

Article

An Authentication Protocol for Future Sensor Networks

Muhammad Bilal * and Shin-Gak Kang

Electronics and Telecommunications Research Institute, University of Science and Technology, 218, Gajeong-ro, yuseong-gu, Daejeon 34129, Korea; sgkang@etri.re.kr

* Correspondence: mbilal@etri.re.kr; Tel.: +82-42-860-6424

Academic Editors: Dongkyun Kim, Houbing Song, Juan-Carlos Cano, Wei Wang, Waleed Ejaz and Qinghe Du
Received: 17 March 2017; Accepted: 24 April 2017; Published: 28 April 2017

Abstract: Authentication is one of the essential security services in Wireless Sensor Networks (WSNs) for ensuring secure data sessions. Sensor node authentication ensures the confidentiality and validity of data collected by the sensor node, whereas user authentication guarantees that only legitimate users can access the sensor data. In a mobile WSN, sensor and user nodes move across the network and exchange data with multiple nodes, thus experiencing the authentication process multiple times. The integration of WSNs with Internet of Things (IoT) brings forth a new kind of WSN architecture along with stricter security requirements; for instance, a sensor node or a user node may need to establish multiple concurrent secure data sessions. With concurrent data sessions, the frequency of the re-authentication process increases in proportion to the number of concurrent connections. Moreover, to establish multiple data sessions, it is essential that a protocol participant have the capability of running multiple instances of the protocol run, which makes the security issue even more challenging. The currently available authentication protocols were designed for the autonomous WSN and do not account for the above requirements. Hence, ensuring a lightweight and efficient authentication protocol has become more crucial. In this paper, we present a novel, lightweight and efficient key exchange and authentication protocol suite called the Secure Mobile Sensor Network (SMSN) Authentication Protocol. In the SMSN a mobile node goes through an initial authentication procedure and receives a re-authentication ticket from the base station. Later a mobile node can use this re-authentication ticket when establishing multiple data exchange sessions and/or when moving across the network. This scheme reduces the communication and computational complexity of the authentication process. We proved the strength of our protocol with rigorous security analysis (including formal analysis using the BAN-logic) and simulated the SMSN and previously proposed schemes in an automated protocol verifier tool. Finally, we compared the computational complexity and communication cost against well-known authentication protocols.

Keywords: authentication; sensor networks; network security; key distribution; privacy; BAN logic

1. Introduction

Wireless sensor networks (WSNs) consist of a vast number of distributed sensor nodes. Each sensor node is an autonomous system that monitors and collects data from the surrounding environment. In wireless sensor networks, the sensor nodes have limited computational power and communication capabilities; hence, most of the conventional cryptographic mechanisms and security protocols are not suitable for resource limited WSNs. For instance, very efficient public key algorithms, such as ECC [1], need a fraction of a second to execute the encryption/decryption procedures, while a symmetric key algorithm, such as RC5 [2], needs only a fraction of a millisecond to perform encryption and decryption procedures [3–5]. In a sensor network where devices have limited resources, the asymmetric cryptographic functions must be used wisely; for instance, use

the asymmetric cryptography when an authentication responder already verified the initiator and wants to share secret information. If the responder does not authenticate the initiator and the initiator uses the asymmetric cryptography to exchange a message, the network becomes vulnerable to DOS attacks [5,6]. The WSN related constraints mentioned above are known to the research community. The currently available authentication protocols are designed for the autonomous WSN from the perspective of the above constraints. Moreover, in the near future in the realization of the vision of emerging technologies such as IoT, D2D, smart home and smart cities, WSNs will provide an invaluable service by acting as a virtual layer between the physical world and the computational devices [7–9]. However, integration of WSNs with IoT will bring forth a new kind of WSN architecture and stricter security requirements; for instance, in a smart hospital (as shown in Figure 1b) a sensor node or a user node may require the establishment of multiple concurrent secure data sessions. To establish a secure data session, authentication is the first step. In a dynamic, mobile WSN environment, where sensors and user nodes can establish multiple concurrent connections, a node moving across the network undergoes the authentication check multiple times and the frequency of the re-authentication process increases in proportion to the number of concurrent connections. Moreover, to establish multiple data sessions, it is essential that a protocol participant has the capability of running multiple instances of the protocol run, which makes the security issue even more challenging. Thus, it is essential to adopt a secure yet lightweight authentication procedure that especially reduces the computational time and communication at the mobile sensor node. The currently available authentication protocols were designed for the autonomous WSN and do not account for these new emerging challenges. This work presents a novel authentication protocol suite called the Secure Mobile Sensor Network (SMSN) Authentication Protocol. The SMSN protocol suite consists of six protocols: three protocols deal with mobile sensor node authentication with sink nodes and the other three deal with user node activation and authentication with the base station, sink nodes, and sensor nodes. In the SMSN, mobile sensors and user nodes can join and leave the system dynamically and can establish secure multiple concurrent connections. After the initial authentication, a mobile sensor or a user node can move across the network and get re-authenticated by a simple ticket-based re-authentication protocol; for instance, a user node can establish concurrent connections with multiple sink and sensor nodes using a re-authentication ticket issued during the initial-authentication protocol run. To establish multiple connections, a node is allowed to run multiple instances of the protocol; consequently, we introduce extra design requirements to meet the goals of a secure authentication protocol. In this paper, we also present an efficient and lightweight key generation and distribution mechanism. In the key generation protocol, a commitment key is generated by a group of participants (the base station and sink nodes) using an irreversible function; the key agreement and key retrieval protocol are the same as that employed in [10]. The commitment key is further applied to drive multiple time-based encryption keys, for example, the ticket encryption key and session key between the sink and user/sensor are derived from the commitment key. The time dimension in the protocol increases the security of the protocol; although the group members do not need to be tightly or loosely time synchronized. To determine the security of the protocol in this study, we performed a rigorous security analysis and also simulated the SMSN and previously proposed schemes [11–16] in an automated security protocol analysis tool called Scyther [17,18], which is a powerful state-of-art tool that finds attacks for defined protocol properties. We observed that our authentication protocol is secure, and it achieves all the objectives of an authentication protocol, which are defined as protocol claims in Section 5.3; for a detailed description of protocol claims, please refer to [19,20]. We also compared the efficiency of the SMSN in terms of computational time and communication complexities as discussed in [11–16]. The remainder of this paper is organized as follows. Section 2 presents the related work. In Section 3, we present a brief system overview and problem statement. Section 4 describes the proposed scheme with a detailed discussion. In Section 5, we assess the strength of our scheme against the known attacks. Section 6 presents an efficiency analysis that compares a few interesting schemes with our scheme. Finally, we provide concluding remarks in Section 7.

2. Related Work

Typically, WSNs are comprised of distributed devices with limited resources. Most of the conventional cryptographic mechanisms and security protocols are computationally expensive and are not suitable for resource-limited WSNs. In the recent past, the research community proposed several authentication protocols [11–16,21–27] that provide security in a WSN environment. Since the sensor nodes have low computational time, storage and communication capabilities, it is essential to design an efficient and lightweight yet secure authentication mechanism. From the point of view of computational and communication complexity, the authentication procedure in a wireless network with a mobile sensor and user node is an expensive task. A node moving across the network undergoes multiple authentication checks. Thus, it is essential to adopt a secure yet lightweight authentication procedure that especially reduces the computational and communication resources at the mobile sensor node. In [28] the authors discussed various anomaly detection techniques for flat and hierarchical wireless sensor networks, but detection techniques are not sufficient for several security threats. However, for a secure system together with detection methods, prevention techniques such as authentication is also vital; various authentication protocols for WSNs were proposed in [11–16,21–27]. In 2006, Wong [21] proposed a user authentication scheme for a dynamic WSN. The scheme is password based and employs lightweight cryptographic hash and XOR operations. Later on, Tesng et al. [11] and Das [22] identified that the Wong [21] scheme had various weaknesses and was vulnerable to replay attacks and forgery attacks, and the user password is known to the sensor node and can be revealed by any sensor node. Tesng et al. [11] proposed an improved version to mitigate the weaknesses that posed security threats in the Wong [21] scheme. The scheme of Tesng et al. [11] is also a password-based scheme, but the password is not revealed to the sensor nodes, and they also introduced a new phase of password change. Nevertheless, the scheme of Tesng et al. [11] is weak against replay attacks, impersonation attacks and forgery attacks [29]. Moreover, the scheme does not provide a mutual authentication between the gateway (GW) and sensor node (SN). In 2009, Das [22] proposed a two-factor user authentication scheme in which legitimate users can register and log in to the remotely deployed sensor nodes to access the collected data. From the point of view of computational and communication complexity, the scheme is reasonably efficient. The author claimed the scheme was secure against various kinds of attacks. However, later work [30–32] suggested that the scheme was vulnerable to different types of attacks, including impersonation, password guessing, insider, and parallel session attacks. Moreover, it did not provide the mutual authentication between the GW and sensor nodes. In 2010, Yoo et al. [12] proposed a user authentication scheme for WSNs and analyzed the protocol using BAN logic [33]. However, the BAN logic provided a foundation for the formal analysis of security protocols, but in the case of authentication, various attacks could slip through the BAN logic [34,35]. The scheme of Yoo et al. provided mutual authentication between the GW and the user and established a session key between the GW and SN. The authors claimed that the scheme was safe against insider attacks, impersonation attacks, and parallel session attacks. However, in [27] the authors provided a detailed analysis of the scheme of Yoo et al. and proved that the scheme is susceptible to various attacks, including insider attacks, impersonation attacks, parallel session attacks, password guessing attacks, fake registration attacks, and DOS attacks. Kumar et al. [36] proposed an authentication protocol for WSNs and claimed it could satisfy all the security requirements of the WSN; however, He et al. [37] proved that the scheme was weak against insider attacks and offline password guessing attacks and could not provide user anonymity. Kumar proposed another enhanced scheme in [13] and once again claimed that the scheme could withstand most of the known attacks and provide user privacy. However, with the Scyther implementation, given in Section 5.3, we found that the improved scheme presented in [13] was still vulnerable to insider attacks, parallel session attacks, and impersonation attacks. Farash et al. [15] proposed a key agreement and authentication protocol for WSNs in the Internet of Things (IoT) environment. The scheme was well designed and provided security against several well-known attacks. The author proved the strength of the protocol with BAN logic and further confirmed the theoretical analysis results by implementing the protocol in

the AVISPA [38] tool. However, similar to other schemes discussed above, Farash et al. [15] did not consider the requirement of concurrent sessions, which are more likely to occur in an IoT environment. Moreover, our implementation of the scheme of Farash et al. [15] in Scyther revealed that the GW was vulnerable against insider attacks and impersonation attacks. The scheme was insecure in the presence of an intruder as discussed in Section 5.3, which assumed that the initial knowledge set of intruders included the identities of all sensor and user nodes. With known identities, an intruder can impersonate the user node and deceive the GW to falsify the authentication properties. Similarly, the Scyther implementation of the recently published work of Y. Lu et al. [16] and Quan et al. [14] revealed that at least two protocol participants falsified the authentication properties. The majority of the schemes discussed above provided the GW and user authentication schemes; however, Farash et al. [15] and Y. Lu et al. [16] also considered the GW and sensor node authentication, while Farash et al. [15] also allowed the user to access the data from sensor nodes directly. Li et al. [39] proposed an authentication protocol for sensor and user nodes. However, the proposed scheme requires time synchronization among protocol participants, and also employs the asymmetric elliptic curve cryptography, which is not a good design choice for resource limited WSN applications. For instance, very efficient public key algorithms, such as ECC [1], need a fraction of a second to execute the encryption/decryption procedures, while a symmetric key algorithm, such as RC5 [2], needs only a fraction of a millisecond to perform encryption and decryption procedures [3–5]. Unlike all the schemes discussed above, our proposed scheme, the SMSN authentication protocol suite, allows the sensor and user nodes to establish multiple concurrent connections with different sensor and sink nodes, which makes our scheme suitable for deploying it in the future to support IoT and related emerging technologies. Moreover, the SMSN authentication protocol suite provides several kinds of mutual authentications; for instance, after the initial authentication, a sensor or the user node receives an authentication ticket issued by the base station. The ticket can be further used for sensor-sink, user-sink, and user-sensor mutual authentication. The SMSN protocol suite consists of six protocols: three protocols deal with mobile sensor node authentication with sink nodes, and the other three deal with user node activation and authentication with the base station, sink nodes, and sensor nodes.

3. System Overview and Problem Statement

A typical WSN consists of the base station (BS), sink node (S), sensor node (N) and user node (U_i). We assume that the BS knows the public keys of the sink (S), sensor (N) and user (U) nodes.

3.1. System Architecture

An IoT smart service provider deploys the WSN with various base stations connected to the internet through a service center (SC). In the WSN, each BS_j forms a group G_j consisting of neighbor base stations and associated sink nodes. All group members share a symmetric group key K_G^j , which is controlled by the group master BS_j using a group key agreement protocol such as discussed in [40]. Furthermore, the base station can access and download the profile of the mobile Sensor (N) and User (U) nodes. Each profile has a unique secret number n_s^i ; besides being known to the base station, this secret number is also known to the corresponding Sensor (N) and User (U) nodes. The profile and unique secret number n_s^i of all legitimate users and sensor nodes are accessible to the base station through the service center.

An example of the overall system architecture is depicted in Figure 1a. BS_1 creates a group consisting of neighbor BS_0 and BS_2 and associated sinks (S_1, S_3). BS_1 generates a group key K_G^1 and shares it among all group members. The profiles and the associated unique secret number of U_1, N_1 , and N_2 are accessible to BS_0, BS_1 , and BS_2 via the service center. An application scenario is given in Figure 1b. In a smart hospital, the health status of a hospitalized patient is continuously monitored by the several sensors mounted over the patient's body. For the secrecy of patients' health data it is essential that only authorized users be able to access the data. When a patient takes a walk in the hospital, all the mounted sensors continuously transfer the different sets of data to the nearby sinks.

With the SMSN, the sensors do not need to follow the full authentication procedure for each data session; instead, the session continues by sharing a simple re-authentication ticket that provides the sensor-sink mutual authentication. Similarly, when an authorized doctor or nursing staff visits patients, he or she can access the sensor data in real time via a user device by establishing multiple data sessions with various sinks and sensor nodes. While he moves from patient to patient, his device does not need to follow a complete authentication procedure for each data session; rather, a simple re-authentication ticket can be used for re-authentication. From outside of the hospital, an authorized person can log into the hospital system and access the necessary data from sink and base stations by using the same re-authentication ticket.

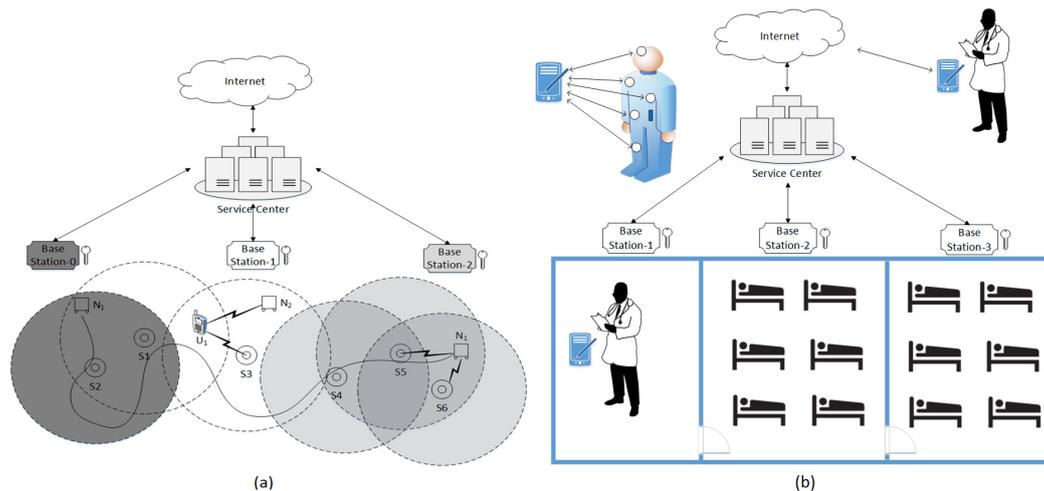


Figure 1. (a) An example scenario of a Wireless Sensor Network (WSN). Sensor node N_1 is initially authenticated by BS_1 ; while moving across the network it is re-authenticated by the re-authentication ticket. At the final destination N_1 shares data with multiple sink nodes while user node U_1 collects data from sensor node N_2 and sink node S_3 . (b) An application scenario of Secure Mobile Sensor Network (SMSN) in a smart hospital. Authorised doctors and nursing staff can access the data from sensor nodes using a user device.

3.2. Problem Statement

As described earlier, in IoT and other related emerging technologies, a mobile sensor or user node may need to exchange data with multiple nodes and so will experience the authentication process multiple times. With multiple concurrent connections, the authentication process becomes even more expensive while a node moves across the network. As shown in Figure 1a, sensor node N_1 is communicating with sink nodes S_5 and S_6 simultaneously. Likewise, user node U_1 is communicating with sink node S_3 and sensor node N_2 simultaneously. In such an application scenario, when a sensor or the user node moves across the network the frequency of the re-authentication process increases in proportion to the number of concurrent connections. Moreover, to establish multiple data sessions, it is essential that a protocol participant run multiple instances of the protocol run, which makes the security issue even more challenging. To perform multiple parallel re-authentications, it is evident that the protocol participants run multiple instances of the protocol. With the multiple protocol runs, the assurance of security of the protocol becomes more challenging. Therefore, for seamless services, lightweight yet secure re-authentication is vital.

According to our knowledge, we are the first to propose an authentication protocol for concurrent secure connections. To perform multiple parallel re-authentications, the protocol participants must run multiple instances of the protocol. With multiple protocol runs, the assurance of security of the protocol becomes more challenging. We developed our scheme, the SMSN, considering the following constraints: (1) the communication channels are insecure; (2) an intruder with the capabilities as

described in Section-5-C is present in the network to launch various attacks; (3) due to the requirements for a WSN deployed in an IoT environment, the protocol participants are allowed to run multiple instances of the protocol; and (4) user and sensor nodes can dynamically leave and join the network and can move across the network.

3.3. Notations

- BS_j = The j th base station
- S_j = The j th sink
- N_i = The i th sensor node
- U_i = The i th user node
- $A \sim B$ = A is associated with B such that B is controlling authority
- G_j = Group of all associated entities of j th BS
- G_j^o = Group of all non-associated entities of j th BS who knows the K_G^j
- K_i^j = The time-based key generated at i th interval by BS_j
- K_G^j = A group key generated by BS_j
- $K_S^i = N \rightarrow S$ session key generated at i th interval
- $K_{PS}^i = N \rightarrow S$ private session key
- K_{TS}^i = Temporary session key
- $E_B^i(m)$ = Encryption of ' m ', using key K_B^i
- V_i = The index value for interval i
- T_k = The k th Ticket
- $Z(A)$ = An intruder Z mimicking the entity A
- n_i = i th nonce in a message exchange

4. Proposed Scheme

The SMSN protocol suite consists of two protocol suites, the Keying Protocol suite and an Authentication Protocol suite. The Keying Protocol suit further comprises a key agreement protocol, a key retrieval protocol (which is the same as the one employed in [10]), and a key management protocol; likewise, the authentication protocol further comprises six protocols, three dealing with mobile sensor node authentication with sink nodes and the other three dealing with user node activation and authentication with a base station, sink nodes, and sensor nodes in different scenarios. In subsequent sections, the SMSN protocol suite is described in detail.

4.1. Keying Protocol Suite

In key generation protocol a 'commitment key' is generated by group participants (the base station and sink nodes) using an irreversible function similar to that as used in [10,41]. The 'commitment key' is further used to drive multiple time-based keys; for instance, the ticket encryption key and session key between the sink and user/sensor are derived from the 'commitment key'.

4.1.1. Key Agreement Protocol

The key generation and distribution mechanism is shown in Figure 2 and consists of the following steps: (1) After every time interval T_d , BS_j broadcasts the key generation information (Key_{MSG}) to all members of G_j . (2) Using the key generation information (KEY_{MSG}), all members generate a Commitment Key Generator $\zeta_0^j = H(T_d, N_0^j)$. (3) All members of G_j can now generate a "Chain of Key Generators" of length L by using an irreversible one-way function: $H(\zeta_0^j) = \zeta_1^j, H(\zeta_1^j) = \zeta_2^j \dots H(\zeta_{l-1}^j) = \zeta_l^j$; i.e., $H(\zeta_k^j)^i = \zeta_{k+i}^j$. (4) Each generator (ζ) in the chain is used by function g at specific intervals to derive indexes and a ticket encryption key pair. For instance, at interval k for any sensor or user node i , which is requesting the BS_j to join the network, the function $g(\zeta_k^j) = H(\zeta_k^j, H(n_0^i)) || H(k)$ generates ticket encryption key K_k^j and index V_k value. In function g the value n_0^i is a secret nonce sent by node i in a network join message. (5) Furthermore, BS_j issues a

session key based upon the ticket encryption key K_k^j and secret nonce n_0^i , i.e., $K_s^i = H(K_k^j, n_0^i)$ where n_0^i is the secret random number assigned to legitimate sensor and user node.

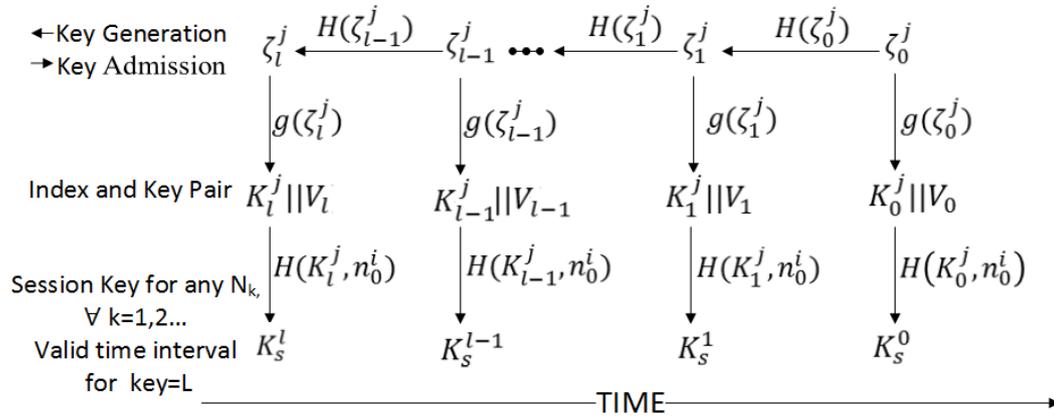


Figure 2. Time-based keys generation and admission with reference to time passage.

All the symmetric keys generated in the above discussion have a size of 256 bits (32 bytes); hence, in the subsequent section of authentication protocols any symmetric encryption supporting the 256-bit key can be used, e.g., RC5/6 [2]; Rijndael [42], Twofish [43], MARS [44], and Blowfish [45].

4.1.2. The Key Retrieval Protocol

After initial authentication, BS_j issues a ticket to the requesting node. The ticket consists of two parts: (1) The first half consists of the sensor node identity (id) N_i , the session key K_s^i , secret nonce n_0^i , and the profile. This part of the ticket is encrypted with time-based key K_i^j . (2) The second half consists of sensor node id N_i , the hash of group id $H(G^j)$, and the required information to retrieve the time-based key K_i^j . The hash of group id $H(G^j)$ is an optional field used only if sink nodes can join multiple base stations; in that case, it is used to identify the group and to select the correct keychain. In the key retrieval, information depends on the selected mode and can be the scrambled index value V , index vector V_{Hash} , or index value i ; the modes of a ticket are explained below.

Mode-01:

In mode-01, the ticket retrieval information comprises the index value (V_i), requesting node id N_k , and the hash of the user’s private key. The ticket verifier searches the appended index value (V_i) within its generated vector (V). A search hit at the i th place means that key K_i^j can decrypt the ticket.

$$T_k = E_i^j(N_i || K_s^i || n_0^i || Prof || E_G^j(N_i || V_i || H(G^j))) \quad (\text{Ticket formate for mode-01})$$

Mode-02:

If the group members are in a fully secure environment, the SMSN employs a simple ticket retrieval strategy. Instead of using a scrambled index value V in mode-02 in the ticket retrieval, the information contains the interval value. In this mode, the ticket verifier does not need to run a search algorithm.

$$T_k = E_i^j(N_i || K_s^i || n_0^i || Prof || E_G^j(N_i || i || H(G^j))) \quad (\text{Ticket formate for mode-02})$$

Mode-03:

In mode-03, the ticket retrieval information is the same as in mode-01 except it includes a hash vector of size $\log_2 |V| - 2$. These hash values are carefully chosen nodes of the binary hash tree, which is generated such that the leaf nodes are indexed vector (V) values as shown in Figure 3.

$$T_k = E_i^j(N_i || K_s^i || n_0^i || Prof || E_G^j(N_i || V_{Hash} || H(G^j))) \quad (\text{Ticket format for mode-03})$$

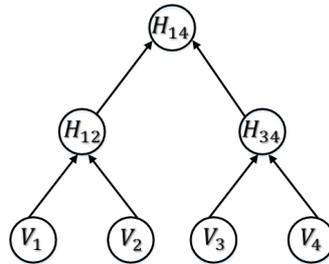


Figure 3. Example of Binary Hash Tree generated with index vector $V = v_1, v_2, v_3, v_4$.

Search Algorithm:

- Start from the root and move down
- Ignore appended values and follow the path of the reconstructed node
- Continue until level $\log_2 |V| - 1$ is reached
- At level $\log_2 |V|$, select the appended value that is the index value.

Mode-03 is suitable if the Chain of the Key Generator is very long. The tree “root node” is included in T_k (optionally), which ensures that a trusted group member generated T_k .

4.1.3. Key Management Protocol for the Sink Node

At the start of the Chain of the Key Generator, the sink node reissues the ticket and session keys to associated nodes. To spread out the workload in the time dimension the sink node keeps the history of the previous key chain and issues a ticket/session key based on the moving window algorithm. Let us consider a chain with a length of 3; Figure 4 shows how the sink node spread the workload throughout the chain by adopting the moving window approach.

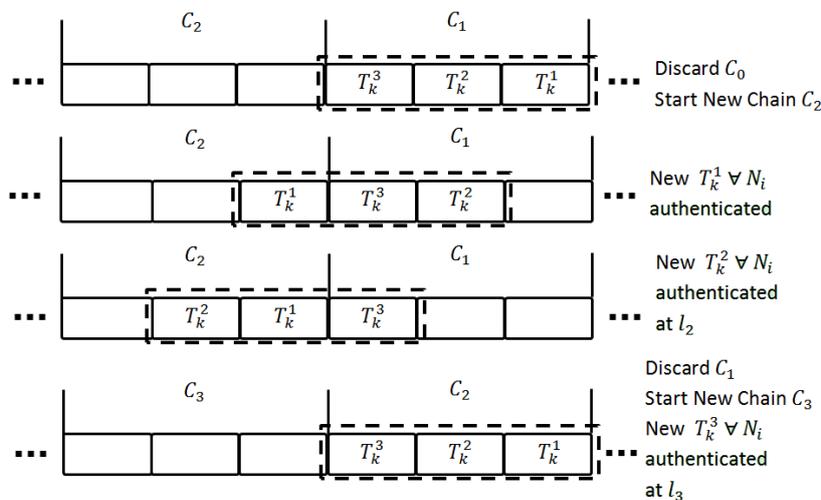


Figure 4. Managing the chain of the key generator in the sink.

Step 1: The sink node discards the previous chain C_0 and generates the commitment key generator G_0^j for the next chain C_2 . Step 2: The sink node reissues the ticket and session key to all sensor nodes authenticated at interval 1. Step 3: The sink node reissues the ticket and session key to all sensor nodes authenticated at interval 2. Step 4: The sink node reissues the ticket and session key to all sensor nodes authenticated at interval 3 and discards the previous chain and generates the commitment key generator G_0^j for the next chain.

4.2. Authentication Protocol Suite

When a sensor node joins the system, it goes through the Sensor Activation and Authentication Protocol (SAAP). After SAAP, N_i can establish multiple concurrent secure connections with sink nodes using the authentication ticket (T_k); similarly, N_i uses the T_k for re-authentication while moving across the network. Likewise, when a user node U_i joins the system, it goes through User Activation and Authentication Protocol (UAAP); subsequently, U_i can use the authentication ticket (T_k) for authorization to collect data from multiple sink and sensor nodes. In a concurrent run of multiple instances of the protocol, the message authentication plays a critical part in preventing the replay attack and to achieve the objectives of the authentication protocol as defined in [19,20]. In an SMSN message, authentication is accomplished by a secure exchange of a randomly generated nonce challenge. Moreover, in all the protocols discussed below, if the protocol initiator (user or sensor node) does not hear the response to an authentication/switch request, the protocol initiator resends the authentication/switch request including a new nonce and 'resend' flag. This step helps detect the impersonation, replay, and parallel session attack.

4.2.1. Sensor Activation and Authentication Protocol (SAAP)

Sink node S_j periodically broadcasts a *Hello* message ($BS_j||S_j$). If a node N_i wants to join the WSN, upon hearing the *Hello* message, N_i generates an encryption key $K_{TS}^i = H(BS_j \oplus n_S^i)$, encrypts the joining message with the generated key, and continues as follows:

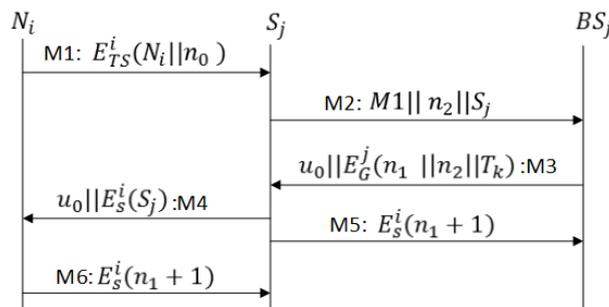


Figure 5. Message exchange for Sensor Activation and Authentication Protocol (SAAP).

- M1 As shown in Figure 5, N_i sends a JOIN message to sink S_j enclosing n_0 and encrypted with the generated encryption key K_{TS}^i .
- M2 Upon receiving the request from N_i , the S_j forwards the request in conjunction with its identity and challenge n_2 to base station BS_j .
- M3 BS_j retrieves the profile from the database, and if N_i is a legitimate sensor node, BS_j generates the key $K_{TS}^i = H(BS_j \oplus n_S^i)$, and sends $u_0 = E_{TS}^i(n_0 + 1||n_1||T_k||T_R||K_S^i)$ to S_j in M3. M3 also includes ticket T_k , n_1 (a challenge for N_i), and n_2 (challenge response for the sink node), all encrypted with K_G^j . The sink node S_j verifies the challenge n_2 , stores n_1 and retrieves the profile and K_S^i from the ticket.
- M4 S_j forwards the u_0 to N_i . After a challenge $(n_0 + 1)$ verification, N_i accepts T_k and may start sending data to S_j .
- M5 S_j sends the challenge response $(n_1 + 1)$ to BS_j for the confirmation of a successful protocol run.

M6 After challenge $(n_1 + 1)$ confirmation, S_j starts accepting sensor data; otherwise, it marks T_k as an invalid ticket.

In the SAAP, a secure exchange of n_0 ensures the message authentication between the sensor node and the base station, n_2 between the sink node and base station, and n_1 between the base station and the sink node, while message authentication between the sensor and sink nodes is established by session key encryption and n_1 . If N_i is already registered with BS_j , in M1 n_0 can be replaced with the hash of the password value.

4.2.2. Sensor Re-Authentication Protocol -1 (SRP1)

If sensor node N_i wants to establish multiple secure connections with sinks $S_j \in G_i$ or when a N_i moves from $S_k \rightarrow S_j$ such that $\{S_k, S_j\} \in G_i$, the Re-Authentication Protocol -1 continues as follows:

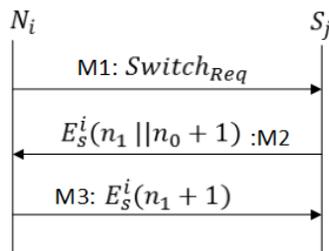


Figure 6. Message exchange for Sensor Re-Authentication Protocol-1 (SRP1).

M1 As shown in Figure 6, N_i sends $Switch_{Req} = E_S^i(N_i || H(n_0^i)) || T_k$ to S_j . The sink S_j decrypts the ticket, retrieves K_s^i , calculates the hash of n_0^i , and makes a comparison with $H(n_0^i)$ received in the switch message. If the value $H(n_0^i)$ does not match, the S_j will ignore the request and otherwise proceed as follows.

M2 S_j sends a challenge response along with the new challenge encrypted with session key K_s^i .

M3 N_i sends the challenge response n_1 . After challenge confirmation, S_j starts accepting data; otherwise, it marks T_k as an invalid ticket.

In the above procedure, secure exchange of n_0^i ensures the message authentication between the sensor node and the sink node. With this feature, a sensor node can establish multiple secure sessions with various sink nodes as shown in Figure 1.

4.2.3. Sensor Re-Authentication Protocol -2 (SRP2)

If sensor node N_i wants to establish another secure connection with sinks $S_j \in G_k^o$ or when a N_i moves from $S_k \rightarrow S_j$ such that $S_j \in G_k^o$, the re-authentication procedure proceeds as follows:

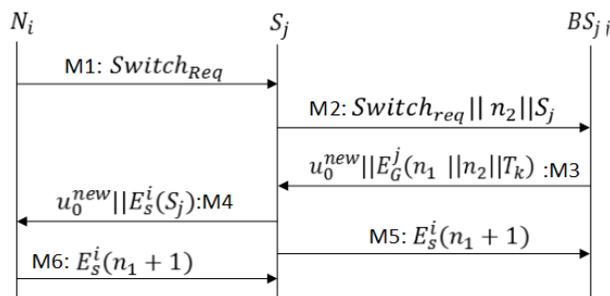


Figure 7. Message exchange for Sensor Re-Authentication Protocol-2 (SRP2).

M1 As shown in Figure 7, N_i sends $Switch_{Req} = E_S^i(N_i || H(n_0^i)) || T_k$ to S_j . The sink S_j decrypts the second half of T_k and verifies the identities of N_i and ticket granting base station.

- M2 After identities verification the sink S_j forwards the request in conjunction with a challenge n_2 to base station BS_j .
- M3 BS_j retrieves the profile from T_k , and if N_i is a legitimate sensor node, the BS_j generates the key $K_{TS}^i = H(BS_j \oplus n_S^i)$, and sends $u_0^{new} = E_{TS}^i(H(n_0)||n_1||T_k||T_R||K_S^i)$ to S_j in M3. M3 also includes ticket T_k , n_1 (a challenge for N_i), and n_2 (challenge response for the sink node), all encrypted with K_G^j . The sink node S_j verifies the challenge n_2 , stores n_1 and retrieves the profile and K_S^i from the ticket.
- M4 S_j forwards u_0^{new} to N_i . After challenge $(n_0 + 1)$ verification N_i accepts T_k and start sending data to S_j .
- M5 S_j sends the challenge response $(n_1 + 1)$ to BS_j ; it confirms a successful protocol run.
- M6 After challenge $(n_1 + 1)$ confirmation S_j start accepting data; otherwise, it marks T_k as an invalid ticket.

In the SRP2, a secure exchange of n_0 ensures the message authentication between the sensor node and the base station, n_2 between the sink node and base station, and n_1 between the base station and the sink node, while message authentication between the sensor and sink nodes is established by session key encryption and n_1 . If N_i wants to share data in a secret mode, both N_j and S_j generate a private session key $K_{sp}^j = H(K_S^i, n_0^i, n_1)$. With this feature, a sensor node can establish multiple secure sessions with various sink nodes as shown in Figure 1.

4.2.4. User Activation and Authentication Protocol (UAAP)

In some scenarios a user may desire to access data from the sensor network, for example, in IoT applications such as smart homes and smart buildings, a smartphone user may want to get sensor node data. In SMSN authentication protocol suites, an authenticated user (holding a valid ticket) can access data directly from sensor nodes and/or can collect from the sink node. The ticket structure for U_i is the same as the ticket structure discussed above for N_i where the sensor node identity and profile are replaced with the user node identity and profile. The user profile information includes the permissible accessibility information. In the User Activation and Authentication Protocol (UAAP) a user can acquire a ticket from BS_j in two different ways. If the user is not in the communication range of BS_j it routes the joining message via the nearest sink node S_j ; the protocol proceeds exactly as in the SAAP. However, if the user is in the communication range of BS_j , the user U_i generates an encryption key $K_{TS}^i = H(BS_j \oplus n_S^i)$, encrypts the joining message with the generated key, and proceeds as follows:

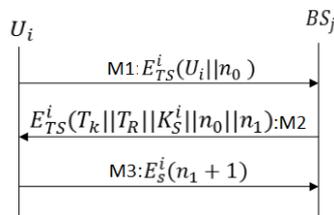


Figure 8. Message exchange for User Activation and Authentication Protocol (UAAP).

- M1 As shown in Figure 8, U_i sends a JOIN message enclosing n_0 and the user identity to base station BS_j .
- M2 BS_j retrieves profile from the database, and if U_i is a legitimate user, the BS_j generates authentication ticket, and send along with, n_1 (a challenge for U_i), and n_0 (challenge response), all encrypted with $K_{TS}^i = H(BS_j \oplus n_S^i)$. The ticket structure is same except the secret nonce n_1 enclosed inside ticket is generated by BS_j .
- M3 The user node U_i verifies the challenge n_0 , stores T_k and n_1 . U_i sends the challenge response $(n_1 + 1)$ to BS_j ; it confirms a successful protocol run . After receiving a challenge response $(n_1 + 1)$ the BS_j updates the status of U_i from idle to active user.

In User Activation and Authentication Protocol, secure exchange of n_0 ensures the message authentication between user node and base station and n_1 between the base station and user node.

4.2.5. User-Sink Authentication Protocol (USiAP)

After acquiring the authentication ticket, if user U_k wants to retrieve data from sink nodes S_j , it sends a JOIN request in conjunction with a ticket to S_j and then follows the same procedure as discussed in Sensor Re-Authentication Protocols 1 and 2; except after ticket verification, S_j can piggyback data with the rest of the messages. Using the authentication ticket, user U_k can also establish multiple concurrent connections with various sink nodes.

4.2.6. User- Sensor Authentication Protocol (USeAP)

After acquiring the authentication ticket, if user U_k wants to access data directly from sensor nodes N_i , it sends a JOIN request in conjunction with a ticket to N_i and the procedure proceeds as follows:

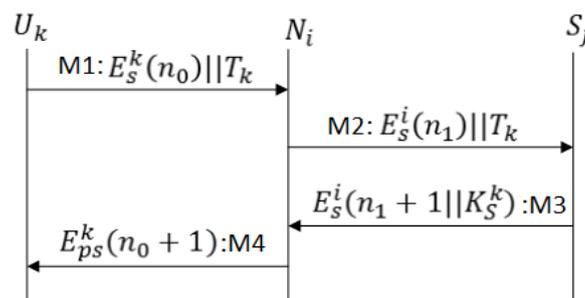


Figure 9. Message exchange for User-Sensor Authentication Protocol (USeAP).

- M1 As shown in Figure 9, U_k sends a JOIN message in conjunction with ticket and challenge n_0 encrypted with its ticket centered session key.
- M2 Upon receiving the request from U_k the N_i forward the ticket and encrypted challenge n_1 to sink node s_j .
- M3 Sink decrypts the ticket and retrieves U_k 's profile and session key K_S^k , and if U_i is a legitimate user node, S_j sends K_S^k and challenge response $(n_1 + 1)$ to N_i .
- M4 After challenge verification, N_i generates a private session key $K_{ps}^k = H(K_S^k, n_0)$ and sends a challenge response encrypted with the private session key. U_k also generates the private session key and verifies the challenge response.

In User-Sensor Authentication Protocol suite, secure exchange of n_0 ensures the message authentication between users and sensor node, while the secure exchange of n_1 ensures the message authentication between sensor node and sink node.

5. Security Analysis

This section presents the comprehensive security analysis of the SMSN protocol, including an informal security analysis and discussion of a formal security analysis using BAN logic [32], and finally presents the Scyther [17,18] implementation result of the SMSN and previously proposed schemes [11–16].

5.1. Informal Analysis and Discussion

To verify the strength of the SMSN protocol against known attacks we introduce an intruder in the network with capabilities as follows: It has an initial information set that contains the IDs of all users, sensor nodes, sink nodes and base stations. It can intercept and record message exchanges between

participating entities. It can redirect, spoof, and replay the messages. The subsequent sections show that the intruders, with all the above-mentioned capabilities, fail to launch a successful replay, parallel session, man-in-middle, impersonation, and several other attacks against the SMSN protocol suite.

5.1.1. Replay, Multiplicity, Parallel and Man in Middle Attacks Against the SMSN

We introduce an intruder, as discussed above, in the network and launch replay, multiplicity, parallel session, and man-in-middle attacks against the SMSN for three different scenarios, as shown in Figures 10–12. The intruder impersonates a protocol participant, intercepts the messages, and replays them to deceive other protocol participants. Replay, multiplicity, parallel, and man-in-middle attacks against the SAAP for three different scenarios are given below.

Scenario 1:

For the given scenario in Figure 10, let us suppose a sensor node N_i sends a request for authentication to sink $S_j \sim BS_j$. During the protocol run an intruder $Z(N_i)$ intercepts the messages and replays them to another sink $S_i \sim BS_j$; the attack proceeds as follows: The intruder $Z(N_i)$ intercepts M1 and replays it to S_i . The attack is detected immediately when BS_j receives two M2 messages enclosing the same M1. BS_j sends M3 to both sinks comprising the 'Alert' flag and a different n_1 nonce challenge. $Z(N_i)$ intercepts M6 and replays it to S_i ; note that the intercepted message M6 comprises a different n_1 which is the only valid response for a sink S_j . Upon receiving the wrong challenge response, the sink S_i identifies the intruder node. The SAAP not only detects the replay attack but also identifies the intruder.

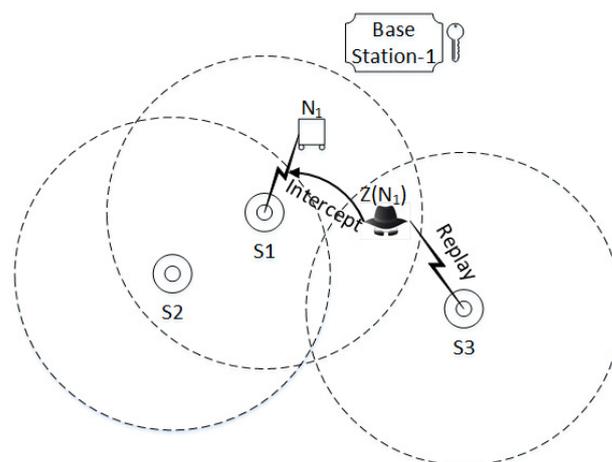


Figure 10. The intruder Z impersonates the protocol initiator N_1 (can be a sensor or user node). Intruder $Z(N_1)$ intercepts the messages between N_1 and sink S_1 and replays them to sink S_2 . Both sinks are associated with base station 1 and share the same keychain.

Scenario 2:

For the given scenario in Figure 11, let us suppose a sensor node N_i sends a request for authentication to sink $S_j \sim BS_j$. During the protocol run an intruder $Z(N_i)$ intercepts the messages and replays them to another sink $S_i \sim BS_i$; the attack proceeds as follows: The intruder $Z(N_i)$ intercepts M1 and replays it to S_i . Unlike in scenario 1, neither base station BS_j and BS_i can detect the attack at this stage and replies with a normal M3 to associated sink nodes S_i and S_j , respectively. However, both M3 messages comprise a different n_1 nonce challenge. $Z(N_i)$ intercepts the M6 and replays it to S_i ; note that the intercepted message M6 contains a different n_1 which is the only valid response for the sink S_j . Upon receiving the wrong challenge response, the sink S_i identifies the intruder.

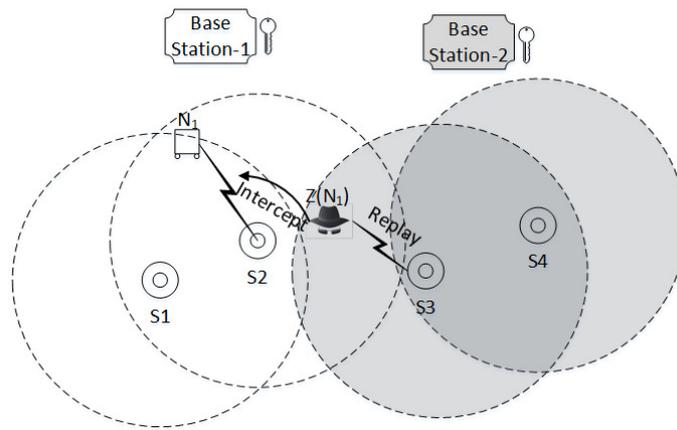


Figure 11. The intruder Z impersonates the protocol initiator N_1 (can be a sensor or user node). Intruder $Z(N_1)$ intercepts the messages between N_1 and sink S_2 and replays them to sink S_3 . Sink S_2 is associated with base station 1, and Sink S_3 is associated with base station 2.

Scenario 3:

For the given scenario in Figure 12, let us suppose two intruders impersonate the sink $S_j \sim BS_j$ and sensor node N_i ; the intruder $Z(N_i)$ is within the region of $S_i \sim BS_i$, and intruder $Z(S_j)$ is outside somewhere close to node N_i . Furthermore, both intruders can communicate through a private link with zero delay. During the protocol run intruder $Z(S_j)$ intercepts the messages sent by sensor node N_i and replays it to BS_j ; also $Z(S_j)$ shares all the intercepted messages with fellow intruder $Z(N_i)$ via a private secure channel; the attack proceeds as follows: The intruder $Z(S_j)$ intercepts M_1 sent by N_i and shares the intercepted message with $Z(N_i)$. The intruder $Z(N_i)$ sends the intercepted message to $S_i \sim BS_i$; the protocol proceeds normally and upon receiving M_4 the intruder $Z(N_i)$ sends the message M_4 to $Z(S_j)$. The intruder $Z(S_j)$ sends M_4 to sensor node N_i ; upon receiving M_6 the intruder $Z(S_j)$ shares the message M_6 with $Z(N_i)$. The intruder $Z(N_i)$ sends M_6 to $S_i \sim BS_i$ and the attack is completed. The attack is only successful if the replay of M_6 is delivered to S_j within a time interval of Td/L where L is the length of the keychain. Moreover, in practice when the private link between fellow intruders adds a communication delay, N_i can detect the attack by comparing the TR (registration time sent in u_0) with the local time.

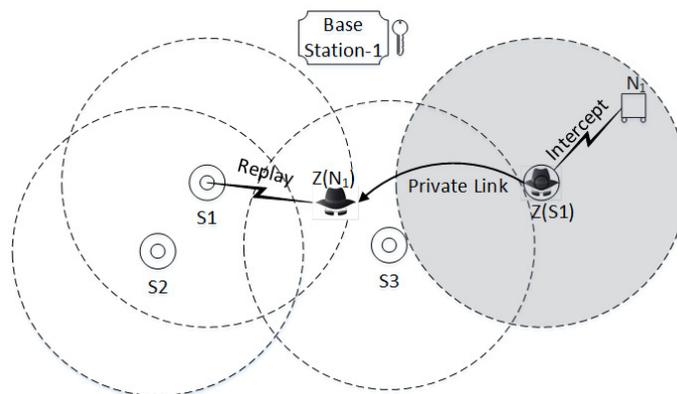


Figure 12. Two intruders impersonate the sink node $S_1 \sim BS_1$ and sensor node N_1 ; the intruder $Z(N_1)$ is within the region of $S_1 \sim BS_1$, and intruder $Z(S_1)$ is outside somewhere close to node N_1 . Both intruders can communicate with zero communication delay via a private link.

As the outcome of the above attack, N_i considers that the authentication process was completed successfully. However, the intruders cannot get any useful information during a protocol run: $Z(S_j)$

does not know the session key delivered to N_i , so $Z(S_j)$ cannot further communicate with N_i . Due to the unavailability of a data link, N_i uses the ticket to run SRP1/2. However, a problem exists on the other side: sink S_i and base station BS_i consider that they authenticated a legitimate sensor N_i successfully; from their point of view, the sensor N_i is within the region of BS_i but in reality, N_i is in the region of BS_j . Similarly, in the case of SRP1, SRP2, UAAP, USiAP, and USeAP the intruder Z fails to launch successful attacks for scenarios 1 and 2; however, the intruder Z completes the attack in scenario 3.

In a nutshell, replay, multiplicity, parallel and man-in-middle attacks against the SMSN protocol are not successful. Even though in scenario 3 the intruders completed the attack, they could not cause a serious security issue because after the completion of the protocol run, the intruder could not get useful information such as the session key, ticket, or partial session key. Furthermore, the attack can be avoided by introducing a timestamp in each message exchange. For illustration, the intruders need to wait for a significantly longer time to replay M4 and M6, and this significant delay can be detected with the time stamp, which reveals the existence of the intruder in the network.

5.1.2. Black Hole Attack

In a black hole attack [46–48] the intruder impersonates a node and blocks or drops the messages upon receiving them. In the SMSN, the sensor and user nodes can connect to multiple sink nodes simultaneously; hence, failure of data exchange on one route does not block the data delivery towards the base station. Moreover, the black hole attack is detectable in our scheme because the SMSN ensures binding by employing an exchange of secret nonce between $N_i \leftrightarrow S_j$, $N_i \leftrightarrow BS_j$, and $B_j \leftrightarrow Z$. Consecutive failures of exchange of challenge detects the black hole attack. Once the black hole is detected, the sensor node can send data via another sink node.

5.1.3. Wormhole Attack

In a wormhole attack [49,50], the intruder captures the messages in one location and tunnel to another location to a fellow intruder who replays the tunneled messages in another location area. The attack discussed in scenario 3 can be regarded as a wormhole attack. From the point of view of replay, multiplicity, parallel and man-in-middle attacks, the attack in scenario 3 did not achieve its objectives, but from the point of view of the wormhole attack, the attack is successful. The solution for the problem is similar to the one we discussed earlier: the attack can be avoided by introducing a timestamp in each message exchange.

5.1.4. Analytical Attacks

In an analytic attack [31,51], the intruder intercepts the messages and using cryptanalysis tries to recover a cryptographic key. With the inclusion of a time-based key, our scheme inherits the freshness property, which defies the capability of the intruder to launch analytical attacks, as it has a max time of T_d to acquire the time-based key, which makes it difficult to launch analytical attacks.

5.1.5. Topological Centered Attacks

In [52,53] the authors presented an authentication protocol for a sensor network in which the sink issues a re-authentication ticket that includes a list of neighbor sink nodes. This information can lead to topological centered attacks [27,28] such as identity replication attacks. In our scheme, the topological information is entirely obscured from the sink and sensor nodes.

5.2. Formal Analysis Using BAN Logic

The BAN logic [32] is a widely used formal method for the formal analysis of security protocols. To prove the security of the SMSN protocol suite it is sufficient to demonstrate the security of the SAAP and UAAP protocols; the rest of the protocols are extensions of the SAAP and UAAP and use the ticket

and session key established in the SAAP and UAAP protocols run. Hence, proof of SAAP and UAAP protocols concludes the security of the SMSN protocol suite.

The three basic objects of BAN logic are principals, formula/statements, and encryption keys. The principals, the protocol participants, are represented by symbols P and Q . The formula/statements are symbolized by X and Y and represents the content of the message exchanged. The encryption keys are symbolized by K . The logical notations of BAN-logic used for our analysis is given below:

- $P \models X$: P believes X , or P would be enabled to believe X ; in conclusion, P can take X as true.
- $P \prec X$: P sees/receives X . P initially has or received a message X and P can see the contents of the message and is capable of repeating X .
- $P | \sim X$: P once said X . P has sent a message including the statement X . However, the freshness of message is unknown.
- $P \Rightarrow X$: P controls X and should be trusted for formula/statement X .
- $\#(X)$: X is fresh; it says, X never sent by any principal before.
- $P \stackrel{K}{\longleftrightarrow} Q$: P and Q shares a key K to communicate in a secure way and K is only known to P , Q and a trusted principal.
- $(X)_K$: The statement X is encrypted by key K .
- $\{X\}_Y$: It stand for X combined with Y . Y is anticipated to be secret and its implicit or explicit presence proves the identity of a principal who completes the $\{X\}_Y$.

Some primary BAN-logic postulates used in the analysis of the SMSN are given below:

- Message meaning rules: $\frac{P \models P \stackrel{K}{\longleftrightarrow} Q, P \prec (X)_K}{P \models Q | \sim X}, \frac{P \models P \stackrel{Y}{\longleftrightarrow} Q, P \prec X_Y}{P \models Q | \sim X}$
- Nonce verification rule: $\frac{P \models \#(X), P \models Q | \sim X}{P \models Q \models X}$
- Jurisdiction rule: $\frac{P \models Q \Rightarrow X, P \models Q \models X}{P \models X}$
- Freshness rule: $\frac{P \models \#(X)}{P \models (X, Y)}$
- Believe rule: $\frac{P \models Q \models (X, Y)}{P \models X, P \models Y}$
- Session key rule: $\frac{P \models Q \#(X), P \models Q \models X}{P \models P \stackrel{K}{\longleftrightarrow} Q}$

5.2.1. BAN Logic Analysis of SAAP

The SAAP protocol should achieve the following goals:

- G1 $N_i \models (N_i \stackrel{K_S^i}{\longleftrightarrow} S_j)$
- G2 $N_i \models S_j \models (N_i \stackrel{K_S^i}{\longleftrightarrow} S_j)$
- G3 $S_j \models (N_i \stackrel{K_S^i}{\longleftrightarrow} S_j)$
- G4 $S_j \models N_i \models (N_i \stackrel{K_S^i}{\longleftrightarrow} S_j)$

Protocol Idealization:

- I1 $N_i \xrightarrow{Via S_j} BS_j : \{n_0, (N_i \stackrel{n_0^i}{\longleftrightarrow} BS_j)\}_{H(X_S)}$
- I2 $S_j \rightarrow BS_j : \{n_2, IDS_j\}$
- I3 $BS_j \rightarrow S_j : \{(n_1, n_2, (N_i \stackrel{K_S^i}{\longleftrightarrow} S_j, \#(N_i \stackrel{K_S^i}{\longleftrightarrow} S_j), n_0)_{K_i^i = H(\zeta_i^j, H(n_0^i))}, Vi)_{K_G^i}, \{n_0, n_1, (N_i \stackrel{K_S^i}{\longleftrightarrow} S_j, \#(N_i \stackrel{K_S^i}{\longleftrightarrow} S_j), n_0)_{K_i^i = H(\zeta_i^j, H(n_0^i))}, Tr, N_i \stackrel{K_S^i}{\longleftrightarrow} S_j, \#(N_i \stackrel{K_S^i}{\longleftrightarrow} S_j), N_i \stackrel{n_1^i}{\longleftrightarrow} BS_j\}_{H(X_S)}\}$
- I4 $S_j \xrightarrow{Via BS_j} N_i : \{n_0, n_1, (N_i \stackrel{K_S^i}{\longleftrightarrow} S_j, \#(N_i \stackrel{K_S^i}{\longleftrightarrow} S_j), n_0)_{K_i^i = H(\zeta_i^j, H(n_0^i))}, Tr, N_i \stackrel{K_S^i}{\longleftrightarrow} S_j, \#(N_i \stackrel{K_S^i}{\longleftrightarrow} S_j), N_i \stackrel{n_1^i}{\longleftrightarrow} BS_j\}_{H(X_S)}$

- I5 $S_j \rightarrow N_i : (IDS_j)_{K_S^i}$
 I6 $N_i \rightarrow S_j : (n_1)_{K_S^i}$

Initial State Assumptions:

- A1 $BS_j \models \#(n_0)$
 A2 $BS_j \models \#(n_2)$
 A3 $S_j \models \#(n_1)$
 A4 $N_i \models \#(n_1)$
 A5 $N_i \models (N_i \xrightarrow{K_{TS=H(X_S)}} BS_j)$
 A6 $BS_j \models (N_i \xrightarrow{K_{TS=H(X_S)}} BS_j)$
 A7 $S_j \models (S_j \xleftrightarrow{K_G^j} BS_j)$
 A8 $BS_j \models (S_j \xleftrightarrow{K_G^j} BS_j)$
 A9 $BS_j \models N_i \models (N_i \xrightarrow{K_{TS=H(X_S)}} BS_j)$
 A10 $N_i \models BS_j \models (N_i \xrightarrow{K_{TS=H(X_S)}} BS_j)$
 A11 $BS_j \models S_j \models (S_j \xleftrightarrow{K_G^j} BS_j)$
 A12 $S_j \models BS_j \models (S_j \xleftrightarrow{K_G^j} BS_j)$

Let us analyze the protocol to show that N_i and S_j share a session key:

From I1, we have

$$BS_j \prec \{n_0, (N_i \xleftrightarrow{n_S^i} BS_j)\}_{H(X_S)} \quad (1)$$

The (1), A6 and message meaning rule infers that

$$S_j \models N_i \sim \{n_0, (N_i \xleftrightarrow{n_S^i} BS_j)\} \quad (2)$$

The A1 and freshness conjunction comprehends that

$$BS_j \models \#\{n_0, (N_i \xleftrightarrow{n_S^i} BS_j)\} \quad (3)$$

The (2), (3) and nonce verification rule deduces that

$$BS_j \models \{N_i \models n_S^i, n_0, (N_i \xleftrightarrow{n_S^i} BS_j)\} \quad (4)$$

The (4) and believe rule infers that

$$BS_j \models N_i \models (N_i \xleftrightarrow{n_S^i} BS_j) \quad (5)$$

From A2, (5) and jurisdiction rule, it concludes

$$BS_j \models (N_i \xleftrightarrow{n_S^i} BS_j) \quad (6)$$

This belief confirms that BS_j has received a message from a legitimate N_i .

From I2, we have

$$BS_j \prec n_2 \quad (7)$$

The (7) and message meaning it infers that

$$BS_j \models S_j \sim n_2 \quad (8)$$

The A2, A1, (3) and freshness concatenation comprehends that

$$BS_j \models \#\{n_0, n_2, (N_i \xleftrightarrow{n_s^i} BS_j)\} \quad (9)$$

According to nonce freshness, this proves that BS_j confirms that N_i is recently alive and running the protocol with BS_j .

From I3, we have

$$S_j \prec (n_1, n_2, (N_i \xleftrightarrow{K_S^i} S_j, \#(N_i \xleftrightarrow{K_S^i} S_j), n_0)_{K_i^i = H(\zeta_i^i, H(n_0^i))}, V_i)_{K_G^i} \quad (10)$$

The A7 and (10) deduce that

$$S_j \models BS_j \mid \sim \{n_1, n_0^i, N_i \xleftrightarrow{K_S^i} S_j, \#(N_i \xleftrightarrow{K_S^i} S_j), V_i\} \quad (11)$$

The A3, (11) and freshness concatenation comprehends that

$$S_j \models \#\{n_1, n_0^i, N_i \xleftrightarrow{K_S^i} S_j, \#(N_i \xleftrightarrow{K_S^i} S_j), V_i\} \quad (12)$$

The (11), (12) and nonce verification rule infers that

$$S_j \models BS_j \models \{n_1, n_0^i, N_i \xleftrightarrow{K_S^i} S_j, \#(N_i \xleftrightarrow{K_S^i} S_j), V_i\} \quad (13)$$

The (13) and believe rule comprehends that

$$S_j \models BS_j \models (N_i \xleftrightarrow{K_S^i} S_j) \quad (14)$$

The logic belief proves that S_j is confident and believes that K_S^i is issued by BS_j ; moreover, the freshness of the key also suggests that BS_j is alive and running the protocol with S_j and N_i .

The (13), (14) and jurisdiction rule concludes that (15 Goal-3)

$$S_j \models (N_i \xleftrightarrow{K_S^i} S_j) \quad (15)$$

From I4, we have

$$N_i \prec \{n_1, T_R, N_i \xleftrightarrow{K_S^i} S_j, \#(N_i \xleftrightarrow{K_S^i} S_j), N_i \xleftrightarrow{n_s^i} BS_j\}_{H(X_S)} \quad (16)$$

The (16), A5 and message meaning rule comprehends that

$$N_i \models BS_j \mid \sim \{n_1, T_R, N_i \xleftrightarrow{K_S^i} S_j, \#(N_i \xleftrightarrow{K_S^i} S_j)\} \quad (17)$$

The (17), A4 and freshness concatenation rule infers that

$$N_i \models \#\{n_1, T_R, N_i \xleftrightarrow{K_S^i} S_j, \#(N_i \xleftrightarrow{K_S^i} S_j)\} \quad (18)$$

The (17), (18) and nonce verification rule deduce that

$$N_i \models BS_j \models \{n_1, T_R, N_i \xleftrightarrow{K_S^i} S_j, \#(N_i \xleftrightarrow{K_S^i} S_j)\} \quad (19)$$

The (19) and believe rule infers that

$$N_i \models BS_j \models \{N_i \xleftrightarrow{K_S^i} S_j, \} \quad (20)$$

The (19), (20) and jurisdiction rule concludes that (21 Goal-1)

$$N_i \models \{N_i \xleftrightarrow{K_S^i} S_j, \} \quad (21)$$

From I5, we have

$$N_i \prec IDS_j \quad (22)$$

The (15), (21), (22) and meaning rule comprehends that (23 Goal-4)

$$S_j \models N_i \models \{N_i \xleftrightarrow{K_S^i} S_j, \} \quad (23)$$

From I6, we have

$$S_j \prec n_1 \quad (24)$$

The (15), (21), (23) and nonce verification rule deduce that (25 Goal-2)

$$N_i \models S_j \models \{N_i \xleftrightarrow{K_S^i} S_j, \} \quad (25)$$

1. BAN Logic Analysis of UAAPP:

The UAAP protocol should achieve the following goals:

- G1 $U_i \models (U_i \xleftrightarrow{K_S^i} S_j)$
- G2 $BS_j \models U_i \models (U_i \xleftrightarrow{K_S^i} S_j)$
- G3 $U_i \models BS_j \models (U_i \xleftrightarrow{K_S^i} S_j)$

Idealization of UAAP:

- I1 $U_i \rightarrow BS_j : \{n_0, (U_i \xleftrightarrow{n_S^i} BS_j)\}_{H(X_S)}$
- I2 $BS_j \rightarrow U_i : \{n_0, n_1, (U_i \xleftrightarrow{K_S^i} S_j, \#(U_i \xleftrightarrow{K_S^i} S_j), n_0)_{K_i^i=H(\zeta_j^i, H(n_0^i))}, (V_i)_{K_G^i}, TR, N_i \xleftrightarrow{K_S^i} S_j, \#(U_i \xleftrightarrow{K_S^i} S_j), U_i \xleftrightarrow{n_S^i} BS_j\}_{H(X_S)}$
- I3 $U_i \rightarrow BS_j : (n_1)_{K_S^i}$

Initial State Assumptions of UAAP:

- A1 $BS_j \models \#(n_0)$
- A2 $U_i \models \#(n_1)$
- A3 $N_i \models (U_i \xleftrightarrow{K_{TS}=H(X_S)} BS_j)$
- A4 $BS_j \models (U_i \xleftrightarrow{K_{TS}=H(X_S)} BS_j)$
- A5 $S_j \models N_i \models (N_i \xleftrightarrow{K_{TS}=H(X_S)} BS_j)$
- A6 $N_i \models BS_j \models (N_i \xleftrightarrow{K_{TS}=H(X_S)} BS_j)$

Let us analyze the protocol to show that UAAPP achieves the mentioned goals:

From I1, we have

$$BS_j \prec \{n_0, (U_i \xleftrightarrow{n_S^i} BS_j)\}_{H(X_S)} \quad (26)$$

The (26), A4 and message meaning rule infers that

$$BS_j \models U_i \sim \{n_0, (U_i \xleftrightarrow{n_S^i} BS_j)\} \quad (27)$$

The A1 and freshness conjunction comprehend that

$$BS_j \models \#\{n_0, (U_i \xleftrightarrow{n_s^i} BS_j)\} \quad (28)$$

The (27), (28) and nonce verification rule deduces that

$$BS_j \models \{U_i \models n_s^i, n_0, (U_i \xleftrightarrow{n_s^i} BS_j)\} \quad (29)$$

The (29) and believe rule infers that

$$BS_j \models U_i \models (U_i \xleftrightarrow{n_s^i} BS_j) \quad (30)$$

From A1, (30) and jurisdiction rule, it concludes

$$BS_j \models (U_i \xleftrightarrow{n_s^i} BS_j) \quad (31)$$

This belief confirms that BS_j has received a message from a legitimate N_i .

From M2, we have

$$U_i \prec \{n_1, T_R, U_i \xleftrightarrow{K_s^i} S_j, \#\{U_i \xleftrightarrow{K_s^i} S_j\}, U_i \xleftrightarrow{n_s^i} BS_j\}_{H(X_S)} \quad (32)$$

The (32), A3 and message meaning rule comprehends that

$$U_i \models BS_j \sim \{n_1, T_R, U_i \xleftrightarrow{K_s^i} S_j, \#\{U_i \xleftrightarrow{K_s^i} S_j\}\} \quad (33)$$

The (33), A2 and freshness conjunction rule infers that

$$U_i \models \#\{n_1, T_R, U_i \xleftrightarrow{K_s^i} S_j, \#\{U_i \xleftrightarrow{K_s^i} S_j\}\} \quad (34)$$

The (33), (34) and nonce verification rule deduce that

$$U_i \models BS_j \models \{n_1, T_R, U_i \xleftrightarrow{K_s^i} S_j, \#\{U_i \xleftrightarrow{K_s^i} S_j\}\} \quad (35)$$

The (35) and believe rule infers that (36 Goal-2)

$$U_i \models BS_j \models \{U_i \xleftrightarrow{K_s^i} S_j, \} \quad (36)$$

The (35), (36) and jurisdiction rule concludes that (37 Goal-1)

$$U_i \models \{U_i \xleftrightarrow{K_s^i} S_j\} \quad (37)$$

From I3, we have

$$BS_j \prec n_1 \quad (38)$$

The (36), (37), (38) and meaning rule comprehends that (39 Goal-3)

$$BS_j \models U_i \models \{U_i \xleftrightarrow{K_s^i} S_j, \} \quad (39)$$

Table 1. Scyther tool parameter settings.

Parameter	Settings
Number of Runs	1~3
Matching Type	Find all Type Flaws
Search pruning	Find All Attacks
Number of pattern per claim	10

Table 2. Comparison of authentication properties.

Claims	H. Tseng [11]			Yoo et al. [12]			Kumar et al. [13]			Quan et al. [14]			Farash et al. [15]			Y. Lu et al. [16]			SMSN		
	U_i	N_i	GW	U_i	N_i	GW	U_i	N_i	GW	U_i	N_i	GW	U_i	N_i	GW	U_i	N_i	GW	U_i	N_i	GW
Aliveness	N	N	N	N	N	N	Y	N	N	Y	N	N	Y	Y	N	N	N	Y	Y	Y	Y
Weak Agreement	N	N	N	N	N	N	Y	N	N	Y	N	N	Y	Y	N	N	N	Y	O	Y	Y
Non-injective Agreement	N	N	N	N	N	N	O	N	N	Y	N	N	Y	Y	N	N	N	Y	O	Y	Y
Non-injective Synch.	N	N	N	N	N	N	O	N	N	Y	N	N	Y	Y	N	N	N	Y	O	Y	Y

N = Authentication claim is not fulfilled; Y = Authentication claim is fulfilled; O = Authentication claim is fulfilled but falsified for protocol instances >3.

5.3. Verifying Protocol Using Scyther Tool

The previous section proved that according to the BAN logic the SMSN is a secure authentication scheme. The BAN logic provided a foundation for the formal analysis of security protocols, but few attacks can slip through the BAN logic [32]. For further proof of the strength of the SMSN protocol suite, we implemented the SMSN and [11–16] schemes in the automated security protocol analysis tool, Scyther [17,18]. Our proposed scheme provides a strong defense against known attacks in the presence of an intruder (Z) which is capable of regulating the communication channel, redirecting, spoofing, replaying or blocking the messages. It has initially known information, e.g., IDs and public keys of all users, and any intercepted message is additional information to the current information set (S), i.e., $S \cup \{m\}$. It can generate a fresh message from known data, e.g., $S \vdash m$, and can run multiple instances of the protocol. The Scyther tool verifies the protocol claims and checks the possibility of attacks against the protocol. The claims are the event that describes the design and security properties of the authentication protocol. We consider four claims as defined below; for a detailed description of protocol claims, please refer to [19,20].

In Scyther the protocol is modeled as an exchange of messages among different participating 'roles'; for instance, in sensor node authentication, the sensor node is in the role of initiator, the sink is in the role of responder and the base station is in the role of a server. The Scyther tool integrates the authentication properties into the protocol specification as a claim event. We tested our protocol [11–16] employing claims, as mentioned earlier, with the parameter settings given in Table 1.

The results are shown in Table 2. It is clear that in the presence of an intruder (as defined above), our protocol qualifies all the protocol claims and no attacks were found. Hence, for a large number of systems and scenarios, our protocol guarantees safety against a large number of known attacks, such as impersonating, man-in-middle and replay attacks, etc. In contrast, [11–16] are susceptible to several attacks and failed to fulfill the authentication claims. Moreover, in protocol schemes [11–16], there is a lack of sender-receiver binding verification, and an intruder can exploit this situation to impersonate a sensor node and run multiple instances of the protocol to launch multiplicity and man-in-middle attacks.

6. Performance Analysis

Although the SMSN authentication protocol suite covers authentication procedures for both sensor and user nodes, for the sake of simplicity, we compare the efficiency of the SMSN with user authentication protocols. We compared the efficiency of the SMSN user authentication protocols considering computational cost, message complexity and time synchronization requirements to that of [11–16]. Unlike in [11–16], the SMSN allows the user to be authenticated with both sensor and sink nodes.

The total computation cost is estimated as the sum of the total number of E = encryptions/decryptions, M = Multiplications, H = hash, X = XOR, and T = Time Synchronization operations. Moreover, we assume that cryptographic hash and symmetric encryption/decryption operations have computational complexity similar to $O(m)$, where m is the size of the message. All the schemes in [11–16] considered that the registration process took place via a secure channel. We assume that in all schemes the user node U_i and Gateway GW_j exchange the registration information using a secure encrypted channel. Furthermore, one encrypted unicast message requires two E operations, one for encryption and one for decryption. Similarly, a broadcast message to N recipient adds the $(N + 1)E$ operations in the total computational complexity.

Table 3. Performance comparison of SMSN with well-known user authentication protocols.

Schemes	Phase	Comp. Complexity	Comm. Complexity	Comm. Cost in Bytes	Time Synch.
H. Tseng [11]	Registration	$(4 + N)E + 1H$	$2UC + 1BC$	$(2N + 3)int$	-
	Login-Authentication	$6H + 1X$	$4UC$	$2CK + 7int$	$1T$
Yoo et al. [12]	Registration	$(4 + N)E + 5H + 2X$	$2UC + 1BC$	$(5 + N)CK + 1int$	-
	Login-Authentication	$19H + 2X$	$6UC$	$5CK + 8int$	$1T$
Kumar et al. [13]	Registration	$4H + 3X$	$2UC$	$5CK + 3int$	-
	Login-Authentication	$14H + 2X$	$3UC$	$6CK + 11int$	$2T$
Quan et al. [14]	Registration	$12E + 9H + 1X + 7M$	$4UC$	$4CK + 9int + 4f$	$4T$
	Login-Authentication	$18H + 2X$	$4UC$	$8CK + 16int$	$4T$
Farash et al. [15]	Registration	$4E + 6H + 2X$	$2UC$	$4CK + 1int$	$2T$
	Login-Authentication	$30H + 16X$	$4UC$	$17CK + 5int$	$4T$
Y. Lu et al. [16]	Registration	$4E + 10H + 2X$	$2UC$	$4CK + 1int$	-
	Login-Authentication	$8E + 17H + 15X$	$4UC$	$4CK + 18int$	$3T$
SMSN (User-Sink)	Registration	$8E + 5H$	$3UC$	$3CK + 10int$	-
	Login-Authentication	$8E + 2H$	$3UC$	$3CK + 8int$	<i>Optional</i>
SMSN (User-Sensor)	Registration	$8E + 5H$	$3UC$	$3CK + 10int$	-
	Login-Authentication	$9E + 2H$	$4UC$	$5CK + 12int$	<i>Optional</i>

Computational Complexity: E = encryptions/decryptions, Ex = modular exponentiations, M = multiplications, H = hash operation, X = XOR operation, N = Total number of nodes, T = Time Synchronization operation; *Communication Complexity and Cost:* BC = Broadcast Message, UC = Unicast Message, CK = 32 bytes (Represents Size of Cryptographic KeyHash and Signature), Int = 4 bytes (Represents Size of nonce, Node Ids and Integers), f = 8 bytes (Represents Size of the floating point real number).

From Table 3 we can see that computational complexity of SMSN authentication protocols in the registration phase is slightly more expensive compared to Kumar et al. [13] and Farash et al. [15]; however, in the authentication phase, the SMSN authentication protocols completely outperform Kumar et al. [13] and Farash et al. [15]. H. Tseng [11] and Yoo et al. [12] are the most computationally expensive schemes during the registration phase, but in the authentication phase, H. Tseng [11] is the most efficient scheme followed by the SMSN. However, regarding sensor node computational efficiency, in our scheme, the overall workload of a sensor node is very low. Moreover, unlike the [12–14,21] schemes, the SMSN does not require time synchronization between the Gateway and user node.

The communication complexity is calculated as the sum of the total unicast and broadcast message exchange. Figure 13 shows the overall communication complexity in a WSN when the number of sensor nodes is fixed in the network, and the number of new users' requests is constantly increasing. The message complexity of [11,12] increases multiplicatively by increasing the number of nodes; conversely, it grows slowly in the case of the SMSN, [13–15]. Figure 14 shows the overall communication complexity for various network sizes with the same number of user requests. This metric is only useful for a WSN with mobile sensor and user nodes. The message complexity of [11,12] increases rapidly with an increase in the number of new users' requests; conversely, it remains constant with the SMSN, [13–15]. This suggests that the schemes proposed in [11,12] are not suitable for highly dynamic mobile WSNs where the frequency of leaving and joining the network is high.

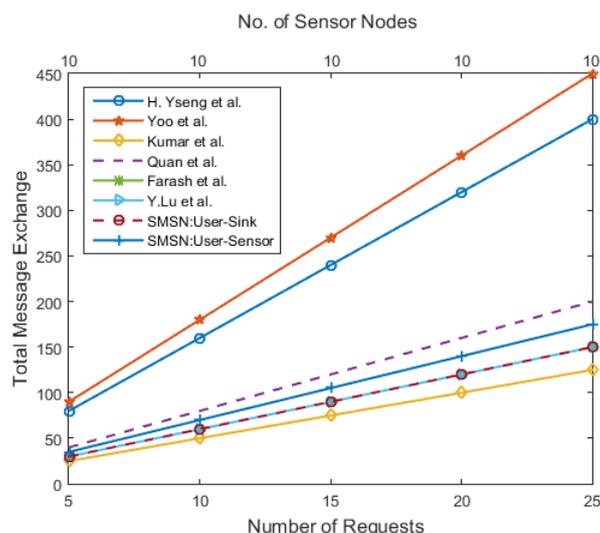


Figure 13. The total number of message exchanged between constant number of sensor nodes with different number of new users' requests.

However, a more interesting comparison in terms of communication efficiency is the comparison based on the amount of data exchanged during the protocol run. The numerical results are taken for a dynamic and mobile sensor network consisting of 100 nodes. The probability that a new user may join the network and an existing user may leave the system defines how frequently the users join and leave the network. The average communication cost per user is calculated for the dynamic probability of 0.05 to 0.5, and the results are shown in Figure 15.

SMSN user-sink authentication outperforms all other schemes from less dynamic to highly dynamic networks. However, in a less dynamic network with a dynamic probability of less than 0.05, the SMSN user-sensor authentication is slightly more expensive than the scheme of H. Tseng [11]. Even though in a less dynamic system the SMSN user-sensor authentication is slightly more expensive than [11], the performance gap decreases, and for highly mobile and dynamic networks, SMSN performs better than the scheme of H. Tseng [11].

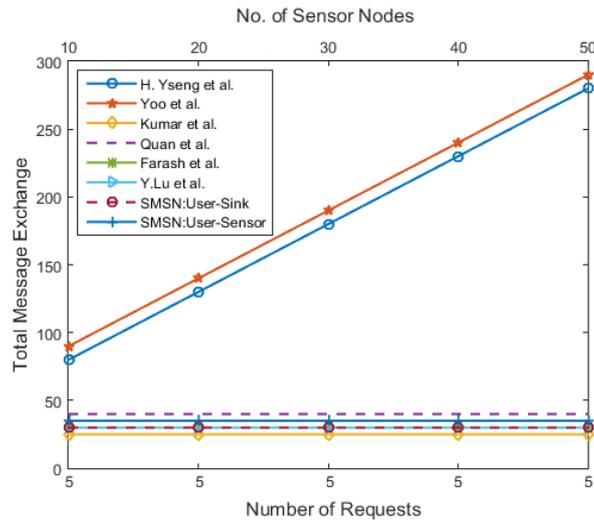


Figure 14. The total number of message exchanged for constant number of new users request in different network size.

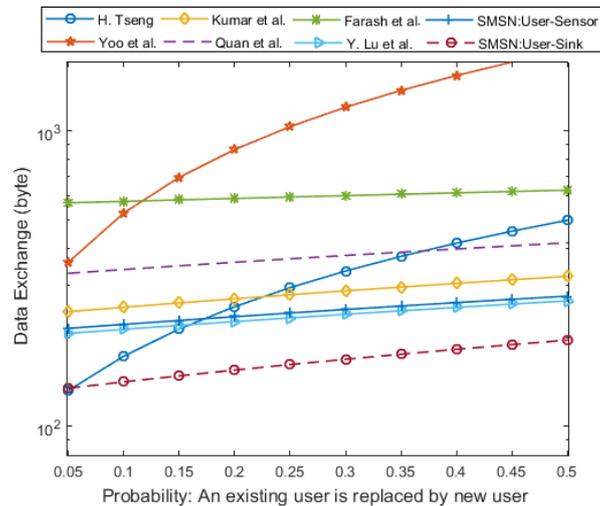


Figure 15. The amount of data exchanged for less to highly dynamic sensor network.

7. Conclusions

Due to the recent growth in WSN technologies, we have observed an enormous paradigm shift in sensor network applications. The authentication and security goals of a sensor network have become more crucial and challenging. Most of the user and sensor node authentication schemes for WSNs have been developed without taking into account the requirements of integrating WSNs with emerging technologies such as IoT. We developed an SMSN scheme considering the requirements of mobile and dynamic WSN applications as discussed in Section-III. We noted that the user authentication schemes designed for sensor networks [11–16] do not meet the authentication properties; for example, the execution of these schemes in the Scyther tool revealed that the participating entities failed to achieve wider objectives (defined as protocol claims in Section 5-C) of the authentication protocol. Finally, we compared the efficiency of the SMSN user authentication protocols with the schemes in [11–16]. We observed that concerning the computational cost, our scheme is slightly more expensive compared to [13,15] during the registration phase but the SMSN totally outperforms both in the authentication phase. Regarding message complexity our proposed scheme totally outperforms [11–15]; however, the performance of the scheme of Y. Lu et al. [16] is close to the SMSN. Finally, unlike the schemes in [11–16],

the SMSN does not require time synchronization between the Gateway (base station) and the user node. The main focus of this work was to discuss and provide solutions for the emerging challenge that has emerged from the integration of the WSN in IoT applications. To prove the usability of the proposed scheme, we made a comprehensive security and performance analysis and simulated the proposed idea in an automated protocol verifier tool, the Scyther. However, for future work, it will be interesting to investigate the usability of the SMSN by implementing it on an application specific testbed. Moreover, in the near future it will be possible to incorporate the basic Internet functionality in the sensor node. We believe it will further enhance the application scenarios for the SMSN; for instance a user device will be able to collect real-time sensor data remotely via the Internet. Moreover, we are further investigating the usage of SMSN for the promising future internet architecture, known as Name-Data-Networking (NDN), which is extensively studied in the literature [54–59]. In NDN the contents verification is achieved by the use of asymmetric cryptography. We argue that in future especially in IoT application scenarios the devices will be resource constraint devices; for instance, the sensor network is going to be the part of IoT. For resource constraint devices the asymmetric cryptography is computationally expensive. We believe that SMSN protocol suit with some modifications can be a suitable candidate for NDN internet architecture.

Acknowledgments: This work was supported by the Ministry of Trade, Industry & Energy (MOTIE, Korea) under Industrial Technology Innovation Program. No. 10065742, Standardization of Vertical and Horizontal Smart-factory Integration.

Author Contributions: Muhammad Bilal proposed the scheme, and performed the security and performance analysis. He also wrote the experiment code, conducted the experiments and evaluated the performance of the proposed scheme. Shin-Gak Kang amended the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

WSN	Wireless Sensor Network
IoT	Internet of Things
SMSN-Protocol	Secure Mobile Sensor Network Protocol
BAN-logic	Burrows Abadi Needham logic
SAAP	Sensor Activation and Authentication Protocol
SRP1	Sensor Re-Authentication Protocol-1
SRP2	Sensor Re-Authentication Protocol-2
UAAP	User Activation and Authentication Protocol
USiAP	User-Sink Authentication Protocol
USeAP	User- Sensor Authentication Protocol

References

1. Koyama, K.; Maurer, U.M.; Okamoto, T.; Vanstone, S.A. New public-key schemes based on elliptic curves over the ring Z_n . In *Advances in Cryptology—CRYPTO '91*; Springer: Berlin/Heidelberg, Germany, 1992; pp. 252–266.
2. Rivest, R.L. The RC5 encryption algorithm. In *International Workshop on Fast Software Encryption*; Springer: Berlin/Heidelberg, Germany, 1994; pp. 86–96.
3. Wong, Y.; Attebury, G.; Ramamurthy, B.A. Survey of security issues in wireless sensor networks. *IEEE Commun. Surv.* **2006**, *8*, 2–23.
4. Zhang, J.; Varadharajan, V. Wireless sensor network key management survey and taxonomy. *J. Netw. Comput. Appl.* **2010**, *33*, 63–75.
5. Panic, G.; Stecklina, O.; Stamenkovic, Z. An embedded sensor node microcontroller with crypto-processors. *Sensors* **2016**, *16*, 607.
6. Kim, D.; An, S. PKC-Based DoS attacks-resistant scheme in wireless sensor networks. *IEEE Sens. J.* **2016**, *16*, 2217–2218.

7. Hancke, G.P.; Silva, B.C.; Hancke, G.P., Jr. The Role of Advanced Sensing in Smart Cities. *Sensors* **2012**, *13*, 393–425.
8. Sagl, G.; Resch, B.; Blaschke, T. Contextual sensing: Integrating contextual information with human and technical geo-sensor information for smart cities. *Sensors* **2015**, *15*, 17013–17035.
9. Flammini, A.; Sisinni, E. Wireless sensor networking in the internet of things and cloud computing era. *Procedia Eng.* **2014**, *87*, 672–679.
10. Bilal, M.; Kang S.G. Time-Assisted authentication protocol. *Int. J. Commun. Syst.* **2017**, doi:10.1002/dac.3309.
11. Tseng, H.R.; Jan R.H.; Yang, W. An improved dynamic user authentication scheme for wireless sensor networks. In Proceedings of the IEEE Global Communications Conference, Washington, DC, USA, 26–30 November 2007.
12. Yoo, S.G.; Park, K.Y.; Kim, J. A security-performance-balanced user authentication scheme for wireless sensor networks. *Int. J. Distrib. Sens. Netw.* **2012**, *2012*, 382810.
13. Kumar, P.; Gurtov, A.; Ylianttila, M.; Lee, S.; Lee, H. A strong authentication scheme with user privacy for wireless sensor networks. *ETRI J.* **2013**, *35*, 889–899.
14. Zhou, Q.; Tang, C.; Zhen, X.; Rong, C. A secure user authentication protocol for sensor network in data capturing. *J. Cloud Comput. Adv. Syst. Appl.* **2015**, *4*, 6.
15. Farash, M.S.; Turkanovi, M.; Kumari, S.; Hölbl, M. An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the Internet of Things environment. *Ad Hoc Netw.* **2016**, *36*, 152–176.
16. Lu, Y.; Li, L.; Peng, H.; Yang, Y. An energy efficient mutual authentication and key agreement scheme preserving anonymity for wireless sensor networks. *Sensors* **2016**, *16*, 837.
17. Cremers, C. The scyther tool: Verification, falsification, and analysis of security protocols. In *International Conference on Computer Aided Verification*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 414–418.
18. Cremers, C.; Mauw, S. Security properties. In *Operational Semantics and Verification of Security Protocols*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012.
19. Lowe, G. A hierarchy of authentication specifications. In Proceedings of the 10th on Computer Security Foundations workshop, Canton, MA, USA, 10–12 June 1997.
20. Cremers, C.; Mauw, S.; De Vink, E.P. Injective synchronisation: An extension of the authentication hierarchy. *Theor. Comput. Sci.* **2006**, *367*, 139–161.
21. Wong, K.H.M.; Zheng, Y.; Cao, J.; Wang, S. A dynamic user authentication scheme for wireless sensor networks. In Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, Taichung, Taiwan, 5–7 June 2006.
22. Das, M.L. Two-Factor user authentication in wireless sensor networks. *IEEE Trans. Wirel. Commun.* **2009**, *8*, 1086–1090.
23. Han, K.; Kim, K.; Shon, T. Untraceable mobile node authentication in WSN. *Sensors* **2010**, *10*, 4410–4429.
24. Han, K.; Shon, T. Efficient mobile sensor authentication in smart home and WPAN. *IEEE Trans. Consum. Electron.* **2010**, *56*, 591–596.
25. Yeh, H.; Chen, T.; Liu, P.; Kim, T.; Wei, H. A secured authentication protocol for wireless sensor networks using elliptic curves cryptography. *Sensors* **2011**, *11*, 4767–4779.
26. Kumari, S.; Om, H. Authentication protocol for wireless sensor networks applications like safety monitoring in coal mines. *Compu. Netw.* **2016**, *104*, 137–154.
27. Kumari, S.; Khan, M.K.; Atiquzzaman, M. User authentication schemes for wireless sensor networks: A review. *Ad Hoc Netw.* **2015**, *27*, 159–194.
28. Xie, M.; Han, S.; Tian, B.; Parvin, S. Anomaly detection in wireless sensor networks: A survey. *J. Netw. Comput. Appl.* **2011**, *34*, 1302–1325.
29. Ko, L. A novel dynamic user authentication scheme for wireless sensor networks. In Proceedings of the IEEE International Symposium on Wireless Communication Systems, Reykjavik, Iceland, 21–24 October 2008.
30. Chen, T.; Shih, W. A robust mutual authentication protocol for wireless sensor networks. *ETRI J.* **2010**, *32*, 704–712.
31. Khan, M.K.; Alghathbar, K. Cryptanalysis and security improvements of 'two-factor user authentication in wireless sensor networks'. *Sensors* **2010**, *10*, 2450–2459.

32. Burrows, M.; Abadi, M.; Needham, R.M. A logic of authentication. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*; The Royal Society: London, UK, 1989; Volume 426, pp. 233–271.
33. Lee, C.-C.; Li, C.-T.; Chen, S.-D. Two attacks on a two-factor user authentication in wireless sensor network. *Parallel Process. Lett.* **2011**, *21*, 21–26.
34. Gligor, V.D.; Kailar, R.; Stubblebine, S.; Gong, L. Logics for cryptographic protocols-virtues and limitations. In *Proceedings of the IEEE Computer Security Foundations Workshop IV*, Franconia, NH, USA, 18–20 June 1991; pp. 219–226.
35. Boyd, C.; Mao, W. On a limitation of BAN logic. In *Advances in Cryptology- EUROCRYPT '93*; Springer: Berlin/Heidelberg, Germany, 1993; pp. 240–247.
36. Kumar, P.; Lee, S.; Lee, H. E-SAP: Efficient-Strong authentication protocol for healthcare applications using wireless medical sensor networks. *Sensors* **2012**, *12*, 1625–1647.
37. He, D.; Kumar, N.; Chen, J.; Lee, C.; Chilamkurti, N.; Yeo, S. Robust anonymous authentication protocol for health-care applications using wireless medical sensor networks. *Multimed. Syst.* **2015**, *21*, 49–60.
38. Armando, A.; Basin, D.; Boichut, Y. The AVISPA Tool for the Automated Validation of Internet Security Protocols and Applications. Available online: <https://www.inf.ethz.ch/personal/basin/pubs/avispa05.pdf> (accessed on 22 September 2016).
39. Li, C.-T.; Weng, C.-Y.; Lee, C.-C. An advanced temporal credential-based security scheme with mutual authentication and key agreement for wireless sensor networks. *Sensors* **2013**, *13*, 9589–9603.
40. Bilal, M.; Kang S.G. A secure key agreement protocol for dynamic group. *Clust. Comput.* **2017**, doi:10.1007/s10586-017-0853-0.
41. Perrig, A.; Canetti, R.; Tygar, J.D.; Song, D. The TESLA Broadcast Authentication Protocol. Available online: https://www.cs.berkeley.edu/~tygar/papers/TESLA_broadcast_authentication_protocol.pdf (accessed on 16 June 2016).
42. Daemen, J.; Rijmen, V. AES Proposal: Rijndael. National Institute of Standards and Technology. Available online: <https://csrc.nist.gov/archive/aes/rijndael/Rijndael-ammended.pdf> (accessed on 25 September 2016).
43. Schneier, B.; Kelsey, J.; Whiting, D.; Wagner, D.; Hal, C. Twofish: A 128-bit Block Cipher. Available online: <https://www.schneier.com/academic/paperfiles/paper-twofish-paper.pdf> (accessed on 25 September 2016).
44. Burwick, C.; Coppersmith, D.; D'Avignon, E.; Gennaro, R.; Halevi, S.; Jutla, C.; Matyas, S.M.; O'Connor L.; Peyravian M.; Safford D.; et al. MARS-a Candidate Cipher for AES. Available online: <http://www.nada.kth.se/kurser/kth/2D1449/99-00/mars.pdf> (accessed on 25 September 2016).
45. Schneier, B. Description of a new variable-length key, 64-bit block cipher (Blowfish). In *Fast Software Encryption*; Springer: Berlin/Heidelberg, Germany, 1994; pp. 191–204.
46. Kumar, V.; Kumar, R. An adaptive approach for detection of blackhole attack in mobile Ad hoc network. *Proced. Comput. Sci.* **2015**, *48*, 472–479.
47. Misra, S.; Bhattarai, K.; Xue, G. BAMBi: Blackhole attacks mitigation with multiple base stations in wireless sensor networks. In *Proceedings of the IEEE International Conference on Communications*, Kyoto, Japan, 5–9 June 2011.
48. Poongodi, T.; Karthikeyan, M. Localized secure routing architecture against cooperative black hole attack in mobile Ad hoc networks. *Wirel. Pers. Commun.* **2016**, *90*, 1039–1050.
49. Hu, Y.; Perrig, A.; Johnson, D.B. Wormhole attacks in wireless networks. *IEEE J. Sel. Areas Commun.* **2006**, *24*, 370–380.
50. Imran, M.; Khan, F.A.; Jamal, T.; Durad, M.H. Analysis of detection features for wormhole attacks in MANETs. *Proced. Comput. Sci.* **2015**, *56*, 384–390.
51. Choi, Y.; Lee, Y.; Won, D. Advances in computer science and ubiquitous computing. In *Cryptanalysis on Symmetric Key Techniques Based Authentication Scheme for Wireless Sensor Networks*; Springer: Berlin/Heidelberg, Germany, 2015; Volume 373, pp. 7–13.
52. Zhou, Y.; Fang, Y. Defend against topological attacks in sensor networks. In *Proceedings of the IEEE Military Communications Conference*, Atlantic City, NJ, USA, 17–20 October 2005.
53. Gabrielli, A.; Mancini, L.V.; Setia, S.; Jajodia, S. Securing Topology Maintenance Protocols for sensor networks. *IEEE Trans. Dependable Secur. Comput.* **2011**, *8*, 450–465.

54. Bouk, H.S.; Ahmed, S.H.; Kim, D.; Song, H. Named-Data-Networking-Based ITS for smart cities. *IEEE Commun. Mag.* **2017**, *55*, 105–111.
55. Bilal, M.; Kang S.G. A cache management scheme for efficient content eviction and replication in cache networks. *IEEE Access* **2017**, *5*, 1692–1701.
56. Ahmed, S.H.; Bouk, H.S.; yaqub, M.A.; Kim, D.; Song, H. CODIE: Controlled data and interest evaluation in vehicular named data networks. *IEEE Trans. Veh. Technol.* **2016**, *65*, 3954–3963.
57. Bilal, M.; Kang S.G. Time aware least recent used (TLRU) cache management policy in ICN. In Proceedings of the IEEE 16th International Conference on Advanced Communications Technology, Pyeongchang, Korea, 16–19 February 2014; pp. 528–532.
58. Ahmed, S.H.; Bouk, H.S.; Kim, D. RUFFS: RobUst forwarder selection in vehicular content-centric networks. *IEEE Commun. Lett.* **2015**, *19*, 1616–1619.
59. Ahmed, S.H.; Kim, D. Named data networking-based smart home. *ICT Express* **2016**, *2*, 130–134.



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).