen and Shih (2010) with our scheme. We have ignored XOR operation in comparison, because XOR operation is negligible. Since the cluster head CH_j and *BS* are more resource-rich as compared to usual sensor nodes, our scheme makes advantages of using the computational power of *BS* and cluster heads to provide secure log-in to user. This makes our scheme more efficient in terms of resource usage as compared to other related schemes.

In Table 4, we have compared the communication costs among our scheme and the related other schemes. The communication cost is considered in terms of number of message exchanges for a successful user authentication. From this table, it is noted that a successful user authentication process in our scheme requires four message exchanges, while Watro et al.'s scheme, Wong et al.'s scheme, M. L. Das's scheme, Nyang-Lee's scheme, Huang et al.'s scheme, He et al.'s scheme, Vaidya et al.'s scheme, Fan et al.'s scheme and Chen-Shih's scheme require two, four, three, three, three, three, five, five and four message exchanges, respectively. Although Watro et al.'s scheme requires minimum number of message exchanges among all the schemes, their scheme is computational expensive due to involvement of public-key computations such as modular exponentiation operations in resource-constrained WSN. In our scheme, the message transmission between the *BS* and the cluster head is done effectively because the message transmission is done using a very few number of hops due to involvement of cluster heads in the communication path. In all other schemes, the message transmission between the *BS* (GW-node) and sensor node often requires multi-hop communication path and as a result our scheme is significantly efficient in term of communication cost as compared to the other schemes.

Finally, we compare the sensor node's energy cost between the proposed scheme and the other schemes in Table 5. The sensor node's energy cost is due to both computational and communication costs involved in the schemes. For Watro et al.'s scheme,

Table 2
Performance comparison between the proposed scheme and other related schemes.

	Watro et al. (2004)	Wong et al. (2006)	Das (2009)	Nyang and Lee (2009)	Huang et al. (2010)	He et al. (2010)	Vaidya et al. (2010)	Fan et al. (2010)	Chen and Shih (2010)	Ours
I1	No	No	No	No	Yes	Yes	Yes	No	No	Yes
I2	Yes	No	No	No	No	No	Yes	Yes	Yes	Yes
I3	No	No	No	No	No	No	Yes	Yes	No	Yes
I4	No	No	No	No	Yes	No	Yes	No	No	Yes
I5	No	No	No	No	No	No	No	Yes	No	Yes
I6	No	No	No	No	No	No	No	No	No	Yes

I1: whether supports change password or not; I2: whether supports mutual authentication or not; I3: whether resists denial-of-service attack or not; I4: whether resilient against node capture attack or not; I5: whether establish secret session key between user and sensor node/cluster head or not; and I6: whether supports dynamic node addition or not.

Table 3
Comparison of computational costs in different phases between the proposed scheme and the other schemes.

Phase	User or Node	Watro et al. (2004)	Wong et al. (2006)	Das (2009)	Nyang and Lee (2009)	Huang et al. (2010)	He et al. (2010)	Vaidya et al. (2010)	Fan et al. (2010)	Chen and Shih (2010)	Ours
	User (U_i)	$t_{pu}+t_{pr}$	–	–	–	–	t_h	t_h	–	–	t_h
Registration	BS	t_{pr}	$3t_h$	$3t_h$	$3t_h$	$4t_h$	$5t_h$	$4t_h$	$6t_h$	$3t_h$	$(m+m')t_{enc}+3t_h$
	Sensor	–	–	–	–	–	–	–	–	–	–
	Cluster head	–	–	–	–	–	–	–	–	–	–
Login	User (U_i)	$2t_{pr}+t_h$	–	$4t_h$	$4t_h+2t_{kdf}+t_{mac}+t_{dec}$	$4t_h$	$5t_h$	$6t_h$	$7t_h$	$4t_h$	$5t_h+t_{enc}$
+	BS	–	t_h	$4t_h$	$3t_h+2t_{kdf}+t_{enc}+t_{mac}$	$6t_h$	$5t_h$	$5t_h$	$2t_h$	$5t_h$	$3t_h+t_{dec}+2t_{enc}$
Authentication	Sensor	$2t_{pu}+t_h$	$3t_h$	t_h	$2t_{kdf}+2t_{mac}+t_{dec}+t_{enc}$	t_h	t_h	$2t_h$	$2t_h$	t_h	–
	Cluster head	–	–	–	–	–	–	–	$8t_h$	–	$2t_h+t_{dec}$

t_{pu}: public-key computation; t_{pr}: private-key computation; t_h: hash computation; t_{enc}: symmetric-key encryption; t_{dec}: symmetric-key decryption; t_{kdf}: key derivation function computation; and t_{mac}: message authentication code (MAC) function computation.

Table 4
Comparison of communication costs between the proposed scheme and the other schemes.

Scheme	Communication cost
Watro et al. (2004)	2 messages
Wong et al. (2006)	4 messages
Das (2009)	3 messages
Nyang and Lee (2009)	3 messages
Huang et al. (2010)	3 messages
He et al. (2010)	3 messages
Vaidya et al. (2010)	5 messages
Fan et al. (2010)	5 messages
Chen and Shih (2010)	4 messages
Ours	4 messages

a sensor node consumes battery due to nonce validation, checksum generation and verification, two public-key operations and then response to the user's query. In Wong et al.'s scheme, a sensor node consumes battery for a lookup table query, three hash operations for parameters generation and then waiting for the GW-node's response before responding to the user's query. In M. L. Das's scheme, a sensor node consumes battery due to timestamp validation and one hash operation for parameter generation and for responding to the user's query. Battery power consumption for a sensor node in Nyang-Lee's scheme is due to timestamp validation, one key derivation function (kdf) operation for computing encryption key, one kdf operation for computing MAC key, two MAC operations for parameter generation, one encryption for encrypting data and then another decryption for retrieving MAC and encryption keys shared between the use and the sensor node. In case of Huang et al.'s scheme, a sensor node consumes battery due to timestamp validation and one hash operation for parameter verification and finally for responding to the user's query. In He et al.'s scheme, a sensor node consumes battery due to timestamp validation, one hash function for parameter generation and response to the user's query. In case of Vaidya et al.'s scheme, battery consumption for a sensor node comes due to timestamp validation, one hash function for parameter generation, another hash function for parameter verification, and then response to the user's query and waiting for the GW-node's response. Fan et al's scheme requires battery consumption for a sensor node due to one hash function for random-nonce validation, another hash function for session key generation and then response to the user's query. Chen-Shih's scheme needs battery consumption for a sensor node due to time-stamp validation, one hash function for parameter generation and response to the user's query. Finally in our scheme, a cluster head consumes battery due to timestamp validation, two hash operations for parameter generation and session key generation, one symmetric key decryption and response to the user's query and sending an acknowledgment to the user via the BS for a successful user authentication and session key establishment. Note that in our scheme, a sensor node does not consume battery for user authentication process. Due to efficient hash and symmetric key operations, sensor node's energy cost in our scheme is comparable with that for other schemes.

6. Conclusion

We have proposed a new password-based user authentication scheme for large-scale hierarchical wireless sensor networks. The proposed scheme allows the user to authenticate at both the base station and the cluster heads inside WSN. After successful authentication, both the user and the cluster head from which user wants to access real-time data in the target field, will be able to establish a secret session key between them. Later using this session key, the user can contact the cluster head for real-time data inside WSN. The proposed scheme supports dynamic node addition phase and in that case, there is no need to update stored information in the user's smart card for accessing real-time data from the added/replaced cluster heads in the network. Further, our scheme provides better security compared with other related schemes and supports changing of user's password locally without contacting the base station. In addition, our scheme is also efficient in terms of communication and computation as compared with other related schemes.

Table 5
Comparison of energy cost of sensor node/cluster head between the proposed scheme and the other schemes.

Scheme	Sensor node/cluster head's energy cost
Watro et al. (2004)	nonce validation+checksum generation and verification + two public-key operations +response to the user's query
Wong et al. (2006)	lookup table query+three hash operations for parameters generation +waiting for the GW-node's response+response to the user's query
Das (2009)	timestamp validation+one hash operation for parameter generation +response to the user's query
Nyang and Lee (2009)	timestamp validation+one key derivation function operation for computing encryption key+one key derivation function operation for computing MAC key+two MAC operations for parameters generation+one encryption for encrypting data+one decryption for retrieving MAC and encryption keys shared between user and sensor node
Huang et al. (2010)	timestamp validation+one hash operation for parameter verification +response to the user's query
He et al. (2010)	timestamp validation+one hash operation for parameter generation +response to the user's query
Vaidya et al. (2010)	timestamp validation+one hash operation for parameter verification +one hash operation for parameter generation+response to the user's query +waiting for the GW-node's response
Fan et al. (2010)	one hash operation for random nonce validation+one hash operation for session key generation +response to the user's query
Chen and Shih (2010)	timestamp validation+one hash operation for parameter generation +response to the user's query
Ours	timestamp validation+two hash operations for parameter generation and symmetric session key generation+one symmetric-key decryption +response to the user's query+acknowledgment to the BS

Acknowledgments

The authors would like to acknowledge the many helpful suggestions of the anonymous reviewers, the Editor and the Editor-in-Chief of this Journal, which have improved significantly the content and the presentation of this paper.

References

Advanced Encryption Standard (AES). FIPS PUB 197, National Institute of Standards and Technology (NIST), U.S. Department of Commerce; November 2001. 〈http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf〉.

Akyildiz IF, Su W, Sankarasubramaniam Y, Cayirci E. Wireless sensor networks: a survey. Computer Networks 2002;38(4):393–422.

Chan H, Perrig A, Song D. Random key predistribution schemes for sensor networks. In: IEEE symposium on security and privacy, Berkeley, California; 2003. p. 197–213.

Chen T-H, Shih W-K. A robust mutual authentication protocol for wireless sensor networks. ETRI Journal 2010;32(5):704–12.

Cheng Y, Agrawal DP. An improved key distribution mechanism for large-scale hierarchical wireless sensor networks. Ad Hoc Networks 2007;5(1):35–48.

Crossbow Technology Inc. 2011. Wireless sensor networks, 〈http://www.xbow.com〉 Accessed on September.

Daemen J, Rijmen V. The design of rijndael, AES—the advanced encryption standard. Springer-Verlag; 2002 pp. 31–79.

Das AK. An unconditionally secure key management scheme for large-scale heterogeneous wireless sensor networks. In: First IEEE international conference on communication systems and networks (COMSNETS 2009); 2009. p. 1–10.

Das AK, Sengupta I. An effective group-based key establishment scheme for large-scale wireless sensor networks using bivariate polynomials. In: 3rd IEEE international conference on communication systems software and middleware (COMSWARE 2008); 2008. p. 9–16.

Das ML. Two-factor user authentication in wireless sensor networks. IEEE Transactions on Wireless Communications 2009;8(3):1086–90.

Diffie W, Hellman ME. New directions in cryptography. IEEE Transactions on Information Theory 1976;22:644–54.

Dolev D, Yao A. On the security of public key protocols. IEEE Transactions on Information Theory 1983;29(2):198–208.

Dong Q, Liu D. Using auxiliary sensors for pairwise key establishment in WSN. In: Proceedings of IFIP international conferences on networking (Networking 2007), Lecture notes in computer science (LNCS), vol. 4479; 2007. p. 251–62.

Eschenauer L, Gligor VD. A key management scheme for distributed sensor networks. In: 9th ACM conference on computer and communication security; November 2002. p. 41–47.

Fan R, Ping L-D, Fu J-Q, Pan X-Z. A secure and efficient user authentication protocol for two-tieres wireless sensor networks. In: Second Pacific-Asia conference on circuits, communications and system (PACCS 2010); 2010. p. 425–8.

He D, Gao Y, Chan S, Chen C, Bu J. An enhanced two-factor user authentication scheme in wireless sensor networks. Ad Hoc & Sensor Wireless Networks 2010;10(4).

Huang H-F, Chang Y-F, Liu C-H. Enhancement of two-factor user authentication in wireless sensor networks. In: Sixth international conference on intelligent information hiding and multimedia signal processing; 2010. p. 27–30.

Khan MK, Alghathbar K. Cryptanalysis and security improvements of two-factor user authentication in wireless sensor networks. Sensors 2010;10:2450–9.

Newsome J, Shi E, Song D, Perrig A. The Sybil attack in sensor networks: analysis and defenses. In: Proceedings of third IEEE international conference on information processing in sensor networks (IPSN 2004); 26–27 April 2004. p. 259–68.

Nyang DH, Lee M-K. Improvement of Das's two-factor authentication protocol in wireless sensor networks. In: Cryptology ePrint Archive. Report 2009/631; 2009.

Parno B, Perrig A, Gligor V. Distributed detection of node replication attacks in sensor networks. In: IEEE symposium on security and privacy; 8–11 May 2005. p. 49–63.

Rivest RL, Shamir A, Adleman LM. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM 1978;21:120–6.

Secure hash standard. FIPS PUB 180-1, National Institute of Standards and Technology (NIST), U.S. Department of Commerce; April 1995.

Sun SPOT. 2011. Wireless sensor networks and beyond. 〈http://www.sunspotworld.com〉 Accessed on September.

Stallings W. Cryptography and network security: principles and practices. 3rd ed. Pearson Education; 2004 p. 328–45.

Vaidya B, Makrakis D, Mouftah HT. Improved two-factor user authentication in wireless sensor networks. In: Second international workshop on network assurance and security services in ubiquitous environments; 2010. p. 600–6.

Watro R, Kong D, Cuti S, Gardiner C, Lynn C, Kruus P. TinyPK: securing sensor networks with public key technology. In: Proceedings of the 2nd ACM workshop on security of ad hoc and sensor networks, SASN 2004, Washington, DC, USA; October 2004. p. 59–64.

Wong K, Zheng Y, Cao J, Wang S. A dynamic user authentication scheme for wireless sensor networks. In: Proceedings of IEEE international conference on sensor networks, ubiquitous, and trustworthy computing, IEEE Computer Society; 2006. p. 244–51.

Yuan J, Jiang C, Jiang Z. A biometric-based user authentication for wireless sensor networks. Wuhan University Journal of Natural Sciences 2010;15(3):272–6.

Zhu B, Setia S, Jajodia S, Roy S, Wang L. Localized multicast: efficient and distributed replica detection in large-scale sensor networks. IEEE Transactions on Mobile Computing 2010;9(7):913–26.

ZigBee. 2011. Wireless technology for low-power sensor networks. 〈http://www.zigbee.org〉 Accessed on September.