IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 21, 2017, accepted October 13, 2017, date of publication October 25, 2017, date of current version November 28, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2766090

Seamless Key Agreement Framework for Mobile-Sink in IoT Based Cloud-Centric Secured Public Safety Sensor Networks

FADI AL-TURJMAN[1], (Member, IEEE), YONEY KIRSAL EVER[2], (Member, IEEE), ENVER EVER[1], (Member, IEEE), HUAN X. NGUYEN[3], (Senior Member, IEEE), AND DEEBAK BAKKIAM DAVID[1]

[1]Computer Engineering Program, Middle East Technical University Northern Cyprus Campus, Mersin 10, Turkey
[2]Software Engineering Department, Near East University, 99138 Nicosia, Cyprus
[3]Design Engineering and Maths Department, Middlesex University, London NW4 4BT, U.K.

Corresponding author: Enver Ever (eever@metu.edu.tr)

This work was supported by the Newton Fund Institutional Links Grant under Grant 216429427 through the U.K. Department of Business, Energy and Industrial Strategy and managed by the British Council.

ABSTRACT Recently, the Internet of Things (IoT) has emerged as a significant advancement for Internet and mobile networks with various public safety network applications. An important use of IoT-based solutions is its application in post-disaster management, where the traditional telecommunication systems may be either completely or partially damaged. Since enabling technologies have restricted authentication privileges for mobile users, in this paper, a strategy of mobile-sink is introduced for the extension of user authentication over cloud-based environments. A seamless secure authentication and key agreement (S-SAKA) approach using bilinear pairing and elliptic-curve cryptosystems is presented. It is shown that the proposed S-SAKA approach satisfies the security properties, and as well as being resilient to node-capture attacks, it also resists significant numbers of other well-known potential attacks related with data confidentiality, mutual authentication, session-key agreement, user anonymity, password guessing, and key impersonation. Moreover, the proposed approach can provide a seamless connectivity through authentication over wireless sensor networks to alleviate the computation and communication cost constraints in the system. In addition, using Burrows–Abadi–Needham logic, it is demonstrated that the proposed S-SAKA framework offers proper mutual authentication and session key agreement between the mobile-sink and the base station.

INDEX TERMS Secure public safety networks, Internet of Things, cloud systems, session-key agreement, bilinear pairing.

I. INTRODUCTION

The Internet of Things (IoT) is a novel paradigm where objects become part of the Internet. It has converged technologies in terms of sensing, computing, information processing, networking and controlling intelligent technologies [1], [2]. Among the technologies converged we can count wireless sensor networks (WSNs), intelligent sensing, remote sensing, radio frequency identification (RFID), near field communications (NFC), low-energy wireless communications, and cloud computing. The technologies involved have particular applications in public safety as well as other domains such as health monitoring, smart homes and environments, smart cities, smart grid, and various types of pervasive systems [3].

WSNs are composed of base-stations and numerous low cost mobility nodes which have restricted resources, such as communication, storage and computation cost. Each mobility node has its own sensing-unit, data-processing unit, module for short-range communication and power-supply unit [9]. Recently, WSNs have had its own prominence in various application fields, namely military (missile target tracking / detection system), environment (hazardous detection), biomedical (health monitoring and patient tracking) and building (smart-homing and threat detection). Since WSNs have limited power-supply unit for the mobility nodes, some researchers [7], [8] have introduced the technique of mobile-sink in the WSNs for the extension of network lifetime. Since the mobility nodes transmit the confidential data via wireless channels, any user may act as an adversary to overhear / tamper the confidential data being transmitted on the WSNs. Lately, Cloud Computing (CC) techniques have been further emerged as well with the WSNs' for the purposes of storage and data access at any time over the Internet [3], [4]. In the cloud, the user can find the set of hardware devices, network connections, storage spaces, data services and application interfaces that are easily accessed over the Internet.

IoT architecture can be implemented as either Internet centric or object centric. The former aims at provisioning services within the Internet, where data are contributed by objects and vendors who deterministically deploy these objects, whereas the latter aims at provisioning services via network of smart objects. Scalability and cost efficiency of IoT services can be achieved by the integration of cloud-computing into the IoT architecture, i.e., cloud-centric IoT [50]. In a cloud-centric IoT framework, sensors provide their sensed data to a storage cloud as a service, which then undergoes data analytics and data mining tools for information retrieval and knowledge discovery [50].

With the evolution of IoT, the global data networks are interconnected and accessed over CC networking systems [4], [5]. As the sensing data are transmitted over public networks, the adversary can easily intercept the exchange of data between the users and the remote servers. This would cause various possible attacks, such as replay, key impersonation, stolen verifier, etc. [14], [15]. As the CC has become a prominent domain for secure authentication, WSNs are in high demands of security schemes for the purpose of user authentication, authorization and accounting while the cloud services are being accessed by the legitimate users.

In literature, the gateway/base-station based authentication schemes have provided the lightweight authentication for the enrichment of security properties [10]–[22]. As a result, WSNs presume the gateway/base-station as imperative part to sense the real-time data over insecure networks. In this paper, users are referred as mobile-sink. In a state of proper access towards the sensor-node, the mobile-sink should complete the proper registration to establish an authorized session between the sensor-node and the base-station. In addition, the successful establishment of this communication can only be achieved through the satisfaction of mutual authentication and session-key agreement. Generally, the authentication schemes try to satisfy all the security properties of authentication and key protocol (AKA), such as mutual authentication, session key agreement, user anonymity, etc [14]. The analyses performed on the existing studies show that several authentication schemes are still susceptible for various potential attacks, such as privileged-insider, key impersonation, stolen smart-card etc. As a result, this paper presents a seamless secure authentication and key agreement (S-SAKA) using bilinear pairing and elliptic-curve cryptosystems. The objective of this framework is to provide mutual authentication, session-key agreement, data confidentiality, user anonymity, intractability and resilient to node-capture attack, key impersonation, replay, stolen smart card, and privileged-insider. In order to ease the computation and communication overhead, the authentication phase of S-SAKA does not invoke the base-station to authenticate the mobile-sink and sensor-node; and thus the S-SAKA framework is more flexible than the three-party authentication scheme when mobile-sink is employed in the Cloud WSNs.

Since the pairwise keys are randomly distributed, the adversary may have a chance to obtain a common session-key to compromise the nodes. Das *et al.* [23] and He [24] presented a dynamic-identity based authentication scheme to resist the attacks, like privileged-insider and key-compromise. However, Das *et al.* and He's approaches are still vulnerable to the potential attack of node-capture. For the enhancement of security efficiency, Deebak [17], Turkanović *et al.* [18], Farash *et al.* [19], Das *et al.* [20], Amin and Biswas [21] and Srinivas *et al.* [22] have proposed lightweight user authentication schemes. However, their authentication schemes fail to mitigate the computation and communication efficiency of the network systems as they invoke the base-station authentication. This paper proposes the S-SAKA framework, which does not only to improve data security while using the mobile-sink in the WSNs, but also provides seamless connectivity over WSNs to reduce the computation and communication overhead.

The major contributions of S-SAKA framework are as follows:

1. The existing authentication schemes [17]–[22] are thoroughly analyzed to show various susceptibilities, such as privileged-insider, key impersonation, denial of service and password guessing.
2. To address the security weaknesses of existing schemes [17]–[24], a lightweight S-SAKA framework is proposed that holds all the original merits of the existing schemes [17]–[22] to resist the potential attacks.
3. To strengthen the proposed S-SAKA framework, a formal security analysis is performed using Burrows–Abadi–Needham (BAN) logic [25]. Besides, the informal analysis is presented to claim that the proposed S-SAKA framework can be resilient to the attacks, which has not been analyzed in the literature to date.
4. Lastly, an experimental analysis is performed using MIRACLE C/C++ library to examine the computation and communication overhead of existing and proposed authentication frameworks. The evaluation result proves that the proposed S-SAKA framework provides less overhead as compared to existing authentication schemes.

When detailed analysis is carried out with formal and informal verifications, it is observed that the proposed S-SAKA scheme provides less communication overhead in comparison with other existing authentication schemes in the literature [17]–[22].

The rest of the paper is organized as follows. Section II discusses the existing secure authentication schemes. Section III illustrates an architecture of hierarchical WSNs and discourses the mathematical assumption model using bilinear pairing. Section IV presents seamless secure authentication and key agreement (S-SAKA) framework along with the security analysis. Section V shows the verification proof. Section VI compares the performance efficiency of proposed and existing authentication schemes. Finally, Section VII concludes this study

II. RELATED WORKS

Various natural or man-made disasters such as earthquakes, floods, tsunamis, nuclear power plant explosions cost significantly in terms of assets/infrastructure damage and more importantly human lives. The WSN based systems such as IoT solutions can help us to save lives since healthy communication and accurate information can make a real difference between life and death for those who are in the areas affected by the disasters. The exposure of sensitive information or similar attacks on confidentiality/integrity of information, and/or availability of resources can become an additional disaster in case proper countermeasures are not planned carefully.

With the modernization of the public safety communications, and the changes in application areas as well as new technologies introduced such as wearables, wireless body area networks, and variety of tracking devices that can be carried by responders such as rescue teams, fire fighters, and police, the IoT is expected to form a solid infrastructure for public safety applications [4]. Furthermore, although the enhancements especially in performance improvements of 3GPP LTE-A look very promising, during disaster situations these infrastructures can also be damaged or out of service [5]. There are some studies focussing on secure wireless powered device-to-device (D2D) communication in case the infrastructure is not available or partially functional [6]. However IoT based public safety networks (PSNs) are expected to have better availability in disaster scenarios since the computation is known to be more towards the distributed fashion.

Nowadays, sensor nodes are mostly used to sense the continuous data, event detection in real time environment and actuators control. These features are particularly useful for public safety applications. Specifically, micro sensing and seamless wireless connectivity became the promising technologies for various information and communication domains. These technologies are further extending for the classical categories, such as bio chemical processing, space exploration and disaster environment [9]. In order to offer better services to the users in WSNs, security is an important concern as the data transmission is performed over public networks [10]–[13] with the restrictions as follows:

1. Sensors are easily render to failure
2. Topologies of sensor networks change often
3. Sensor networks always prefer broadcast paradigms, but most of the Ad-hoc networks are point-to-point communication
4. Sensors have limited power, computation and storage

WSNs are one of the essential components of the infrastructures employed for establishment of IoT based public safety applications. Recently, security issues in WSNs have gained much attention of the researchers not only to satisfy the security properties of authentication and key agreement (AKA) protocol but also to mitigate the computation and communication cost of the system. For the achievement of minimum overhead, several lightweight authentication schemes have been proposed [26]–[30]. Watro *et al.* [35] proposed the lightweight two-factor user authentication based on RSA cryptosystem for WSNs. However, the Watro *et al.* scheme [35] is vulnerable to replay, denial of service and key impersonation attacks [27]–[31]. Wong *et al.* [16] presented a lightweight user authentication scheme for WSNs, which only demands the computation of a hashing function. Later on, Srinivas *et al.* [22] show that the Wong *et al.* scheme [35] is vulnerable to stolen verifier and many logged-in users with the same login identity attack [27]–[31]. Tseng *et al.* [29] improved the version of Wong *et al.*, which does not offer mutual authentication between the base-station and sensor-node [28]. To overcome the security weakness of mutual authentication, Lee [30] presented a novel password based dynamic user authentication scheme, which also fails to satisfy mutual authentication between the base-station and the sensor-node [28].

Eschenauer and Gligor [32] presented a random based predistribution key mechanism to provide an initial trust between the sensor nodes. In random based key pre-distribution scheme, a key is randomly chosen from a key-pool and stored in the sensor node before it is deployed in the field. As a consequence, there are some certainties to have one common key for more than one sensor node. Chan *et al.* [33] improved this authentication scheme as two-key pre-distribution that has random pairwise-key and q-composite based key predistribution. Rasheed and Mahapatra [34] proposed a threetier authentication scheme to provide a pairwise key establishment between the mobile-sink and the sensornode. Nonetheless, the schemes, such as Chan *et al.* and Rasheed *et al.* have some serious security issues, namely user anonymity, intractability, privileged-insider and impersonation attack. Watro *et al.* [35] proposed an authentication scheme using Diffie-Hellman and RSA protocol as TinyPK scheme. But then, the TinyPK scheme is still susceptible to the masquerade attack [20]. To address this issue, Das *et al.* [20] introduced a two-factor user authentication scheme.

Chen *et al.* [37] shown that the Das *et al.* scheme is unsuccessful to provide the mutual authentication between the mobile-sinks and the sensor-nodes. To overcome the security weakness of Das *et al.* [23], Chen and Shih [36] proposed a novel authentication protocol for WSNs. He [24] extended the authentication scheme of Das *et al.* [23] to resist attacks such as privileged-insider and key impersonation. Yuan *et al.* [28] presented a biometric based user authentication scheme, which has a similar architecture of Das *et al.* scheme [23] to satisfy the security properties of AKA protocol. However, the Yuan *et al.* scheme is susceptible to denial of service (DoS) and node compromise attack.

Very few studies have focused on security issues for mobile-sink in WSNs [34], [38], [46], [47]. Let us assume that the adversary wants to impersonate as a legal mobile-sink to sense the most sensitive information from sensor-node or pretend as a legitimate sensor to upload incorrect or pseudo-kind of messages to the mobile-sink. Owing to mobility in the wireless environment, most of the existing authentication schemes [20], [37], [39]–[41] are not well suited to authenticate the sensor-node and mobile-sink. As secure authentication scheme is believed to be essential to the mobile-sink in WSNs, this paper presents a novel seamless secure authentication scheme to improve the security efficiencies of the communication systems in terms of mutual authentication, session key agreement, user anonymity and intractability. The objectives of interaction and cooperation between the objects and the things are to send the data over wireless networks to signify the purpose of rapid development in the emerging technologies of IoT and cloud computing. In order to examine its common features and related discoveries, Stergiou *et al.* [48] presented a comparative study work, which focuses on the security issues of both the technologies. To provide the promising features, such as seamless interaction and interoperation, these technologies offer a smart home concept to associate the embedded computing technologies and network coverage. To solve the security and privacy preservation issues in the associated technologies, Tao *et al.* [49] presented a model of multilayer cloud. However, their architectural model fails to examine the mutual authentication and session key agreement between the communication entities.

Unlike the previous studies, S-SAKA framework tackles security issues, like data confidentiality, mutual authentication, session-key agreement, user anonymity, intractability and resilient to node-capture, key impersonation, password guessing and stolen smart-card attack for WSN configurations using the mobile-sink while providing seamless connectivity over WSNs to reduce the computation and communication overhead.

III. NETWORK MODEL AND ASSUMPTIONS

In this section, an architecture of a hierarchical WSN and the mathematical assumptions are discussed to signify the importance of the communication overhead and system security. The former is considered to mitigate the communication cost between the cluster head and the base station, whereby the network lifetime can be extended. The latter is derived to provide a better security mechanism to protect the system under various potential attacks, such as privileged-insider, replay, stolen smart-card and node capture. The important notation used in proposed S-SAKA is illustrated in Table 1.

A. NETWORK MODEL

The purpose of mobile-sink is to collect and upload the sensing data to the base-station. The principle use of mobile-sink is to mitigate the communication cost between cluster-head and base-station to enhance the network lifetime of the WSNs and reduce the communication overhead. The major disadvantage of the cloud-centric IoT is that it is usually based on a flat-topology structure that causes many problems such as scalability, increased traffic congestion among the nodes much closer to the sink (known as the broadcast storm), and an increase in overhead complexity. Therefore, clustering was introduced to subdivide the broadcast area into smaller cluster areas.

TABLE 1. Important notation used in proposed S-SAKA.

Notation	Description
M_S	Mobile-sink
B_{St}	Base-station
$U_{ser\,j}$	j^{th} user
PID_j	Identity of j^{th} user
SK_j	Secret key of j^{th} user
CH_i	i^{th} cluster head
CID_i	Unique identity of i^{th} cluster head
$H_1: \{0,1\}^*$	Map to point hashing functional operation
$H_2: \{0,1\}^*$	Secure collision free one way cryptography hashing function
$\hat{e}$	Mapping function $G \times G \rightarrow G_T$
x	Secret random integer controlled by B_{St}
$E_{S_k}(.)$	Symmetric key encryption function
ΔTS	Expected delay transmission time
TS_S	Timestamp
$\|$	Concatenation operator
$\oplus$	Bitwise X-OR operator
$Data_j$	sensing data collected by CH
S, r, y, z	Random integers $\in Z_q^*$
LC_{DB_S}	Legal cluster-head database
p_{pub}	Public key
S_{k1}, S_{k2}	Secure session key
q	prime order integers

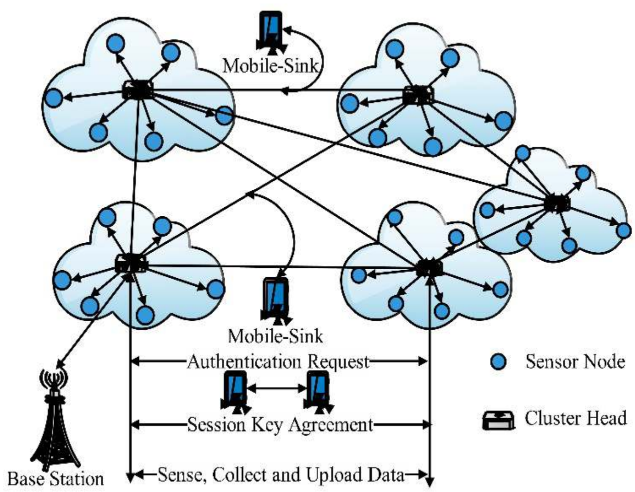

FIGURE 1. Architecture of hierarchical Cloud Based WSNs.

Many practical applications have had the model of hierarchical WSNs' for the purpose of power consumption; but then this paper is aimed to design a seamless secure authentication and key agreement (S-SAKA) under the architecture of hierarchical WSNs' to provide one-time user authentication. Fig. 1 shows the architecture of the network model considered in this study.

Fig. 1 shows that the sensor nodes are connected to cluster heads and cluster heads can communicate with each other as well as mobile-sinks. Dual way arrows on mobile sinks demonstrate the connections between sensor nodes and mobile sinks. For the cases where the communication is not within the range of the base station, the multi-hop data transmission can be employed; however, this can deteriorate the network lifetime and increase the communication overhead significantly. Hence, mobile-sinks can be employed which collects the sensed data from cluster head and upload it to the base station. The principle motivation behind the usage of mobile-sink in WSNs is to curtail the computation and communication overhead between the cluster head and the base station in order to enhance the WSNs lifetime.

B. MATHEMATICAL ASSUMPTIONS

In this subsection, the significance of Elliptic-Curve (EC) and Bilinear Pairing (BP) are introduced. In comparison with RSA, EC can provide better security level with minimum key length size [33].

Elliptic Curve: Assume that p is a prime number and f_p is a finite integer field with modulus p. Hence, an elliptic curve can be expressed as:

$$y^2 = x^3 + ax + b\,(mod\ p) \tag{1}$$

where $a,\ b\ \in\ f_p$ to satisfy the equation $4a^3 + 27b^2 \neq 0$. The scaling points $Q\,(x, y)$, which satisfies the above equation with ∞ is called as "Point at Infinity" to form an additive cyclic group as:

$$E\left[f_p\right] = \Big\{(x, y) : x, y \in f_p\ satisfy\ with\ y^2 = x^3 + ax + b\,(mod\ p) \cup \infty\}.$$

In this aspect, the scalar multiplication of $Q\,(x, y)$ on EC can be computed with the repetitive addition of n, $i.e.$, $P = P + P + P \ldots\ldots\ldots + P\,(n\ times)$. The details of the assumptions can be found in [42] and [43]. Table 1 shows the important notation used in proposed S-SAKA framework.

Bilinear Pairing: Assume that G is a cyclic (additive) group generated by a key-point P_K and G_T is a cyclic (multiplicative) group. The group parameters, such as $G\ and\ G_T$ have same prime order q. Also, assume $\hat{e} :\ G \times G \rightarrow G_T$ be a computational bilinear mapping to satisfy the properties that are as follows:

Bilinearity: Assume, $X, Y \in G$ and $p, q \in Z_q^*, \hat{e}\,(pX, qY) = \hat{e}\,(X, Y)^{pq}$; also $Z_q^* = \{k|1 \leq k \leq q-1\}$.

Non-degenerate: Assume, $X \in G, \hat{e}(X, X) \neq e$, where e is the identity of the group element G_T.

Computability: Assume, $\hat{e}(X, Y)$ be an existing algorithm to compute the key-secrecy, for any $X, Y \in G$.

Mathematical Assumptions: To prove the importance of S-SAKA mechanism, some significant mathematical problems are derived from [44] and [45] that are as follows:

- Discrete Logarithm (DL) Problem: Assume $(P, Q) \in G$ to find an integer $n \in Z_q^*$ such that $Q = nP$.
- Computational Diffie-Hellman (CDH) problem: Assume P, Px, Py, Pz for any random integer $x, y, z \in Z_q^*$ to determine xyP.
- Decisional Diffie-Hellman (DDH) problem: Assume P, Px, Py, Pz for any random integer $x, y, z \in Z_q^*$ to determine whether $zP = xyP\ or\ z = xy\,(mod\ q)$.
- Bilinear Diffie-Hellman (BDH) problem: Assume $\mathrm{P, Px, Py, Pz}$ for any random integer $\mathrm{x, y, z} \in \mathrm{Z_q^*}$ to determine $\hat{\mathrm{e}}(\mathrm{P, P})^{\mathrm{xyz}}$.

IV. SEAMLESS SECURE AUTHENTICATION AND KEY AGREEMENT (S-SAKA) FRAMEWORK

In order to resolve the problem of security issue between mobile-sink and cluster-head, a framework of S-SAKA is proposed using bilinear-pairing. The S-SAKA framework is composed of seven phases: Initialization; System registration; Cluster-Head Registration; Mobile-Sink Registration; System Login; Authentication; Extraction of Sensing Data and secret key update. The mechanisms of S-SAKA are discussed in the following subsections.

A. INITIALIZATION PHASE

In this phase, base station $\mathrm{B_{St}}$ performs an initialization to generate the prerequisite parameters keys to publish the system requirements. The procedural steps are as follows:

Step 1: Bilinear parameters are generated $\{q, P, G, G_T, \hat{e}\}$, where G is a cyclic additive group that generate prime order integers q by P and $\hat{e} : G \times G \rightarrow G_T$ is a bilinear group map.

Step 2: After the generation of bilinear parameters, $\mathrm{B_{St}}$ chooses a random integer $S \in Z_q^*$ as its corresponding master key to compute its public key p_{pub} using $p_{pub} = S.P$.

Step 3: After the generation of public key p_{pub}, $\mathrm{B_{St}}$ determines two secure collision resistance hashing operator H_1 and H_2, where $H_1\ :\ \{0, 1\}^*$ is called as map to point hashing functional operation and $H_2\ :\ \{0, 1\}^*$ is a one way secure hashing.

Step 4: After the determination of hashing function, $\mathrm{B_{St}}$ selects a symmetric encryption $E_{S_k}\,(.)$ as the system encryption function.

Step 5: Lastly, $\mathrm{B_{St}}$ publishes the system parameters $\{q, P, G, G_T, \hat{e}, P, p_{pub}, H_1, H_2, E_{S_k}\,(.)\}$ and keeps the secret parameters $\{S, x, y\}$ confidentially.

The procedural steps can be represented as an illustrative diagram as follows:

$\mathrm{B_{St}}$ performs an initialization

$$G \text{ generates} \begin{cases} q \text{ by } P \\ \hat{e} : G \times G \rightarrow G_T \end{cases} \rightarrow \{q, P, G, G_T, \hat{e}\}$$

$$\mathrm{B_{St}} \text{ chooses a random integer: } S \in Z_q^* \xrightarrow{Compute} p_{pub}$$

$$p_{pub} = S.P$$

$$\mathrm{B_{St}} \text{ two secure collision resistance: } \begin{cases} H_1 : \{0, 1\}^* \\ H_2 : \{0, 1\}^* \end{cases}$$

$$\mathrm{B_{St}} \xrightarrow[\text{encryp}]{\text{symmetric}} E_{S_k}\,(.)$$

$$\mathrm{B_{St}} \begin{cases} sys\ para : \{q, P, G, G_T, \hat{e}, P, p_{pub}, H_1, H_2, E_{S_k}\,(.)\} \\ secret\ para : \{S, x, y\} \end{cases}$$

B. SYSTEM REGISTRATION PHASE

Prior to the deployment of the system, the available mobile sinks and cluster-head should be registered with the base-station. This phase is composed of two-parts, namely cluster-head and mobile-sink registrations. In the S-SAKA framework, cluster-head and mobile-sink have its own unique identity, as CID_i and PID_j respectively. Moreover, the base-station B_{St} has a database table DB_S which is initialized to maintain the non-compromised cluster-head.

1) CLUSTER-HEAD REGISTRATION

Before the deployment of cluster-head $\mathrm{CH_i}$ In WSNs', the base-station integrate a unique identity $\mathrm{CID_i}$ to compute $\mathrm{S.H_1(CID_i)}$, and then, the base-station stores back the computed value of $\mathrm{S.H_1(CID_i)}$ in to the memory of $\mathrm{CH_i}$. Eventually, the base-station places $\mathrm{CH_i}$ at an appropriate position and insert a new identifier $\mathrm{CID_i}$ in the table of $\mathrm{DB_S}$.

The cluster-head registration can also be represented as an illustrative diagram as follows:

$$\mathrm{B_{St}}\ \text{integrate}\ \mathrm{CID_I} \xrightarrow{Compute} \mathrm{S.H_1(CID_i)}.$$
$$\mathrm{B_{St}} \xrightarrow[\mathrm{S.H_1(CID_i)}]{Store} \mathrm{CH_i}.$$
$$\mathrm{DB_S} \xleftarrow[\mathrm{CID_i}]{} \mathrm{CH_i}.$$

2) MOBILE-SINK REGISTRATION

In this phase, the mobile-sink M_S which is authorized would try to store the random integer x into smart card. The procedural steps are as follow:

Step 1: The authorized mobile-sink can freely opt an identity PID_j, a secret-key SK_j and a random integer $x \in Z_q^*$. Afterwards, the mobile-sink determines $H_2(x \oplus SK_j)$ and sends the message-request $\{PID_j, H_2(x \oplus SK_j)\}$ to $\mathrm{B_{St}}$ via a secure communication channel.

Step 2: After receiving the message-request $\{PID_j, H_2(x \oplus SK_j)\}$, the $\mathrm{B_{St}}$ determines the following expressions: $Certify_j = S.H_1(PID_j \parallel H_2(x \oplus SK_j))$; $TS_j = H_2(PID_j \parallel y)$; $H_j = H_2(TS_j)$; $V_j = TS_j \oplus H_2(x \oplus SK_j))$; $A_j = H_2(PID_j \parallel x \parallel y)$. Then, $\mathrm{B_{St}}$ delivers a smart-card which is integrated of $\{Certify_j, V_j, H_j, A_j\}$ to $\mathrm{M_S}$ via a secure communication channel.

Step 3: After receiving the smart-card, the authorized mobile-sink stores back the random-integer x securely in the smart-card. Now, the smart-card is integrated of $\{Certify_j, V_j, H_j, A_j, x\}$.

The following illustrative diagram shows mobile-sink registration over base station.

$$M_S : \begin{cases} PID_j \\ SK_j \\ x \in Z_q^* \end{cases}$$
$$\mathrm{Commit} = H_2(x \oplus SK_j)$$
$$M_S \xrightarrow{\{PID_j, \mathrm{Commit}\}} \mathrm{B_{St}}$$

$$\left.\begin{array}{l} Certify_j = S.H_1\left(PID_j \parallel H_2\left(x \oplus SK_j\right)\right) \\ TS_j = H_2\left(PID_j \parallel y\right) \\ H_j = H_2\left(TS_j\right) \\ V_j = TS_j \oplus H_2(x \oplus SK_j)) \\ A_j = H_2\left(PID_j \parallel x \parallel y\right) \end{array}\right\} : \mathrm{B_{St}}$$
$$M_S \xleftarrow[\{Certify_j, V_j, H_j, A_j\}]{Smart-card} \mathrm{B_{St}}$$
$$M_S : \{Certify_j, V_j, H_j, A_j, x\}$$

C. SYSTEM LOGIN PHASE

When any user wishes to use $\mathrm{M_S}$ to sense data in an appropriate network, then the user should insert a valid smart-card into the terminal to enter the credentials, such as identity $\mathrm{PID_j}$ and secret-key $\mathrm{SK_j}$. The execution flows are as follows:

Step 1: The smart-card $\mathrm{S_C}$ determines $\mathrm{TS_j^*} = \mathrm{V_j} \oplus \mathrm{H_2}\left(\mathrm{PID_j} \parallel \mathrm{H_2}\left(\mathrm{x} \oplus \mathrm{SK_j}\right)\right)$ and $\mathrm{H_j^*} = \mathrm{H_2}\left(\mathrm{TS_j^*}\right)$ to verify whether $\mathrm{H_j^*}$ is equal to $\mathrm{H_j}$ or not. If the equality holds, then $\mathrm{M_S}$ executes the subsequent steps. Otherwise, $\mathrm{S_C}$ requests the user to provide the proper credentials, such as user identity and secret key.

Step 2: The smart-card determines a random-integer value $\mathrm{rN_j}$ to determine the following:

$$\mathrm{b_j} = \mathrm{H_2}\left(\mathrm{TS_j^*} \parallel \mathrm{rN_j}\right) \oplus \mathrm{H_2}\left(\mathrm{x} \oplus \mathrm{SK_j}\right);$$
$$\mathrm{c_j} = \mathrm{H_2}\left(\mathrm{A_j} \parallel \mathrm{H_2}\left(\mathrm{x} \oplus \mathrm{SK_j}\right) \parallel \mathrm{rN_j}\right),$$

then the mobile-sink sends the message-request $\{\mathrm{PID_j}, \mathrm{b_j}, \mathrm{c_j}, \mathrm{rN_j}\}$ to $\mathrm{B_{St}}$ via secure common channel.

Step 3: Upon receiving the message $\{\mathrm{PID_j}, \mathrm{b_j}, \mathrm{c_j}, \mathrm{rN_j}\}$ from smart-card, $\mathrm{B_{St}}$ performs the computation

$$\mathrm{A_j^*} = \mathrm{H_2}\left(\mathrm{PID_j} \parallel \mathrm{x} \parallel \mathrm{y}\right);$$
$$\mathrm{D_j} = \mathrm{b_j} \oplus \mathrm{H_2}\left(\mathrm{H_2}\left(\mathrm{PID_j} \parallel \mathrm{x}\right) \parallel \mathrm{rN_j}\right) = \mathrm{H_2}\left(\mathrm{x} \oplus \mathrm{SK_j}\right);$$
$$\mathrm{c_j^*} = \mathrm{H_2}\left(\mathrm{A_j^*} \parallel \mathrm{D_j} \parallel \mathrm{rN_j}\right).$$

After computation, $\mathrm{B_{St}}$ verifies whether $\mathrm{c_j^*} = \mathrm{c_j}$ or not. If the equality holds, then $\mathrm{B_{St}}$ determines $\mathrm{M_S}$ as a legal mobile-sink to compute

$$\mathrm{Q_j} = \mathrm{H_2}\left(\left(\mathrm{PID_j} \parallel \mathrm{H_2}\left(\mathrm{x} \oplus \mathrm{SK_j}\right)\right) \parallel \mathrm{rN_j}\right),$$
$$\mathrm{M_K}\hat{\mathrm{e}}\left(\mathrm{S.H_1}\left(\mathrm{PID_j} \parallel \mathrm{H_2}\left(\mathrm{x} \oplus \mathrm{SK_j}\right)\right), \mathrm{p_{pub}}\right),\ \text{and}$$
$$\mathrm{CLC_{DB_S}} = \mathrm{E_{M_K}}\left(\mathrm{LC_{DB_S}}\right).$$

Lastly, $\mathrm{B_{St}}$ sends the computation parameters $\{\mathrm{Q_j}, \mathrm{CLC_{DB_S}}\}$ to $\mathrm{M_S}$.

Step 4: Upon receiving the parameters $\{\mathrm{Q_j}, \mathrm{CLC_{DB_S}}\}$ from $\mathrm{B_{St}}$, $\mathrm{M_S}$ determines $\mathrm{Q_j^*} = \mathrm{H_2}\left(\left(\mathrm{PID_j} \parallel \mathrm{H_2}\left(\mathrm{x} \oplus \mathrm{SK_j}\right)\right) \parallel \mathrm{rN_j}\right)$ to verify whether $\mathrm{Q_j^*} = \mathrm{Q_j}$ or not. If the equation holds, then $\mathrm{M_S}$ computes $\mathrm{M_K} = \hat{\mathrm{e}}\left(\mathrm{cert_j}, \mathrm{p_{pub}}\right)$ and $\mathrm{LC_{DB_S}} = \mathrm{D_{M_K}}\left(\mathrm{CLC_{DB_S}}\right)$, where $\mathrm{D_{M_K}}$ is the decryption of $\mathrm{E_{M_K}}(.)$.

After the successful computation, $\mathrm{M_S}$ stores the updated list of legal cluster heads $\mathrm{LC_{DB_S}}$ into its storage memory. Upon the successful of $\mathrm{LC_{DB_S}}$, $\mathrm{M_S}$ retains the complete database of legal cluster heads to run the subsequent step to ensure the authentication during message transmission.

Moreover, to obtain the legal cluster head, the above steps are periodically executed.

The steps of system login phase can be presented as:

$$\mathrm{S_C}\text{ determines: }\begin{cases}\mathrm{TS_j^*} = \mathrm{V_j} \oplus \mathrm{H_2}\left(\mathrm{PID_j} \parallel \mathrm{H_2}\left(\mathrm{x} \oplus \mathrm{SK_j}\right)\right)\\ \mathrm{H_j^*} = \mathrm{H_2}\left(\mathrm{TS_j^*}\right)\end{cases}$$

if $\mathrm{H_j^*} \neq \mathrm{H_j}$ then

$$\mathrm{S_C} \xrightarrow[\mathrm{PID_j},\ \mathrm{SK_j}]{requests} user$$

else

$$\mathrm{S_C}\text{ determines } \mathrm{rN_j}: \begin{cases}\mathrm{b_j} = \mathrm{H_2}\left(\mathrm{TS_j^*} \parallel \mathrm{rN_j}\right) \oplus \mathrm{H_2}\left(\mathrm{x} \oplus \mathrm{SK_j}\right)\\ \mathrm{c_j} = \mathrm{H_2}\left(\mathrm{A_j} \parallel \mathrm{H_2}\left(\mathrm{x} \oplus \mathrm{SK_j}\right) \parallel \mathrm{rN_j}\right)\end{cases}$$

$$M_S \xrightarrow[\{\mathrm{PID_j, b_j, c_j, rN_j}\}]{} \mathrm{B_{St}}$$

$$\left.\begin{array}{c}\mathrm{A_j^*} = \mathrm{H_2}\left(\mathrm{PID_j} \parallel \mathrm{x} \parallel \mathrm{y}\right)\\ \mathrm{D_j} = \mathrm{b_j} \oplus \mathrm{H_2}\left(\mathrm{H_2}\left(\mathrm{PID_j} \parallel \mathrm{x}\right) \parallel \mathrm{rN_j}\right) = \mathrm{H_2}\left(\mathrm{x} \oplus \mathrm{SK_j}\right)\\ \mathrm{c_j^*} = \mathrm{H_2}\left(\mathrm{A_j^*} \parallel \mathrm{D_j} \parallel \mathrm{rN_j}\right)\end{array}\right\} : \mathrm{B_{St}}$$

if $\mathrm{c_j^*} = \mathrm{c_j}$ then

$$M_S : \left\{\begin{array}{l}\mathrm{Q_j} = \mathrm{H_2}\left(\left(\mathrm{PID_j} \parallel \mathrm{H_2}\left(\mathrm{x} \oplus \mathrm{SK_j}\right)\right) \parallel \mathrm{rN_j}\right)\\ \mathrm{M_K}\hat{e}\left(\mathrm{S.H_1}\left(\mathrm{PID_j} \parallel \mathrm{H_2}\left(\mathrm{x} \oplus \mathrm{SK_j}\right)\right), \mathrm{p_{pub}}\right)\\ \mathrm{CLC_{DB_S}} = \mathrm{E_{M_K}}\left(\mathrm{LC_{DB_S}}\right)\end{array}\right\} : \mathrm{B_{St}}$$

$$M_S \xleftarrow[\left\{\mathrm{Q_j, CLC_{DB_S}}\right\}]{} \mathrm{B_{St}}$$

$$M_S\text{ determines: } \mathrm{Q_j^*} = \mathrm{H_2}\left(\left(\mathrm{PID_j} \parallel \mathrm{H_2}\left(\mathrm{x} \oplus \mathrm{SK_j}\right)\right) \parallel \mathrm{rN_j}\right)$$

$$\begin{array}{c} \textit{if } \mathrm{Q_j^*} = \mathrm{Q_j}\text{ then}\\ \mathrm{M_K} = \hat{e}\left(\mathrm{cert_j}, \mathrm{p_{pub}}\right)\\ \mathrm{LC_{DB_S}} = \mathrm{D_{M_K}}\left(\mathrm{CLC_{DB_S}}\right)\end{array}$$

D. SYSTEM AUTHENTICATION PHASE

Once the system login phase is completed successfully, M_S can move into the WSN's coverage area to collect the sensing data. In order to authenticate its communication with CH_j, M_S has the procedural executions, which are as follows:

Step 1: While M_S actuates its current vicinity into CH_j, it transmits its connection-request to nearby CH_j for user authentication.

Step 2: After receiving the connection-request, CH_j sends its unique identity of CID_j to the requested mobile-sink.

Step 3: Upon receiving the CID_j, M_S verifies the legitimacy of CH_j using the un-compromised cluster database table LC_{DB_S}. If CH_j is found as legal, then M_S generates the following: $H_t = H_2(TS_S)$; $m_1 = H_t.H_1(PID_j \parallel H_2(x \oplus SK_j)$; and $\Delta_j = \hat{e}\left(Certify_j.H_t.H_1\left(CID_j\right)\right)$. Then, the mobile-sink sends the message-request $\left\{m_1, \Delta_j, TS_S\right\}$ to CH_j. Eventually, the mobile-sink determines an initial session key $S_{K1} = \hat{e}(H_t.Certify_j, H_t.H_1\left(CID_j\right))$ as a secret-session key.

Step 4: After the successful computation of $\left\{m_1, \Delta_j, H_t\right\}$, CH_j verifies whether $TS_S - TS_C \leq \Delta TS$, where TS_C is the current timestamp of CH_j message transmission and ΔTS is the expected transmission delay.

If the delay interval is permissible, then CH_j determines

$$H_t = H_2\left(TS_S\right) \text{ and } \Delta_j^* = \hat{e}\left(Certify_j.H_t.H_1\left(CID_j\right)\right)$$

to check whether $\Delta_j^* = \Delta_j$ or not. If the equation holds, then CH_j identifies M_S to be legitimate. To create a common session key, CH_j computes a final session key

$$\begin{aligned} S_{k2} &= \hat{e}\left(m_1, H_t.\mathrm{r}.H_1\left(CID_j\right)\right) \text{ and}\\ \nabla &= H_2\left(S_{k1} \parallel PID_j \parallel CID_j \parallel TS_S\right)\end{aligned}$$

and then sends the message parameter $\{\nabla, TS_S\}$ to M_S.

To verify the identical in session key, M_S computes the equation that is as follows:

$$\begin{aligned} S_{k2} &= \hat{e}\left(m_1, H_t.\mathrm{r}.H_1\left(CID_j\right)\right)\\ &= \hat{e}\left(H_t.H_1\left(PID_j \parallel H_2(x \oplus SK_j)\right),\quad H_t.\mathrm{r}.H_1\left(CID_j\right)\right.\\ &= \hat{e}\left(H_t.\mathrm{r}.H_1\left(PID_j \parallel H_2(x \oplus SK_j)\right),\quad H_t.H_1\left(CID_j\right)\right.\\ &= \hat{e}\left(H_t.Certify_j, H_t.H_1\left(CID_j\right)\right) = S_{K1}\end{aligned}$$

Step 5: Once the message parameter $\{\nabla, TS_S\}$ is received from CH_j, M_S determines $V_{erify} = H_2\left(S_{k1} \parallel PID_j \parallel CID_j \parallel TS_S\right)$ to check whether it holds with ∇ or not. If the verification is successful, M_S uses the session key S_{k1} to establish a session with CH_j.

The steps of system authentication phase can be presented as:

$$M_S \xrightarrow{connection-request} CH_j$$
$$\xleftarrow{CID_j}$$

M_S *checks*

CH_j in LC_{DB_S}

if CH_j is legal then

$$M_S : \begin{cases} H_t = H_2\left(TS_S\right)\\ m_1 = H_t.H_1(PID_j \parallel H_2(x \oplus SK_j)\\ \Delta_j = \hat{e}\left(Certify_j.H_t.H_1\left(CID_j\right)\right)\end{cases}$$

$$M_S \xrightarrow[\{m_1, \Delta_j, TS_S\}]{message-request} CH_j$$

$$S_{K1} = \hat{e}(H_t.Certify_j, H_t.H_1\left(CID_j\right))$$

$$TS_S - TS_C \leq \Delta TS\} : CH_j$$

$$\left.\begin{array}{l} H_t = H_2\left(TS_S\right)\\ \Delta_j^* = \hat{e}\left(Certify_j.H_t.H_1\left(CID_j\right)\right)\end{array}\right\} : CH_j$$

if $\Delta_j^* = \Delta_j$ then

$$M_S \xleftarrow{identifies} CH_j$$

Computes

$$\left.\begin{array}{l} S_{k2} = \hat{e}\left(m_1, H_t.\mathrm{r}.H_1\left(CID_j\right)\right)\\ \nabla = H_2\left(S_{k1} \parallel PID_j \parallel CID_j \parallel TS_S\right)\end{array}\right\} : CH_j$$

$$M_S \xleftarrow{\{\nabla, TS_S\}} CH_j$$

$$\begin{aligned} S_{k2} &= \hat{e}\left(m_1, H_t.\mathrm{r}.H_1\left(CID_j\right)\right)\\ &= \hat{e}\left(H_t.H_1\left(PID_j \parallel H_2(x \oplus SK_j)\right),\quad H_t.\mathrm{r}.H_1\left(CID_j\right)\right.\\ &= \hat{e}\left(H_t.\mathrm{r}.H_1\left(PID_j \parallel H_2(x \oplus SK_j)\right),\quad H_t.H_1\left(CID_j\right)\right.\end{aligned}$$

$$= \hat{e}\left(H_t.Certify_j, H_t.H_1\left(CID_j\right)\right) = S_{K1}$$
$$M_S{:}V_{erify}$$
$$= H_2\left(S_{k1} \parallel PID_j \parallel CID_j \parallel TS_S\right)$$

E. EXTRACTION OF SENSING DATA

While M_S is successfully established its communication with CH_j, it can read the sensing data from CH_j. The processing steps are as follows:

Step 1: Initially, CH_j computes the cipher text $CT_j = E_{S_{k2}}\left(Data_j\right)$ using the session key S_{k2}, where $Data_j$ is the internal storage data of CH_j. Lastly, CH_j sends the parameter CT_j to M_S.

Step 2: After receiving the parameter CT_j from CH, M_S determines $CData_j = E_{M_K}\left(D_{S_{k1}}\left(CT_j\right)\right)$ to stores its corresponding values in internal storage device or sends the value to B_{St} directly, where $M_K = \hat{e}\left(cert_j, p_{pub}\right)$. The aforesaid equation may be inferred as: $CData_j = E_{M_K}\left(D_{S_{k1}}\left(CT_j\right)\right) = CData_j = E_{M_K}\left(D_{S_{k1}}\left(E_{S_{k2}}\left(Data_j\right)\right)\right) = E_{M_K}\left(Data_j\right)$. Hence, the verification proves that the storage data of M_S is encrypted by M_K.

Step 3: After successful verification, M_S sends the transmission message $\left\{PID_j, CData_j, b_j, c_j, rN_j\right\}$ to B_{St}, where b_j, c_j *and* rN_j are computed in system login phase.

Step 4: Upon receiving the transmission message from M_S, B_{St} tries to extract PID_j to compute the equations, which are as follows: $A_j^* = H_2\left(PID_j \parallel x \parallel y\right)$; $D_j = H_2\left(x \oplus SK_j\right)$ and $c_j^* = H_2\left(A_j^* \parallel D_j \parallel rN_j\right)$.

After the successful computation, B_{St} verifies whether $c_j^* = c_j$ or not. If the equation holds, then B_{St} determines M_S to be a legitimate. Then, B_{St} determines $M_K = \hat{e}\left(S.H_1\left(PID_j \parallel H_2\left(x \oplus SK_j\right)\right), p_{pub}\right)$ to decrypt the storage-data $D_{M_K}\left(Data_j\right) = D_{M_K}\left(E_{M_K}\left(Data_j\right)\right) = Data_j$. After decryption of storage data, B_{St} is allowed to extract the sensing data $Data_j$ from M_S.

The following diagram shows extraction of sensing data.

$$CT_j = \left.E_{S_{k2}}\left(Data_j\right)\right\} : CH_j$$
$$M_S \xleftarrow{CT_j} CH_j$$

M_S determines

$$CData_j = E_{M_K}\left(D_{S_{k1}}\left(CT_j\right)\right)$$
$$M_S \xrightarrow[sends]{M_K=\hat{e}\left(cert_j, p_{pub}\right)} B_{St}$$
$$CData_j = E_{M_K}\left(D_{S_{k1}}\left(CT_j\right)\right) = CData_j$$
$$= E_{M_K}\left(D_{S_{k1}}\left(E_{S_{k2}}\left(Data_j\right)\right)\right) = E_{M_K}\left(Data_j\right)$$
$$M_S \xrightarrow{\left\{PID_j, CData_j, b_j, c_j, rN_j\right\}} B_{St}$$
$$\left.\begin{array}{l} A_j^* = H_2\left(PID_j \parallel x \parallel y\right) \\ D_j = H_2\left(x \oplus SK_j\right) \\ c_j^* = H_2\left(A_j^* \parallel D_j \parallel rN_j\right) \end{array}\right\} : B_{St}$$

if $c_j^* = c_j$ *then* B_{St} determines M_S

$$M_K = \hat{e}\left(S.H_1\left(PID_j \parallel H_2\left(x \oplus SK_j\right)\right), p_{pub}\right)$$
$$D_{M_K}\left(Data_j\right) = D_{M_K}\left(E_{M_K}\left(Data_j\right)\right) = Data_j$$

F. SECRET KEY UPDATE PHASE

In this phase, U_{ser} can modify his / her secret key when he / she wants to change. The procedural steps of key update phase are as follows:

Step 1: U_{ser} tries to enter his / her smart card into the user terminal to verify the credentials, such as PID_j and SK_j.

Step 2: After the successful entries, the smart card computes $TS_j^* = Ver_j \oplus H_2\left(PID_j \parallel H_2\left(x \oplus SK_j\right)\right)$ and $H_j^* = H_2\left(TS_j^*\right)$ to verify whether $H_j^* = H_j$ or not. If the equation holds, then U_{ser} is permitted to change his /her secret key SK_j^{new} and x^{new}. Otherwise, the smart card disapproves the request of U_{ser}. Lastly, M_S sends the transmission message $\left\{PID_j \parallel H_2\left(x \oplus SK_j\right), Ver_j, H_2\left(x^{new} \oplus SK_j^{new}\right)\right\}$ to B_{St} through secure communication channel.

Step 3: After receiving the transmission message $\left\{PID_j \parallel H_2\left(x \oplus SK_j\right), Ver_j, H_2\left(x^{new} \oplus SK_j^{new}\right)\right\}$ from M_S, B_{St} determines $Ver_j^* = H_2\left(PID_j \parallel y\right) \oplus H_2\left(PID_j \parallel H_2\left(x \oplus SK_j\right)\right)$ to verify whether $Ver_j^* = Ver_j$ or not. If the equation holds, then B_{St} performs a computation of $cert_j^{new} = r.H_1\left(PID_j\right) \parallel H_2\left(x^{new} \oplus SK_j^{new}\right)$ and $Ver_j^{new} = H_2\left(PID_j \parallel y\right) \oplus H_2\left(PID_j \parallel H_2\left(x^{new} \oplus SK_j^{new}\right)\right)$. Then, B_{St} sends the computation message $\left\{cert_j^{new}, Ver_j^{new}\right\}$ to M_S through a secure communication channel.

Step 4: After receiving the message $\left\{cert_j^{new}, Ver_j^{new}\right\}$ from B_{St}, the smart card modifies the parameters, such as $cert_j, verf_j$ *and* x into $cert_j^{new}, Ver_j^{new}$ and x^{new} in the given order.

The following illustrative diagram shows secret key update phase.

U_{ser} verify the credentials PID_j and SK_j

$$\left.\begin{array}{l} TS_j^* = Ver_j \oplus H_2\left(PID_j \parallel H_2\left(x \oplus SK_j\right)\right) \\ H_j^* = H_2\left(TS_j^*\right) \end{array}\right\} : \text{ smart card}$$

if $H_j^* = H_j$ *then*

U_{ser} is permitted to change his /her secret key SK_j^{new} and x^{new}

$$M_S \xrightarrow{\left\{PID_j \parallel H_2\left(x \oplus SK_j\right), Ver_j, H_2\left(x^{new} \oplus SK_j^{new}\right)\right\}} B_{St}$$
$$\left.Ver_j^* = H_2\left(PID_j \parallel y\right) \oplus H_2\left(PID_j \parallel H_2\left(x \oplus SK_j\right)\right)\right\} : B_{St}$$

if $Ver_j^* = Ver_j$ *then*

$$\left.\begin{array}{l} cert_j^{new} = r.H_1\left(PID_j\right) \parallel H_2\left(x^{new} \oplus SK_j^{new}\right) \\ Ver_j^{new} = H_2\left(PID_j \parallel y\right) \oplus H_2\left(PID_j \parallel H_2\left(x^{new} \oplus SK_j^{new}\right)\right) \end{array}\right\} : B_{St}$$
$$M_S \xrightarrow{\left\{cert_j^{new}, Ver_j^{new}\right\}} B_{St}$$

TABLE 2. Important notation used in BAN logic.

Notation	Description
$X\mid\equiv P$:	X relies on a statement of P
$\neq P$:	P be sure as fresh
$X\mid\Rightarrow P$:	X takes the jurisdiction over P
$X \triangleleft P$:	X realizes P
$X\mid\sim P$:	X formerly believed as P
(P,Q):	P *or* Q is an individual part of (P,Q)
$\{P\}_{s_k}$:	P is encrypted using secret ket s_k
$\langle P\rangle^{Q}_{s_k}$:	P is mutually shared with Q
$X \overset{s_k}{\leftrightarrow} Y$:	X *and* Y uses a secret-key s_k to establish a communication. Besides, s_k is totally secure; and thus can not be discovered by any principal excluding X *and* Y.

$$M_S modifies : \begin{cases} cert_j, verf_j = cert_j^{new}, Ver_j^{new} \\ x = x^{new} \end{cases}$$

V. SECURITY ANALYSIS

This section is composed of stringent formal and informal security analysis of S-SAKA. The analysis result shows that the proposed S-SAKA framework not only offers security properties of authentication protocols for mutual authentication, session-key agreement and data confidentiality, but also prevents the various potential attacks, such as node-capture, stolen smart-card, key impersonation and privileged-insider.

A. FORMAL SECURITY ANALYSIS

S-SAKA framework offers secret session-key agreement between a legal cluster head CH_i, base station B_S, smart card S_C and a mobile-sink MS_i and it is proven using BAN logic [19]. Assume that X *and* Y be the principles, P *and* Q be the statement / formula and s_k be the secret key. The important notation used in the BAN logic is given in Table 2.

The BAN logic postulates are as follows:

Rule 1 – Meaning of Messages: $\frac{X\mid\equiv X \overset{s_k}{\longleftrightarrow} Y, X\triangleleft\{P\}_{s_k}}{X\mid\equiv Y\mid\sim P}$ and $\frac{X\mid\equiv X \overset{Q}{\longleftrightarrow} Y, X\triangleleft\{P\}_{Q}}{X\mid\equiv Y\mid\sim Q}$: If X trusts that s_k is shared among X *and* Y and perceives P encrypted with s_k, then X trusts the Y as a legal client.

Rule 2 – Verification of Nonce: $\frac{X\mid\equiv\neq P, X\mid\equiv Y\mid\sim P}{X\mid\equiv Y\mid\equiv P}$ and $\frac{X\mid\equiv\neq Q, X\mid\equiv Y\mid\sim Q}{X\mid\equiv Y\mid\equiv Q}$: If X trusts that X has just been communicated and thus Y ony perceives P, then X trusts that Y be certain of P.

Rule 3 – Belief: $\frac{X\mid\equiv P X\mid\equiv Q}{X\mid\equiv(P,Q)}$: If X trusts P *and* Q, then X beliefs in P *and* Q.

Rule 4 – Rule of Fresh-Concatenation: $\frac{X\mid\equiv\neq P}{X\mid\equiv\neq(P,Q)}$: If X trusts the freshness in key generation of P, then Y be certain of freshness in (P, Q).

Rule 5 – Rule of Jurisdiction: $\frac{X\mid\equiv Y\Rightarrow P X\mid\equiv Y\mid\equiv P}{X\mid\equiv(P,Q)}$: If X trusts that Y has influence over P and X believes Y in the accuracy of P, then X trusts in P.

To satisfy the security properties of AKA protocol, the proposed S-SAKA framework must be able to meet all the test goals, given in below.

$$Goal_1 : M_{S_i}\mid\equiv B_S\mid\equiv CH_i \overset{s_k}{\longleftrightarrow} S_C$$

$$Goal_2 : M_{S_i}\mid\equiv CH_i \overset{s_k}{\longleftrightarrow} S_C$$

$$Goal_3 : S_C\mid\equiv M_{S_i}\mid\equiv CH_i \overset{s_k}{\longleftrightarrow} S_C$$

$$Goal_4 : S_C\mid\equiv CH_i \overset{s_k}{\longleftrightarrow} S_C$$

The structural flow of BAN logic is as follows:

1. Messages in Generic Form:

$$M_1 : M_{S_i}\rightarrow B_S : \langle H_2(x\oplus SK_j)\rangle, \langle PID_j, H_2(x\oplus SK_j)\rangle_{x\in Z_q^*}$$

$$M_2 : B_S \rightarrow S_C : \langle Certify_j = S.H_1(PID_j\|H_2(x\oplus SK_j));$$
$$TS_j = H_2(PID_j\|y);\ H_j = H_2(TS_j);$$
$$V_j = TS_j \oplus H_2(x\oplus SK_j));$$
$$A_j = H_2(PID_j\|x\|y)\rangle_{x\in Z_q^*}$$

$$M_3 : M_{S_i} \rightarrow S_C : \langle Certify_j, V_j, H_j, A_j, x\rangle_{x\in Z_q^*}$$

2. Transmission of Messages in Idealized Form:

$$T_{M1} : M_{S_i} \rightarrow B_S : \langle PID_j, H_2(x\oplus SK_j)\rangle_{MS_i \overset{PID_j}{\longrightarrow} B_S}$$

$$T_{M2} : B_S \rightarrow S_C : \langle Certify_j, V_j, H_j, A_j\rangle_{MS_i \overset{PID_j}{\longrightarrow} B_S}$$

$$T_{M3} : M_{S_i} \rightarrow S_C : \langle Certify_j, V_j, H_j, A_j, x\rangle_{MS_i \overset{PID_j}{\longrightarrow} B_S}$$

3. Messages in Hypotheses Form:

$$H_{M1} : M_{S_i}\mid\equiv\neq(CID_i),\quad CH_i\mid\equiv\neq(TS_1, TS_2)$$

$$H_{M2} : B_S\mid\equiv\neq(PID_i),\quad B_S\mid\equiv\neq(TS_3, TS_4)$$

$$H_{M3} : M_{S_i}\mid\equiv B_S\mid\equiv CH_i \overset{s_k}{\longleftrightarrow} S_C$$

$$H_{M4} : M_{S_i}\mid\equiv CH_i \overset{s_k}{\longleftrightarrow} S_C$$

$$H_{M5} : S_C\mid\equiv MS_i\mid\equiv CH_i \overset{s_k}{\longleftrightarrow} S_C$$

$$H_{M6} : S_C\mid\equiv CH_i \overset{s_k}{\longleftrightarrow} S_C$$

The idealized form of S-SAKA framework is examined on the base of BAN logic postulates and goal settings. The key proofs of S-SAKA are as follows:

- From the transmission message T_{M1}, the S-SAKA has $P_1 : B_S \triangleleft \langle PID_j, H_2(x\oplus SK_j)\rangle_{MS_i \overset{PID_j}{\longrightarrow} B_S}$.
- From H_{M2}, P_1 and *Rule* (1), the S-SAKA acquires $P_2 : B_S\mid\equiv M_{S_i}\mid\sim\langle PID_j, H_2(x\oplus SK_j)\rangle$.
- From the transmission message T_{M2}, the S-SAKA has $P_3 : S_C \triangleleft \langle Certify_j, V_j, H_j, A_j\rangle_{MS_i \overset{PID_j}{\longrightarrow} B_S}$.
- From H_{M5}, P_3 and *Rule* (1), the S-SAKA acquires $P_4 : CH_i\mid\equiv S_C\mid\sim\langle Certify_j, V_j, H_j, A_j\rangle$.
- From $H_{M1}, P_4, Rule\,(2)$ and *Rule* (4), the S-SAKA obtains $P_5 : M_{S_i}\mid\equiv B_S\mid\equiv CH_i \overset{s_k}{\longleftrightarrow} S_C$. $\langle\mathbf{Goal_1}\rangle$
- Try, from H_{M5}, P_5 and *Rule* (1), the S-SAKA gets $P_6 : M_{S_i}\mid\equiv CH_i \overset{s_k}{\longleftrightarrow} S_C$. $\langle\mathbf{Goal_2}\rangle$
- From the transmission message T_{M3}, the S-SAKA has $P_7 : B_S \triangleleft \langle Certify_j, V_j, H_j, A_j, x\rangle_{MS_i \overset{PID_j}{\longrightarrow} B_S}$.
- From H_{M1}, P_7 and *Rule* (1), the S-SAKA acquires $P_8 : B_S\mid\equiv MS_i\mid\sim\langle Certify_j, V_j, H_j, A_j, x\rangle$.
- From $H_{M5}, P_8, Rule\,(2)$ and *Rule* (4), the S-SAKA obtains $P_9 : S_C\mid\equiv M_{S_i}\mid\equiv CH_i \overset{s_k}{\longleftrightarrow} S_C$. $\langle\mathbf{Goal_3}\rangle$
- From H_{M4}, P_9 and *Rule* (3), the S-SAKA achieves $P_{10} : S_C\mid\equiv CH_i \overset{s_k}{\longleftrightarrow} S_C$. $\langle\mathbf{Goal_4}\rangle$

- Again, from H_{M6}, P_{10} and *Rule* (5), the S-SAKA produces $P_{11} : M_{S_i}| \equiv CH_i \stackrel{s_k}{\longleftrightarrow} S_C \langle \boldsymbol{Goal_2} \rangle$

Provided the goals $\langle \boldsymbol{Goal_1} - \boldsymbol{Goal_4} \rangle$, the S-SAKA protocol asserts that it uses a shared secret-key s_k to establish a communication; and hence the proposed S-SAKA framework is proficient to achieve the proper mutual authentication, session-key agreement and confidentiality.

B. INFORMAL SECURITY ANALYSIS

In this subsection, the informal security analysis of S-SAKA protocol is performed in which the adversary has some unique capabilities that are as follows:

1. The adversary is able to control over the communication channel especially with mobile-sink, cluster-head and base-station to do message intercept, insert, delete or modify any exchange of information.
2. The adversary may incur either user identity and secret key or the storage information of smart card but he / she cannot obtain both. For an instance, if the adversary obtains the user identity and secret key, he / she can't have any chance to obtain the storage information of smart card.

1) PROPER MUTUAL AUTHENTICATION AND SESSION-KEY AGREEMENT

In the authentication phase, the cluster-head CH_j and mobile-sink authenticate each other by the verification of $\Delta_j = \hat{e}(Certify_j.H_t.H_1\left(CID_j\right))$ to validate the secret-session key $S_{K1} = \hat{e}(H_t.Certify_j, H_t.H_1\left(CID_j\right))$. Using $S.H_1\left(CID_j\right)$, the cluster-head performs the computation, which is as follows:

$$\begin{aligned}
&\hat{e}\left(m_1, H_t.\text{r}.H_1\left(CID_j\right)\right)\\
&\quad = \hat{e}\left(H_t.H_1\left(PID_j \parallel H_2(x \oplus SK_j)\right), S.H_1\left(CID_j\right)\right.\\
&\quad = \hat{e}\left(S.H_1\left(PID_j \parallel H_2(x \oplus SK_j)\right), H_t.H_1\left(CID_j\right)\right.\\
&\quad = \hat{e}\left(Certify_j, H_t.H_1\left(CID_j\right)\right) = \Delta_j
\end{aligned}$$

On the other hand, the mobile-sink authenticates the cluster-head using $Ver_j = \nabla$ to render the transmission message $\{\nabla, TS_S\}$. As the certificate authorization is given only for the authorized mobile-sink, the other cannot infer / forge to generate a valid authentic key value Δ_j. To establish a secure communication, the mobile-sink and the cluster-head shares s session key $S_{K1} = \hat{e}(H_t.Certify_j, H_t.H_1\left(CID_j\right))$.Hence, the S-SAKA framework provides proper mutual authentication and session-key agreement.

2) DATA-CONFIDENTIALITY

In S-SAKA framework, to collect the sensing data, the mobile-sink should try to achieve the proper mutual authentication with cluster-head using shared session key. After the establishment of session key, the mobile-sink can acquire the sensing information through the knowledge of cluster-head. As the shared session key is kept secretly between the mobile-sink and cluster-head, the adversary cannot deduce the $Data_j$ in plaintext. Thus, the S-SAKA framework claims that it can provide a secure communication between the mobile-sink and the cluster-head.

On the one hand, the secret-session key $S_{K1} = \hat{e}(H_t.Certify_j, H_t.H_1\left(CID_j\right))$ is interfaced between the cluster-head CH and the mobile-sink. On the other hand, the cluster-head determines $CData_j = E_{M_K}\left(D_{S_{k1}}\left(CT_j\right)\right) = CData_j$ to save and send it to the base-station. Even if the adversary acquires the information of mobile-sink, he / she cannot infer $Data_j$ as it could not obtain the key value of M_K. As the adversary cannot tamper the sensing-data without knowledge of S_K, the S-SAKA framework provides data-confidentiality for users.

3) RESILIENT TO NODE-CAPTURE ATTACK

The resistance of node-capture attack can be measured effectively with the elimination of network communication, which are compromised by $'N'$ captured nodes directly [36]. Owing to inattentive property of WSNs', an adversary may capture the information of sensor-node or cluster-head. For your kind note, cluster-head has authentic identity CID_j and secret key value $S.H_1\left(CID_j\right)$ in initialization phase. Consequently, the adversary may have a chance to compromise the nodes, which are yet to communicate with mobile-sink and cluster-head. But then, the nodes, which are not compromised are still secure to establish the communication between mobile-sink and cluster-head. Subsequently, the S-SAKA framework claims that the adversary cannot provide any security disruption for uncompromised cluster-head and mobile-sink.

According to [42] and [43], the mobile-sink owner can deduce the recent updated cluster-head from LC_{DB_S} database, as soon as he / she has successfully logged into the base-station. Similarly, the mobile-sink can identify the compromised cluster-head timely to reject the compromised cluster-head. The un-compromised database table DB_S is associated with the mobile-sink securely; and thus the adversary cannot affect / damage the secure communication between the cluster-head and mobile-sink. Hence, the S-SAKA framework is resilient to node-capture attack.

4) RESILIENT TO STOLEN SMART-CARD ATTACK

Assume that adversary obtains the smart-card of the user MS_i; and thus he / she acquires the details of $Certify_j = S.H_1(PID_j \parallel H_2\left(x \oplus SK_j\right))$, $TS_j = H_2(PID_j \parallel y)$, $H_j = H_2(TS_j)$, $V_j = TS_j \oplus H_2(x \oplus SK_j))$, $A_j = H_2\left(PID_j \parallel x \parallel y\right)$. But then, the adversary can not deduce the users' unique identity PID_j and secret-key SK_j from $Certify_j$, V_j, H_j and A_j owing to one-way property of the hash function H_1 (.) and H_2 (.). Therefore, the adversary cannot compute a precise $m_1 = H_t.H_1(PID_j \parallel H_2(x \oplus SK_j)$ to form a valid request message $\left\{m_1, \Delta_j, TS_S\right\}$. Therefore, the S-SAKA claims that it is resilient to stolen smart-card attack.

5) RESILIENT TO REPLAY ATTACK

Using replay attack, the adversary uses a falsified authentication process to acquire the system access. In order to deduce such false assumption, the S-SAKA uses timestamp TS_S.

Assume an adversary wishes to launch a replay attack to infer the sensing data from cluster head CH. To extract the sensed data, the adversary needs to send an authentic message to CH. If the message $\left\{m_1^*, \Delta_j^*, TS_s^*\right\}$ is found to be expired or already used by another mobile-sink M_S, CH determines to be a susceptible behavior. Even though, the adversary changes the timestamp TS_s^*, he / she cannot find a proper Δ_j^* without the key parameter of B_{st} value S.

On the other hand, an adversary may wish to launch a replay attack to intercept with authentic mobile-sink M_S. To establish the communication, the adversary need to generate an authentic message $\{\nabla, TS_S\}$. As the key parameter of B_{st} value S is always kept secret between M_S and B_{st}, the adversary cannot determine a valid secret session key $S_{k2} = \hat{e}\left(m_1, H_t.\text{r}.H_1\left(CID_j\right)\right)$. Hence, the S-SAKA claims that the adversary cannot launch a replay attack without the proper computation of $\nabla = H_2\left(S_{k1} \parallel PID_j \parallel CID_j \parallel TS_S\right)$. This proves that the S-SAKA framework is resilient to replay attack.

6) RESILIENT TO KEY IMPERSONATION ATTACK

By using this attack, an adversary provides a forged information $\left\{m_1^*, \Delta_j^*, TS_s^*\right\}$ to impersonate as a legitimate mobile-sink M_S as to overhear the sensing information. However, the adversary cannot infer / forge Δ_j^* without the determination of $Certify_j$. According to DL problem, it is very much difficult to derive the secret key parameter S using P and p_{pub}. As a result, the adversary cannot determine $S.H_1(PID_j \parallel H_2\left(x \oplus SK_j\right)) = Certify_j$ and Δ_j^*. The above analysis proves that S-SAKA framework is resilient to key impersonation attack.

7) RESILIENT TO PRIVILEGED-INSIDER ATTACK

In the system registration phase of S-SAKA framework, the mobile-sink owner U_{ser} does not share his/her secret key SK_j in plaintext form. But, he / she shares its information as $H_2(x \oplus SK_j)$ to B_{st}. As $H_2(.)$ is a one-way point secure hashing function, it is computationally not possible to obtain SK_j. Moreover, the administrator or privileged B_{st} cannot determine a valid SK_j of U_{ser} and thus he / she cannot impersonate as a legal user U_{ser} to communicate with CH. Hence, the S-SAKA framework claims to be secure against the privileged-insider attack.

8) USER ANONYMITY AND INTRACTABILITY

In the system authentication phase, the S-SAKA framework uses mobile-sink M_S to send the transmission message $\left\{m_1, \Delta_j, TS_S\right\}$ to cluster-head CH in turn to obtain a proper user authentication. As each message transmission has unique time stamp TS_S that traverses between M_S and CH, there will be no correlation of two authentic messages, namely $\left\{m_1, \Delta_j, TS_S\right\}$ and $\left\{m_1^*, \Delta_j^*, TS_s^*\right\}$. Moreover, as the message transmission has one way point to map hashing function, it is much difficult to retrieve PID_j from m_1. Hence, the S-SAKA framework claims that the adversary cannot identify any authentic mobile-sink or communication link launched by the same mobile-sink.

9) RESILIENT TO OFFLINE PASSWORD-GUESSING ATTACK

This attack is categorized into two cases that are as follows:

Case 1: Assume an insider wishes to know the information of legitimate user, such as user identity PID_j and secret-key SK_j during system registration. The registration request of insider $\{PID_j, SK_j)\}$ is sent securely to the base-station. Besides, the insider has a smart-card, which are stolen from U_{ser}. Even though he has the device access and user information, he / she could not derive a proper secret session key without the knowledge of secret key value x.

Case 2: Assume an outsider has stolen the smart-card of U_{ser}. As a consequence, he / she can extract all the confidential information of smart-card, such as $\left\{PID_j, b_j, c_j, rN_j\right\}$, where

$$b_j = H_2\left(TS_j^* \parallel rN_j\right) \oplus H_2\left(x \oplus SK_j\right),$$
$$c_j = H_2\left(A_j \parallel H_2\left(x \oplus SK_j\right) \parallel rN_j\right).$$

To derive a secret key SK_j, the outsider needs to know secret key value x, which is a bilinear parameter corresponding to Z_p^*. As it is controlled and changed its value periodically by the base-station, the outsider cannot guess the proper secret key SK_j to gain the U_{ser} access.

The above analysis proves that the proposed S-SAKA framework can be resilient to offline password-guessing attack.

10) RESILIENT TO DoS ATTACK

Without proper user identity PID_j and secret key SK_j, none of the user can successfully log in to the systems. Even if they have stolen the smart card of legitimate user, they can infer the information like $\left\{PID_j, b_j, c_j, rN_j\right\}$, where $b_j = H_2\left(TS_j^* \parallel rN_j\right) \oplus H_2\left(x \oplus SK_j\right), c_j = H_2\left(A_j \parallel H_2\left(x \oplus SK_j\right) \parallel rN_j\right)$. After that, the smart-card verifies whether $H_j^* = H_2\left(TS_j^*\right)$ is valid or not, where $H_j = H_2(TS_j)$, $TS_j = H_2(PID_j \parallel y)$ and $TS_j^* = V_j \oplus H_2\left(PID_j \parallel H_2\left(x \oplus SK_j\right)\right)$. As the timestamp TS_j and secret value x periodically changes, they cannot derive a proper secret key SK_j to gain the user access. Hence, the proposed S-SAKA framework is resilient to denial of service attack.

11) RESILIENT TO MANY LOGGED-IN USERS WITH THE SAME LOGIN IDENTITY ATTACK

In the proposed S-SAKA framework, the user must provide valid credentials $\{PID_j, SK_j)\}$ to obtain the access of cluster-head through the knowledge of base-station, which verifies the secret value x to authorize the service access. As the secret value x is unique to U_{ser} and controlled by base-station, the user redundancy cannot be determined using following expressions: $Certify_j = S.H_1(PID_j \parallel H_2\left(x \oplus SK_j\right))$; $TS_j = H_2(PID_j \parallel y)$; $H_j = H_2(TS_j)$; $V_j = TS_j \oplus H_2(x \oplus SK_j))$;

TABLE 3. Comparison of communication efficiencies during system login and authentication phase.

Authentication Schemes	Mobile-Sink	Cluster-Head	Base-Station	Total Cost	Execution Time (ms)
Deebak [17]	$9T_{MH}$	$5T_{MH}$	$12T_{MH}$	$26T_{MH}$	0 .0104
Turkanovic et al. [18]	$5T_{MH}$	$7T_{MH}$	$7T_{MH}$	$19T_{MH}$	0 .0076
Farash et al. [19]	$11T_{MH}$	$7T_{MH}$	$14T_{MH}$	$32T_{MH}$	0 .0128
Das et al. [20]	$9T_{MH}$+ $1T_{ED}$	$3T_{MH}$+ $1T_{ED}$	$5T_{MH}$+$2T_{ED}$	$17T_{MH}$+$4T_{ED}$	1.2480
Amin [21]	$7T_{MH}$	$5T_{MH}$	$8T_{MH}$	$20T_{MH}$	0 .0080
Srinivas et al. [22]	$10T_{MH}$	$6T_{MH}$	$13T_{MH}$	$29T_{MH}$	0.0116
Proposed S-SAKA	$10T_{MH}$+$2T_P$	$5T_{MH}$+$2T_P$	$14T_{MH}$+$1T_P$+$3T_M$	$29T_{MH}$+$5T_P$+$3T_M$	0.0064 [Mobile-Sink and Cluster-Head Only]

$A_j = H_2\left(PID_j \parallel x \parallel y\right)$ as it is already in use. Hence, the proposed S-SAKA framework claim that it is resilient to many to many logged-in users with the same login identity attack.

VI. PERFORMANCE EVALUATION

In this section, the proposed S-SAKA framework is evaluated and compared with its related authentication schemes. The evaluation criteria of communication cost, some notation is defined as follows:

T_{SH} is defined as the execution time of one-way secure hashing function H_2 (.) .T_{MH} is defined as the execution time of one-way point to map hashing function H_1 (.).

T_P is defined as the computation time of bilinear pairing function.

T_A is defined as the execution time of one-point additional operational function.

T_{ED} is defined as the execution time of encryption and decryption algorithmic function.

T_M is defined as the execution time of elliptic-curve scalar multiplication function.

In WSNs, energy efficiency is a major constraint and thus lightweight user authentication protocols are preferred to mitigate the computational cost of the systems.

In order to reduce the amount of computations required, the proposed S-SAKA protocol uses cost inexpensive operations like hashing function and less cost expensive operation, such as bilinear pairing, encryption/decryption and scalar multiplication operation. To evaluate the cryptographic operations employed, an extensive verification is performed using MIRACLE C/C++ library with the system features of 32-bit Windows 7 Operating Systems and Microsoft Visual C++. To examine realistically, the execution time of symmetric key encryption/decryption ($AES - 128$), elliptic-curve point scalar multiplication over finite-field f_p and $SHA - 1$ hashing function are set as $T_P \approx 0.0001\ ms, T_{ED} \approx 0.1303\ ms$, $T_M \approx 7.3529\ ms$ and $T_{SH} \approx T_{MH} \approx 0.0004\ ms$ as referred in [22]. Table 3 demonstrates the communication efficiencies of the proposed S-SAKA and its related existing authentication schemes [17]–[22] during system login and authentication phase. Results show that, the computation cost of the bilinear pairing and scalar multiplication of proposed S-SAKA is comparatively short.

The examination results prove that the proposed S-SAKA has less communication overhead as it does not invoke the base-station to authenticate the mobile-sink and sensor node except during the secure communication establishment to provide seamless connectivity. Thus, it can be well suitable for WSN's environment in relation with the existing authentication schemes [17]–[22].

Table 3 compares the communication overhead involved in system login and authentication phases for proposed and other existing schemes [17]–[22]. While using SHA-1 hashing, the one-way hash function is assumed to be $160 - bits\,[20\ bytes]$. In addition, for each random nonce, the identity of sensor node is set to be $152 - bits\,[19\ bytes]$. In proposed S-SAKA, during system login phase, the login message transmission request $T_{Msg1} = \left\{\mathrm{PID_j, b_j, c_j, rN_j}\right\}$ and $T_{Msg2} = \left\{\mathrm{Q_j, CLC_{DB_S}}\right\}$ involves 78 *bytes* and 39 *bytes* respectively. In the course of system authentication and key agreement phase, the message transmission request $T_{Msg3} = \left\{m_1, \Delta_j, TS_S\right\}$ and $T_{Msg4} = \left\{PID_j, CData_j, b_j, c_j, rN_j\right\}$ encompasses 58 *bytes* and 98 *bytes*. As a result, during system login and authentication phase, the communication overhead is cumulated as follows: $[78 + 39 + 58 + 98] = 273 bytes$. On the other hand, the communication overheads involved in system login and authentication phases for Deebak [17], Turkanović *et al.* [18], Farash *et al.* [19], Das *et al.* [20], Amin and Biswas [21] and Srinivas *et al.* [22] are calculated as 315 *bytes*, 489 *bytes*, 434 *bytes*, 391 *bytes*, 373 *bytes*

TABLE 4. Assertion threshold test values.

Parameter	Value
g_{t1}	0.00
g_{t2}	0.33
g_{t3}	0.66

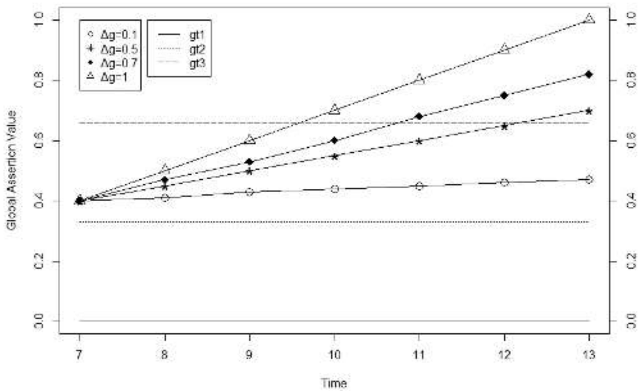

FIGURE 2. S-SAKA response with different Δg settings.

and 353 *bytes* respectively. It is observed that the proposed S-SAKA scheme provides less communication overhead in comparison with other existing authentication schemes [17]–[22].

Although the computation and execution times of mobile sink, cluster head and base stations are lower than the proposed protocol, total execution time in term of milliseconds the proposed S-SAKA performs lowest time. It is also remarked that S-SAKA can divide the protected resources in IoT-based environments into a number of assertion levels ($= n$). Assuming a total count of assertion levels n equal to 3, threshold values of the assertion levels can be as shown in Table 4.

$g_{ti} \in [0, 1]$ and represents the assertion level threshold value of the i^{th} assertion level that can be calculated by $g_{ti} = (i-1) * \frac{1}{n}$. g_{ti} reflects how confident the S-SAKA system must be about a user in order to assert his/her identity before granting access to them. It defines a control parameter representing the rate of change of the assertion value and referred to as Δg. This parameter can be used to control the speed by which the assertion value in safety-inspired applications' increases or decreases.

Changing the value of Δg affects how the system confidence is about a user access as shown in Fig. 2. In this figure when Δg is set to values between 0.1 and 0.5, the S-SAKA system confidence increases slowly and needs at least 6 events for the global assertion value to reach the next threshold value. This setting would be useful in IoT environments, where high security levels are a must such as the case in safety-inspired applications. However, when Δg has high values such 0.7 to 1.0, the system confidence rises much faster with less number of events to reach the second level; this setting would be useful for more relaxed IoT environments.

VII. CONCLUSION

In this paper, WSN security schemes are considered in terms of authentication and secure key agreement, which can be essential particularly for the IoT applications for public safety applications with cloud interactions. Enhancement of security framework can be essential for public safety paradigm since the IoT systems can be used for communication of sensitive information. Addressing the potential security based challenges, Seamless secure authentication and key agreement (S-SAKA) framework using bilinear-pairing and elliptic-curve cryptosystems has been proposed for the security issues, like data confidentiality, mutual authentication, session-key agreement, user anonymity, intractability and resilient to node-capture, key impersonation, password guessing and stolen smart-card attack. While using mobile-sink in WSNs, the S-SAKA framework does not only solve some major security issues, but also ensures a seamless connectivity to reduce the computation and communication cost of the network systems. In terms of authentication and authorization, recent studies on formal verifications that are based on bilinear pairing and elliptic-curve cryptosystems [44] are not provided for WSNs. As stated earlier, considering life-time of WSN, security aspects are critical and new solutions should be provided effectively and efficiently. The formal verification method and critical analyses performed prove that proposed S-SAKA provides mutual authentication, secure key agreement and data confidentiality. Furthermore, the results of performance evaluation show the reduced overhead of the proposed approach compared to the existing studies. Thus, the proposed S-SAKA framework can be well suited to the environments where public safety networks make use of IoT based applications with wireless sensors networks.

REFERENCES

[1] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.

[2] O. Vermesan and P. Friess, *Internet of Things-Global Technological and Societal Trends From Smart Environments and Spaces to Green ICT*. Aalborg, Denmark: River Publishers, 2011.

[3] D. Minoli, S. Kazem, and B. Occhiogrosso, "IoT considerations, requirements, and architectures for smart buildings—Energy optimization and next-generation building management systems," *IEEE Internet Things J.*, vol. 4, no. 1, pp. 269–283, Feb. 2017.

[4] I. Butun, M. Erol-Kantarci, B. Kantarci, and H. Song, "Cloud-centric multi-level authentication as a service for secure public safety device networks," *IEEE Commun. Mag.*, vol. 54, no. 4, pp. 47–53, Apr. 2016.

[5] M. Kamruzzaman, N. I. Sarkar, J. Gutierrez, and S. K. Ray, "A study of IoT-based post-disaster management," in *Proc. ICOIN*, Da Nang, Vietnam, Jan. 2017, pp. 406–410.

[6] Z. Chu *et al.*, "Game theory based secure wireless powered D2D communications with cooperative jamming," in *Proc. Wireless Days*, Porto, Portugal, Mar. 2017, pp. 95–98.

[7] G. Solmaz and D. Turgut, "Event coverage in theme parks using wireless sensor networks with mobile sinks," in *Proc. ICC*, Budapest, Hungary, Jun. 2013, pp. 1522–1526.

[8] R. Rahmatizadeh, S. Khan, A. P. Jayasumana, D. Turgut, and L. Bölöni, "Routing towards a mobile sink using virtual coordinates in a wireless sensor network," in *Proc. ICC*, Sydney, NSW, Australia, Jun. 2014, pp. 12–17.

[9] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Comput. Netw.*, vol. 52, no. 12, pp. 2292–2330, Aug. 2008.

[10] S. Kumari, M. K. Khan, and M. Atiquzzaman, "User authentication schemes for wireless sensor networks: A review," *Ad Hoc Netw*, vol. 27, pp. 159–194, Apr. 2015.
[11] D. He, N. Kumar, and N. Chilamkurti, "A secure temporal-credential-based mutual authentication and key agreement scheme with pseudo identity for wireless sensor networks," *Inf. Sci.*, vol. 321, pp. 263–277, Nov. 2015.
[12] D. He and D. Wang, "Robust biometrics-based authentication scheme for multiserver environment," *IEEE Syst. J.*, vol. 9, no. 3, pp. 816–823, Sep. 2015.
[13] D. Wang and P. Wang, "Understanding security failures of two-factor authentication schemes for real-time applications in hierarchical wireless sensor networks," *Ad Hoc Netw.*, vol. 20, pp. 1–15, Sep. 2014.
[14] D. Wang and P. Wang, "On the usability of two-factor authentication," in *Proc. Secure Commun.*, Beijing, China, 2014, pp. 141–150.
[15] D. Wang, D. He, P. Wang, and C.-H. Chu, "Anonymous two-factor authentication in distributed systems: Certain goals are beyond attainment," *IEEE Trans. Depend. Sec. Comput.*, vol. 12, no. 4, pp. 428–442, Jul./Aug. 2015.
[16] K. H. Wong, Y. Zheng, J. Cao, and S. Wang, "A dynamic user authentication scheme for wireless sensor networks," in *Proc. SUTC*, Taichung, Taiwan, Jun. 2006, pp. 244–251.
[17] B. D. Deebak, "Secure and efficient mutual adaptive user authentication scheme for heterogeneous wireless sensor networks using multimedia client–server systems," *Wireless Pers. Commun.*, vol. 87, no. 3, pp. 1013–1035, Apr. 2016.
[18] M. Turkanović, B. Brumen, and M. Hölbl, "A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion," *Ad Hoc Netw.*, vol. 20, pp. 96–112, Sep. 2014.
[19] M. S. Farash, M. Turkanović, S. Kumari, and M. Hölbl, "An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the Internet of Things environment," *Ad Hoc Netw.*, vol. 36, pp. 152–176, Jan. 2016.
[20] A. K. Das, A. K. Sutrala, S. Kumari, V. Odelu, M. Wazid, and X. Li, "An efficient multi-gateway-based three-factor user authentication and key agreement scheme in hierarchical wireless sensor networks," *Secur. Commun. Netw.*, vol. 9, no. 13, pp. 2070–2092, Sep. 2016.
[21] R. Amin and G. P. Biswas, "A secure light weight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks," *Ad Hoc Netw.*, vol. 36, pp. 58–80, Jan. 2016.
[22] J. Srinivas, S. Mukhopadhyay, and D. Mishra, "Secure and efficient user authentication scheme for multi-gateway wireless sensor networks," *Ad Hoc Netw.*, vol. 54, pp. 147–169, Jan. 2017.
[23] A. K. Das, P. Sharma, S. Chatterjee, and J. K. Sing, "A dynamic password-based user authentication scheme for hierarchical wireless sensor networks," *J Netw. Comput. Appl.*, vol. 35, no. 5, pp. 1646–1656, Sep. 2012.
[24] D. He, "An efficient remote user authentication and key agreement protocol for mobile client–server environment from pairings," *Ad Hoc Netw.*, vol. 10, no. 6, pp. 1009–1016, Aug. 2012.
[25] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," *ACM Trans. Comput. Syst.*, vol. 8, no. 1, pp. 18–36, Feb. 1990.
[26] D. Wang, N. Wang, P. Wang, and S. Qing, "Preserving privacy for free: Efficient and provably secure two-factor authentication scheme with user anonymity," *Inf. Sci.*, vol. 321, pp. 162–178, Nov. 2015.
[27] M. L. Das, "Two-factor user authentication in wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 8, no. 3, pp. 1086–1090, Mar. 2009.
[28] J. Yuan, C. Jiang, and Z. Jiang, "A biometric-based user authentication for wireless sensor networks," *Wuhan Univ. J. Natural Sci.*, vol. 15, no. 3, pp. 272–276, Jun. 2010.
[29] H. R. Tseng, R. H. Jan, and W. Yangand, "An improved dynamic user authentication scheme for wireless sensor networks," in *Proc. GLOBECOM*, Washington, DC, USA, Nov. 2007, pp. 986–990.
[30] T.-H. Lee, "Simple dynamic user authentication protocols for wireless sensor networks," in *Proc. SENSORCOMM*, Cap Esterel, France, Aug. 2008, pp. 657–660.
[31] E. G. AbdAllah, H. S. Hassanein, and M. Zulkernine, "A survey of security attacks in information-centric networking," *IEEE Commun. Surveys Tuts.*, vol. 7, no. 3, pp. 1441–1454, 3rd Quart., 2015.
[32] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proc. CCS*, Washington, DC, USA, 2002, pp. 41–47.
[33] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *Proc. SSP*, Oakland, CA, USA, May 2003, pp. 197–213.
[34] A. Rasheed and R. N. Mahapatra, "The three-tier security scheme in wireless sensor networks with mobile sinks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 5, pp. 958–965, May 2012.
[35] R. Watro, D. Kong, S.-F. Cuti, C. Gardiner, C. Lynn, and P. Kruus, "TinyPK: Securing sensor networks with public key technology," in *Proc. SASN*, Washington, DC, USA, 2004, pp. 59–64.
[36] T.-H. Chen and W.-K. Shih, "A robust mutual authentication protocol for wireless sensor networks," *ETRI J.*, vol. 32, no. 5, pp. 704–712, Oct. 2010.
[37] D. He, Y. Gao, S. Chan, C. Chen, and J. Bu, "An enhanced two-factor user authentication scheme in wireless sensor networks," *Ad-Hoc Sensor Wireless Netw.*, vol. 10, no. 4, pp. 361–371, Jan. 2010.
[38] S. Park, B. Aslam, D. Turgut, and C. Zou, "Defense against Sybil attack in the initial deployment stage of vehicular ad hoc network based on roadside unit support," *Secur. Commun. Netw.*, vol. 6, no. 4, pp. 523–538, Apr. 2013.
[39] O. Delgado-Mohatar, A. Fúster-Sabater, and J. M. Sierra, "A light-weight authentication scheme for wireless sensor networks," *Ad Hoc Netw.*, vol. 9, no. 5, pp. 727–735, Jul. 2011.
[40] D. K. Altop, M. A. Bingöl, A. Levi, and E. Sava , "DKEM: Secure and efficient distributed key establishment protocol for wireless mesh networks," *Ad Hoc Netw.*, vol. 54, pp. 53–68, Jan. 2017.
[41] W. Shi and P. Gong, "A new user authentication protocol for wireless sensor networks using elliptic curves cryptography," *Int. J. Distrib. Sensor Netw.*, vol. 9, no. 4, p. 730831, 2013.
[42] T. Bonaci, L. Bushnell, and R. Poovendran, "Node capture attacks in wireless sensor networks: A system theoretic approach," in *Proc. CDC*, Atlanta, GA, USA, Dec. 2010, pp. 6765–6772.
[43] T. M. Vu, R. Safavi-Naini, and C. Williamson, "Securing wireless sensor networks against large-scale node capture attacks," in *Proc. ASIACCS*, Beijing, China, 2010, pp. 112–123.
[44] D. Hankerson, A. J. Menezes, and S. Vanstone, *Guide to Elliptic Curve Cryptography*. New York, NY, USA: Springer, 2004.
[45] L. C. Washington, *Elliptic Curves: Number Theory and Cryptography*, 2nd ed. Boca Raton, FL, USA: Chapman & Hall, 2008.
[46] N. Kumar, S. Misra, N. Chilamkurti, J.-H. Lee, and J. J. P. C. Rodrigues, "Bayesian coalition negotiation game as a utility for secure energy management in a vehicles-to-grid environment," *IEEE Trans. Depend. Sec. Comput.*, vol. 13, no. 1, pp. 133–145, Jan./Feb. 2016.
[47] F. Al-Turjman, M. Imran, and A. V. Vasilakos, "Value-based caching in information-centric wireless body area networks," *Sensors*, vol. 17, no. 1, p. 181, Jan. 2017.
[48] C. Stergiou, K. E. Psannis, B. G. Kim, and B. Gupta, "Secure integration of IoT and cloud computing," *Future Generat. Comput. Syst.*, vol. 78, pp. 964–975, Jan. 2018. [Online]. Available: https://doi.org/10.1016/j.future.2016.11.031.
[49] M. Tao, J. Zuo, Z. Liu, A. Castiglione, and F. Palmieri, "Multi-layer cloud architectural model and ontology-based security service framework for IoT-based smart homes," *Future Generat. Comput. Syst.*, vol. 78, pp. 1040–1051, Jan. 2018. [Online]. Available: https://doi.org/10.1016/j.future.2016.11.011.
[50] B. Kantarci and H. T. Mouftah, "Trustworthy sensing for public safety in cloud-centric Internet of Things," *IEEE Internet Things J.*, vol. 1, no. 4, pp. 360–368, Aug. 2014.

FADI AL-TURJMAN (M'07) received the Ph.D. degree in computing science from Queen's University, Canada, in 2011. He is currently a Visiting Associate Professor with the Computer Engineering Department, Middle East Technical University Northern Cyprus Campus. He is also a leading authority in the areas of smart/cognitive, wireless and mobile networks' architectures, protocols, deployments, and performance evaluation. Recently, he published the book *Cognitive Sensors & IoT: Architecture, Deployment, and Data Delivery* (Taylor and Francis, CRC New York) (a top-tier publisher in the area). His record spans more than 140 publications in journals, conferences, patents, books, and book chapters, in addition to numerous keynotes and plenary talks at flagship venues, including the IEEE ICC, LCN, GLOBECOM, and IWCMC conferences. He has received several recognitions and best papers awards at top international conferences, and led a number of international symposia and workshops in flagship ComSoc conferences.

YONEY KIRSAL EVER (M'98) received the B.Sc. degree from the Department of Computer Engineering, Eastern Mediterranean University, Cyprus, in 2002, the M.Sc. degree in Internet computing from the University of Surrey, Guilford, U.K., in 2003, and the Ph.D. degree from the School of Engineering and Information Sciences, Middlesex University, London, U.K. In 2012, she completed the Post Graduate Certificate in Higher Education. She is involved in the development of security strategies using Kerberos in wireless networks. She was a part-time Lecturer while earning the B.Sc. and Ph.D. degrees, and a Lecturer with the Computer and Communications Engineering Department, Middlesex University. She is currently an Assistant Professor at Near East University, Cyprus. She has published international conference papers earning various awards, including the IEEE Best Paper for promising research. Her research interests are in network security, authentication protocols, and formal verification methods. She has been a member of ACM since 2007. She reviews papers for various journals mainly on network security.

ENVER EVER (M'15) received the B.Sc. degree from the Department of Computer Engineering, Eastern Mediterranean University, Cyprus, in 2002, and the M.Sc. degree in computer networks and the Ph.D. degree in performance evaluation of computer networks and communication systems from Middlesex University in 2004 and 2008, respectively. He was with Bradford University as a Post-Doctoral Research Associate for a year. He was a Senior Lecturer with the Computer and Communications Engineering Department, Middlesex University. He is currently an Associate Professor with Middle East Technical University Northern Cyprus Campus. His current research interests include computer networks, wireless communication systems, parallel computing paradigms, wireless sensor networks, integrated circuits, and performance/reliability modeling. He serves on various programme committees and received the Exemplary Reviewer Award for his contributions as a reviewer.

HUAN X. NGUYEN (M'06–SM'15) received the B.Sc. degree with the Hanoi University of Science and Technology, Vietnam, in 2000, and the Ph.D. degree from the University of New South Wales, Australia, in 2007. He was a Research Officer with Swansea University, U.K., from 2007 to 2008, and a Lecturer with Glasgow Caledonian University, U.K., from 2008 to 2010. He is currently an Associate Professor of communication networks with the Faculty of Science and Technology, Middlesex University, London, U.K. He has published over 90 research papers, mainly in the IEEE journals and conferences. His research interests include PHY security, energy harvesting, MIMO techniques, communications for critical applications, network coding, relay communication, cognitive radio, and multi-carrier systems. He received a grant from the Newton Fund/British Council Institutional Links Program (2016–2018) for Disaster Communication and Management Systems using 5G Networks. He was the Co-Chair of the 2017 International Workshop on 5G Networks for Public Safety and Disaster Management. He is currently serving as an Editor of the *KSII Transactions on Internet and Information Systems*.

DEEBAK BAKKIAM DAVID received the B.Tech. degree in information technology from Anna University, Chennai, the M.E. degree in embedded system and computing from RTM Nagpur University, and the Ph.D. degree in multimedia communication and security from SASTRA University, Thanjavur, in 2007, 2009, and 2016, respectively. He is currently with Middle East Technical University Northern Cyprus Campus as a Post-Doctoral Researcher. His research interests include multimedia systems, network security, authentication and key agreement, network routing, computer networks, embedded systems, mobile cloud computing social networks, and cyber physical systems.