Egyptian Informatics Journal 22 (2021) 15–26

Contents lists available at ScienceDirect

Egyptian Informatics Journal

journal homepage: www.sciencedirect.com

Full length article

AAA-WSN: Anonymous access authentication scheme for wireless sensor networks in big data environment

Shadi Nashwan

Department of Computer Science, College of Computer and Information Sciences, Jouf University, Sakaka, Saudi Arabia

ARTICLE INFO

Article history:
Received 4 November 2019
Revised 30 December 2019
Accepted 20 February 2020
Available online 9 March 2020

Keywords:
Big data
Mutual authentication
Perfect forward secrecy
Real-time data environments
User anonymity
Wireless sensor networks

ABSTRACT

The data capturing and access process is an important stage in the big data applications. Most of these applications are exploited the wireless sensor networks (WSNs) to accomplish this process through the sensor nodes that are deployed in unsecure and unattended environments. Therefore, these applications are suffered from numerous security weaknesses, an adversary may exploit such weaknesses to break user's privacy, confidentiality of sensor nodes, and control the communication channel between the network components. Consequently, the security issues for this process have attracted much interest of the researchers with the increasing spread of use such applications. The majority of the proposed authentication schemes fail to solve all existing security weaknesses simultaneously. Thus, the authentication scheme is a critical issue in the WSNs. This paper proposes anonymous access authentication scheme for wireless sensor networks in big data environments (AAA-WSN) to achieve appealing security services. Comparing with the recent WSNs authentication schemes, the AAA-WSN scheme cannot only achieve strong security services such as user anonymity and mutual authentication, but also performs the perfect forward secrecy feature with high level of efficiency. The security analysis shows that the AAA-WSN scheme is resistant to the current known attacks. Moreover, the performance analysis in terms of the storage, computations and communications costs demonstrates that the AAA-WSN scheme achieves high level of security with desirable level of efficiency comparing the recent WSNs authentication schemes.

1. Introduction

The big data applications are deployed widely in different real-time data environments such as Healthcare, Smart homes, Industrial, Military, Agriculture, and Environmental applications. The basic stages of big data applications can be categorized as: (1) data capturing and access process; (2) data storage process; (3) data integration process; (4) data interpretation process; and (5) data extracting process [1,29].

The data capturing and access process has gained a great attention for the researchers mainly in the information security field for the following challenges: (1) most of big data applications are based on the wireless sensor networks (WSNs), which normally contain a large number of the sensor nodes that are described by limited computational capabilities, energy resources, storage, and bandwidth within an unsecure and unattended environments; (2) in some cases within the real-time data environments, the external users can access the real- time data from sensor node directly; and (3) comparing with other application domains, the number of authentication sessions between the users and the sensor nodes is relatively much higher [1].

For example, to monitor a patient in healthcare applications using the Wireless medical sensor networks (WMSN), different body sensor nodes are used to capture the vital signs of a patient, such as body temperature, heart rate, blood pressure and respiratory rate as real-time data of the patient [25,28]. In regular situations, the medical workers are able to collect the patient's data indirectly through the gateway node (GWN) of the healthcare service provider from the patient's sensor nodes. But in the urgent alarm case to reduce the communication delay time, the physicians can access the patient's sensor nodes through the cluster head node directly to prepare the medical feedback report within short time and to take the medical decisions before escalade the patient problem.

Fig. 1 illustrates the different ways of real-time data access by the legitimate users in the WSNs. Normally, the users can access the data from the sensor nodes indirectly through the GWN of service provider as trusted node. Nevertheless, in some cases the GWN forces the users to access the real-time data from the sensor

Email address: Eshadi_nashwan@ju.edu.sa,
Peer review under responsibility of Faculty of Computers and Information, Cairo University.

https://doi.org/10.1016/j.eij.2020.02.005
1110-8665/

nodes directly [1,2]. Hence, these applications are categorized as internet of thing (IOT) systems where the communication between the GWN of service providers, users and sensor nodes take place through the internet during the access data process. Although only the legitimate user that proves his/her identity can be accessed the WSNs resources, the large number of sensor nodes within an unsecure and unattended environment leads to open the door to a set of security weaknesses. Therefore, an adversary can exploit the existing weaknesses to break the sensor nodes privacy and control the communication channel between the network components during the data capturing and access process.

The common and known security attacks that exploit these vulnerabilities are: (1) user impersonate attack; (2) sensor node spoofing attack; (3) replay attack; (4) Man-in-the-middle attack; (5) wrong password login/update attack; (6) de-synchronization attack; (7) smart card loss attack; (8) denial-of-service (DoS) attack; and (9) user anonymity violation attack.

In the last decade, with increasing the demand on the big data applications, many authentication schemes have been suggested to overcome the security weaknesses of the data capturing and access process by the WSNs. It is plausible to point that none of these schemes have presented an integrated solution for WSNs to be against all the above mentioned attacks. In general, the authentication scheme plays an important role to inquire and spread the real-time data of the WSNs in secure manner through utilizes different cryptography methods such as RSA crypto, hashing techniques and ECC crypto [3–8].

There are three main approaches to design the authentication schemes between (n) user and (m) sensor nodes in the WSNs that can be summarized as the following: (1) the authentication schemes that are based on establish symmetric key between each user and each sensor node; (2) the authentication schemes that are based on establish asymmetric key between the user and all sensor nodes; and (3) the authentication schemes that are based on establish one key for each authentication session between the user and sensor node through a trusted GWN component. In the first approach, each sensor node stores (n) keys and the WSNs will include long-term (nm) symmetric keys. Obviously, this approach is not suitable for the big data applications and the adversary can compromise the keys of sensor nodes in simply manner. Due to the limitations of the computational capabilities and communication resources properties of the sensor nodes, the second approach is not convenient for WSNs where the asymmetric cryptography methods are tending to be more system resources consuming and most of them are used complex mathematical operations with big numbers to be secured. The majority of authentication schemes that have proposed to defeat the common attacks of the WSNs are based on the third approach.

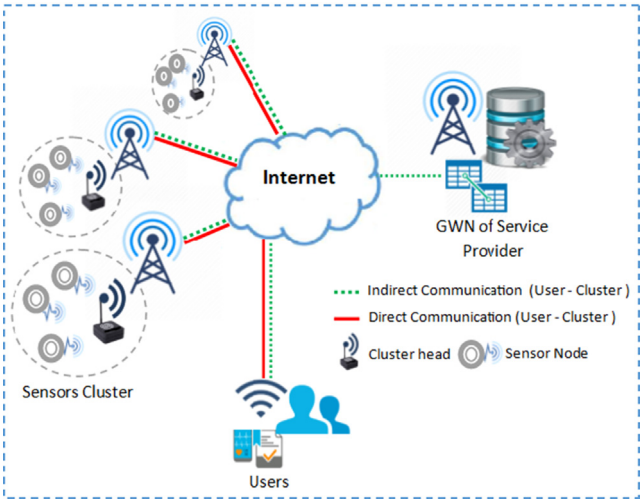

Fig. 1. Data access in WSNs.

In 2010, Khan and Alghathbar [9] analyzed the Das's scheme [10], the results shown that it included a set of threats, such as it could not have accomplished mutual authentication, and the password update for each authentication session is not secure. Therefore, the scheme could not resist the bypassing attacks, compromise attack, and privileged-insider attack. They also introduced an improved for this authentication scheme.

In the same year, both of Refs. [9,10] were evaluated by the Vaidya et al. [11], the results illustrated that both schemes have included a set of flaws such as cannot defeat the stolen smart card attacks. Besides, many researches have been conducted to enhance the Das's scheme by different ways of modifications, but were at the cost of the efficiency [12–14].

In 2011, Yeh et al. [7] introduced a secure authentication protocol for WSNs using elliptic curves cryptography. They claimed that their protocol was suitable for such environment according to different terms such as the computations and the communications costs with higher protection requirements.

In 2012, Das et al. [15] developed a dynamic password-based user authentication scheme for hierarchical WSNs. This scheme deployed AKA concepts with hash function to change the user's password in dynamic manner without back to the base station or gateway node. In 2013, the Turkanovic and Holbl [16] pointed out that Das et al.'s scheme [15] contained redundant elements and is infeasible for implementation in real-life environment. To overcome these imperfections, they also proposed an improved dynamic password-based user authentication scheme for hierarchical WSNs.

In 2014, Yuan [17] proposed a two-factor authentication scheme for wireless sensor networks. He claimed his scheme fulfills various security requirements such as the mutual authentication, non-repudiation, and is not susceptible to the attack due to a lost smart card based on biometric technique.

In 2015, Amin and Biswas [18] introduced a secure lightweight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks. This scheme concentrated on practical implementation to defeat a smart card theft attack by designed a novel architecture for the WSNs environment. They claimed their scheme fulfills complete security requirements containing specially energy efficiency, user anonymity, mutual authentication, and user-friendly password change in efficient manner.

Therefore, the author believes that the third approach is more secure and efficient when the GWN of service provider is responsible about establishment a shared session key for each authentication session between the user and sensor node. Consequently, the user can capture and access the real-time data from the sensor node in the next authentication sessions without back to the GWN.

Furthermore, the authentication scheme of WSNs that is designed for the big data environments have to include the following attractive requirements: (1) the authentication entities (i.e. users of the application, sensor nodes, and GWN of the service provider) have to authenticate each other successfully. Therefore, the authentication scheme can achieve the fully mutual authentication feature; (2) the authentication entities generate new temporary identification numbers for each authentication session to protect the real user's identity. Hence, the authentication scheme can provide the user anonymity and untraceability feature; (3) the session keys have to update after each successful authentication sessions to prevent the adversary from obtains the previous session keys. Therefore, the authentication scheme can support the perfect forward secrecy feature; and (4) the login and authentication service between the user and its smart card should depend on a three-factor authentication scheme to remove the security flaws that are

correlated with the a two-factor authentication. Thus, authentication scheme can defeat the smart card loss attack.

This paper proposes anonymous access authentication scheme for wireless sensor networks (AAA-WSN scheme) to provide a set of attractive security services for the big data environments. Comparing with the recent authentication schemes [19–21], the AAA-WSN scheme cannot provide only full mutual authentication and anonymity features, but also can support the perfect forward secrecy feature with low operations cost. The AAA-WSN scheme can prevent the existing known attacks using a set of the lightweight symmetric cryptography and hash functions. Moreover, the efficiency analysis in terms of the computations cost and communications messages costs of authentication process conducts that the proposed scheme is efficient for resource constrained sensor nodes platform. The remaining parts of this research are arranged as follows: Section 2 discusses the related works, the preliminaries of proposed scheme are presented in Section 3, Section 4 describes the AAA-WSN scheme stages, Sections 5 and 6 illustrate the security and performance analysis of the AAA-WSN scheme, respectively. Finally, the conclusion is given in Section 7.

2. Related works

In order to perform both of the security analysis and performance analysis of the proposed scheme in comparative manner, this section summarizes three related authentication schemes: (1) Lu et al. [19] scheme; (2) Jung et al. [20] scheme; and (3) Xiong et al. [21] scheme. The reason to choose these schemes is that they are exemplifying the last proposed authentication schemes for WSNs in the reported investigation with the following design specifications: (1) the symmetric encryption methods and hash functions are used as basic operations for verification process; and (2) the AKA principles are used to achieve the mutual authentication between all communication entities. The user node, GWN node, and sensor node represent the communication entities of these schemes, each entity has a unique identity, and the GWN communicates with the user and the sensor node via public channels using different secret keys.

In 2016, Lu et al. [19] proposed an energy efficient mutual authentication and key agreement scheme to preserve the user anonymity for WSNs. This scheme contains four stages: (1) user registration stage; (2) sensor node registration stage; (3) login and authentication stage; and (4) password change stage. In the first stage, the new user and the GWN exchange the registration messages via secure channels to accomplish the authentication parameters of the user's smart card. The overall computations of this stage contain: (1) generate two random numbers; and (2) execute the hash functions six times. A new sensor node is registered in the GWN through exchange the registration messages via private channel in second stage. The overall computations of this stage include: (1) generate symmetric key one time; and (2) execute the hash function one time. Through the third stage, the WSNs services can be accessed by the user where the mutual authentication is required as a mutual chain between all communication entities. The overall computations of this stage include: (1) generate four timestamps; (2) generate four random numbers; (3) generate authentication session keys two times; (4) execute the one-way hash functions at least nineteen times; (5) execute the encryption and decryption functions eight times; and (6) perform the verification processes nine times. The last stage is considered as an optional stage, the user's password can be updated. The overall computations of this stage include: (1) execute the one-way hash functions five times; and (2) verification process one time.

In the same year, Jung et al. [20] proposed an anonymous user authentication scheme using the key agreement method depend on symmetric key cryptography algorithms in WSNs. This scheme has the same structure, communication entities and functions of the Lu et al. [19] scheme with a few differences. The scheme contains just three stages: (1) user registration stage; (2) login authentication stage; and (3) the last stage to change the user password. This scheme has not included the sensor node registration stage, the secret key between the sensor node entity and the GWN entity is created when the sensor node is developed. In the first stage, the new user and the GWN exchange the registration messages via secure channels to accomplish the authentication parameters of the user's smart card. The overall computations of this stage include: (1) generate one random number; and (2) execute the hash functions four times. In the second stage, the mutual authentication is required between all communication entities to access the WSNs services by the user. The overall computations of this stage include: (1) generate four timestamps; (2) generate two random numbers; (3) generate authentication session keys two times; (4) execute the one-way hash functions at least thirteen times; (5) execute the encryption and decryption functions four times; and (6) perform the verification processes twelve times. In the last stage which is considered also as an optional stage, the user's password can be updated. The overall computations of this stage include: (1) execute the one-way hash functions five times; and (2) verification process one time.

In 2017, Xiong et al. [21] proposed anonymous authentication scheme for WSNs. Actually, the scheme contains of five stages: (1) user registration stage; (2) sensor node registration stage; (3) authentication and key agreement stage; (4) password change stage; and (5) dynamically deploy sensor nodes stage. In first stage, the GWN prepares the user's smart card as a replay to the registration request of the user through private channel. The user's pseudonym identity that is generated by GWN distinguishes this stage from the previous registration stages as in the Lu et al. [19] and Jung et al. [20]. The user's identity and password is determined by the user, the GWN generates a set of authentication parameters and stores them into the smart card, then the GWN updates the user's information table and sends the smart card to the user via secure channel. The overall computations in this stage include: (1) generate four random numbers; and (2) execute the hash functions four times. A new sensor node and the GWN exchange the registration messages via a secure channel in the second stage. The overall computations in this stage include: (1) generate one random number; and (2) initiate a sequence number. In the third stage, the user achieves the mutual authentication with the GWN and sensor node that needs to access. The overall computations in this stage include: (1) generate one timestamp; (2) generate two random numbers; (3) generate authentication session keys two times; (4) execute at least the one-way hash function twenty-five times; (5) execute the encryption and decryption functions four times; and (6) perform at least the verification process eight times. The user's password can be updated by the user himself through the fourth stage. The computations in this stage include: (1) execute the one-way hash functions six times; and (2) verification process one time. In the last stage, a new sensor node can be deployed by the system administrator, this process can be accomplished using the same steps of the sensor node registration stage.

The performance analysis of the Lu et al. [19] and Jung et al. [20] schemes observes that both schemes contain impractical GWN search operations (the GWN entity has to perform the exhaustive search operations to determine the random user identity that generated by the GWN entity itself including the time of decryption operations). Besides, the performance analysis of Xiong et al. [21] scheme detects that the scheme cannot avoid the unnecessary computation costs in case that the GWN entity received wrong pseudonym identity, the timestamp that is sent within the login

message from the user to GWN is checked after the verification process of the pseudonym identity.

Although, the Lu et al. [19] and Jung et al. [20] schemes have claimed that their schemes can fulfill a set of desirable security features and resist to enormous types of attacks. The security analysis detects that both schemes are suffered from different kinds of drawbacks such as: (1) no provision for perfect forward secrecy feature where the user's identity is fixed for all authentication session, the user's long-term key can be disclosed, and then the keys of previous authentication session will be retrieved [21,24]; (2) cannot implement the user anonymity and untraceability security features in real world [8,21].

In addition, the security analysis finds that the Lu et al. [19], Jung et al. [20] and Xiong et al. [21] schemes are vulnerable to the smart card loss attack since the user identity and password are not strength with low entropy [22,23,25]. Also all of these schemes cannot support the sensor node anonymity which naturally will lead in future to loss the user anonymity feature also. In addition to, theses authentication schemes cannot support the real-time communication between the user entity and the sensor node, where all the authentication sessions between them must be through the GWN entity.

3. Preliminaries

This section introduces some preliminaries of the proposed scheme (AAA-WSN scheme), such as list of notations, structure, security design requirements, and assumptions.

3.1. Notations

All the notations of the AAA-WSN scheme are listed in the Table 1.

Table 1
AAA-WSN scheme notations.

Notation	Description
Ui	The participant user
SC	The smart card
Sj	The sensor node that user needs to access
GWN	The gateway node
UIDi	The user identity
SIDj	The sensor node identity
IDi	User identity that used in the user side
IDip	Prefix Pseudonym identity
IDis	Suffix pseudonym identity
V	Verification code
XV	Expected verification code
*PWi0, PW*0i*	Old and new passwords of the user
PWi1	Secret code of the user
SNi	Session identity for user
SNj	Session identity of Sensor node
X	Secret code of GWN
SSj0	Initial sequential number for GWN
SSj1	Initial sequential number for Sensor node
h, h0, h1, h2, h3, h4	One way hash functions
r, r0, r1, r2, r3	Random numbers
X	Secret code of GWN for the user
SKij	Shared key between the user and Sensor node
Ki, Kj, SKi, SKj	Secret keys
E()	Encryption function
D()	Decryption function
A||k	Separate the A into k equal blocks
Fi	Hidden value of user secret code
T1, T2, T3, T4, T5	Timestamps
N, M, C, ΔT	Constants that are determined by the applications
||	String concatenation operation
⊕	XOR operation
Φ	Empty value

3.2. Structure

The proposed scheme involves three authentication entities: (1) the participant user node (Ui); (2) the Gateway node (GWN) of service provider; and (3) the sensor node (Sj) that the user needs to access. The AAA-WSN scheme consists of five stages: (1) the User registration stage between the Ui and GWN; (2) the Sensor node activation stage between the GWN and Sj; (3) the Login authentication stage between all authentication entities; (4) the Subsequence authentication stage between the Ui and Sj; and (5) the Password change phase between the Ui and user's smart card.

3.3. Security design requirements

The most prominent design requirements of the AAA-WSN scheme can be summarized as the following: (1) for each authentication session, the communication entities must authenticate each other and set up a reliable communication connection to exchange the information based on the AKA concepts; (2) the authenticity of each participant Ui must be verified before capture a new information from the Sj; (3) the information that are transmitted by the Sj must remain confidential from an adversary and only the authorized Ui can receive these information; (4) an adversary cannot modify the authentication messages to be similar to the original messages that are sent by the authorized authentication entities; (5) the session keys are established using secure method to be used by the authentication entities; (6) timestamps are used to verify the data freshness of the authentication messages; (7) the hash functions are used to conceal the identities for both of the Ui entity and Sj entity in the whole authentication sessions; finally, (8) in order to derive a new secret session key, the hash functions are used by the authentication entities to satisfy the perfect forward secrecy feature.

3.4. Assumptions

The AAA-WSN scheme is presented under a set of valid and widely accepted assumptions that will be used in the security analyzing section: (1) an adversary can retrieve the smart card information based on the power-consumption methods that have presented in Refs. [26,27]; (2) all authentication messages that are generated by the communication entities throughout the scheme execution are transmitted through unsecure communication channels. Therefore, an adversary can intercept, delete, capture, and retransmit the authentication messages over these channels; (3) an adversary can act as a legitimate communication entities where an adversary knowns how the AAA-WSN scheme can be performed; (4) the user registration stage is accomplished though secure channels between the user and the GWN where the GWN is considered as trusted node for the service provider; (5) the sensor node activation stage is accomplished though secure channels between the Sj and the GWN; and (6) the symmetric encryption and decryption functions use high entropy parameters. Thus, its hard an adversary to guess these parameters in polynomial time.

4. AAA-WSN scheme description

According to the notations that are mentioned in Table 1, the details of each stage of the AAA-WSN scheme will describe in the following subsections.

4.1. Sensor node activation stage

As illustrated in Fig. 2, when a new Sj is installed, the Sj sends the activation request message to the GWN, this message includes the SIDj that had assigned to the Sj when the Sj has developed.

Upon receiving the activation request message, the GWN executes the following steps: (1) randomly initiates the session identity SNj = r0; (2) derives a new secret key Kj = h2 (SIDj ⊕ SNj); (3) computes the verification code V0j = h2 (Kj||SNj); (4) computes a new IDj = h2 (SIDj||SNj); (5) sets the initial session numbers SSj0 = SS j1 = 0; (6) inserts the information of Sj record to sensor node database including [SIDj, SSj0, Kj, IDj, SNj]; finally (7) sends the activation values [SNj, and SSj1] to the Sj through secure channel, then the Sj stores the values [SNj, and SSj1] into its memory.

4.2. User registration stage

Suppose the new participant user Ui wishes to access the information that is captured by a specific sensor node Sj. Initially, the Ui requests to register into the GWN entity. As illustrated in Fig. 3, the GWN constructs the smart card (SC) for the Ui, the SC is considered as a response to the registration request message that is sent by the Ui. This stage is accomplished by a new Ui and the GWN. In order to prepare the registration request message, a new Ui executes the following steps: (1) selects the user identity UIDi; (2) selects password PWi0; (3) selects secret security code PWi1; (4) generates random number r; (5) computes IDPW = h (UIDi||PWi0||r); (4) transmits [UIDi, IDPW, and PWi1] to the GWN through secure channel as a registration request message. Upon receiving the registration request message from the Ui, the GWN verifies whether the user's UIDi is existed in the database of the users. If it exists, the GWN asks the Ui to choose another identity number and rejects the registration request. Otherwise, the GWN executes the following steps: (1) randomly initiates the session number SNi = h5 (r1); (2) derives a secret key Ki = h1 (UIDi||X||SNi), where X is a secret code that has generated by the GWN for a specific Ui; (5) computes Fi = Ki ⊕ PWi1; (3) computes the verification code V0i = h2 ((SNi|| PWi1)⊕ (IDPW||Ki)); (4) computes the prefix virtual identities IDi = IDip = h2 (UIDi||SNi); (5) initiates the suffix virtual identity IDis = Φ, where the Φ is a null value; (6) inserts the information of Ui record to the users database including [UIDi, IDi, IDip, IDis, X and SNi]; and (7) embeds the registration values [SNi, Fi, and V0i] into the SC; (8) gives the SC to the Ui; Finally, Ui stores the r in the SC.

4.3. Login authentication stage

In order to access the WSNs services, the Ui achieves the mutual authentication with the GWN and Sj that needs to access, a session key (SKij) is established between the Ui and the Sj at the end of this stage.

Initially, the SC verifies the legitimately of the Ui as follows: (1) requests from the Ui to insert the UIDi, PWi0, PWi1, and SIDj of the Sj that needs to access; (2) computes IDPW = h (UIDi||PWi0||r); (3) computes Ki = Fi ⊕ PWi1; (4) computes XV0i = h2 ((SNi||PWi1) ⊕ (IDPW||Ki)); and (5) checks whether the XV0i matches with the V0i that has stored in the SC by the GWN. If not satisfied, the SC terminates the login request. Otherwise, the verification stage passes successfully and the SC believes the Ui is a legitimate as shown in Fig. 4.

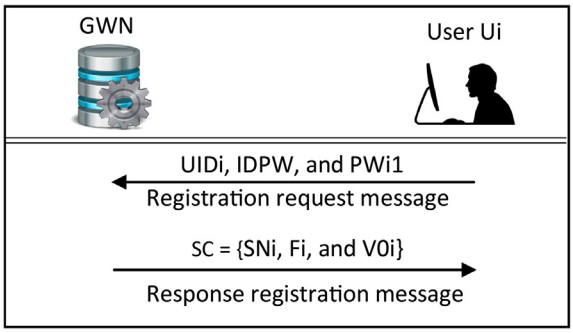

Fig. 3. User registration stage.

After that, to initiate the login request message, the SC executes the following steps: (1) generates random number r2; (2) computes IDi = h2 (UIDi||SNi); (3) derives a new session key SKi = IDi ⊕ Ki; (4) computes CT0 = E_{SKi} (T0||r2||SIDj), where the T0 is a current timestamp; (5) computes V1i = h3 (T0||SKi||SNi||IDi||r2); finally, (6) SC sends the login request message (M1) including {IDi, CT0, V1i} to GWN through unsecure channel.

When the GWN receives the IDi by the login request message from the Ui, the GWN searches in the user's database to get the pair of the IDip and IDis of Ui.

As illustrated in Fig. 5, the GWN has the same cases that have been listed in Ref. [21], these cases can be summarized as the following:

Case 1: (IDi ≠ IDip and IDi ≠ IDis). In this case, the GWN terminates the authentication session and rejects the login request message.

Case 2: (IDi = IDip and IDis ≠ Φ). In this case, the GWN performs the following steps: (1) updates the session number SNi = h5 (SNi); (2) computes a new secret key as Ki = h1 (UIDi||X||SNi), (3) derives SKi = IDi ⊕ Ki; (4) extracts T0||r2||SIDj = DSKi (CT0); (5) checks the value of T0, if it holds, the GWN executes next step, else terminates the authentication session; (6) computes XV1i = h3 (T0||SKi||SNi|| IDi||r2); (7) checks whether the XV1i matches with the V1i that has received from the Ui. If it satisfied, the GWN executes next step; (8) updates the value of suffix and prefix identities together as IDis = IDip, IDip = h2 (r2||UIDi) respectively. Otherwise, the GWN terminates the login request message.

Case 3: (IDi = IDip and IDis = Φ). In this case, the GWN performs the following steps: (1) derives SKi = IDi ⊕ Ki; (2) extracts T0||r2|| SIDj = DSKi (CT0); (3) checks the value of T0, if it holds, the GWN

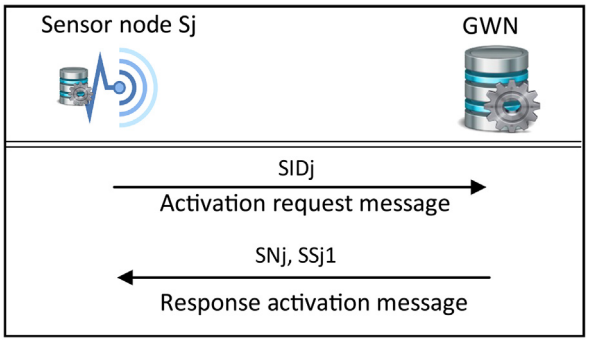

Fig. 2. Sensor node activation stage.

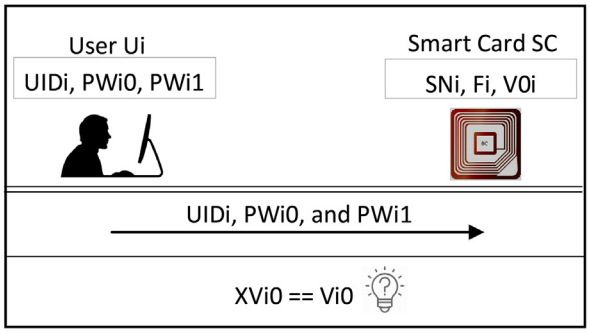

Fig. 4. Authentication (SC – Ui).

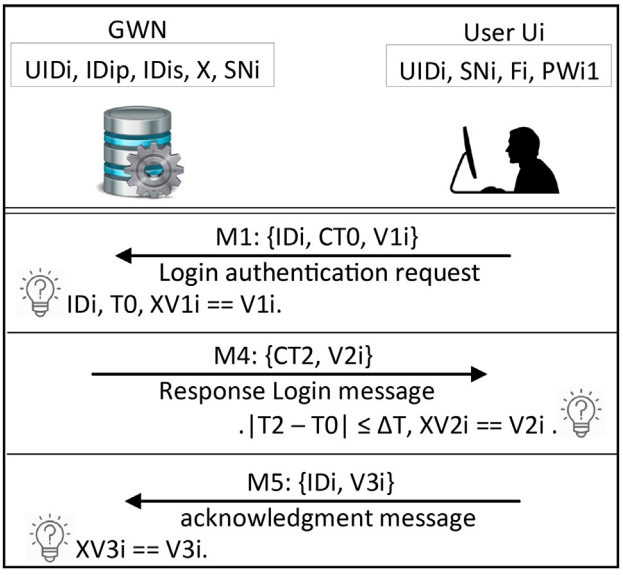

Fig. 5. Authentication (Ui – GWN).

executes next step, else terminates the authentication session; (4) computes XV1i = h3 (T0||SKi||SNi||IDi||r2); (5) checks whether the XV1i matches with the V1i that has received from the Ui. If it satisfied, the GWN executes next step; (6) updates the value of suffix and prefix identities together as IDis = IDip, IDip = h2 (r2||UIDi), respectively. Otherwise, the GWN terminates the login request message.

Case 4: (IDi = IDis). In this case, the GWN performs the following steps: (1) derives SKi = IDi ⊕ Ki; (2) extracts T0||r2||SIDj = DSKi (CT0); (3) checks the value of T0, if it holds, the GWN executes next step, else terminates the authentication session; (4) computes XV1i = h3 (T0||SKi||SNi||IDi||r2); (5) checks whether the XV1i matches with the V1i that has received from the Ui. If it satisfied, the GWN executes next step; (6) updates the prefix identity only as IDip = h2 (r2||UIDi). Otherwise, the GWN terminates the login request message.

Now, after identifying both of the IDi and SIDj, the GWN executes the following steps: (1) computes the V0j = h0 (Kj||SNj); (2) generates randomly r3; (3) creates a new session key SKj = h (SNj||r3); (4) computes CT1 = ((T1 ⊕ V0j)||(SKj ⊕ V0j)||(IDi ⊕ V0j)||(SKi ⊕ V0j)), where the T1 is a current timestamp at the GWN entity; (5) computes V1j = h4 (T1||SKj||SIDj||IDi); (6) updates SNj = h5 (SNj); (7) updates Kj = h2 (SIDj ⊕ SNj); (8) updates IDj = h2 (SIDj||SNj); (9) updates SSj0 = SSj0 + 1; finally, (10) GWN sends the login request message (M2) including {CT1, V1j, and SSj0} to Sj through public channel as shown in Fig. 6.

When the Sj receives the Login request message from the GWN, the Sj executes the following steps: (1) computes the value of ΔSSj as ΔSSj = SSj0 − SSj1; (2) checks the $1 \leq \Delta SSj \leq N$, the value of N is determined according to the application specifications. If not satisfied, terminates the login request. Otherwise, (1) updates (ΔSSj −1) times the value of SNj = h5 (SNj) and Kj = h2 (SIDj ⊕ SNj); (2) computes V0j = h (Kj||SNj); (3) extracts T1||SKj||IDi||SKi = ((CT1|| * 4) ⊕ V0j); and (4) checks the value of T1. If it not satisfied, terminates the authentication session. Otherwise, (1) computes XV1j = h4 (T1||SKj||SIDj||IDi); and (2) checks whether the XV1j matches with the V1j that has received from the GWN. If not satisfied, the Sj terminates the login request. Otherwise, the verification stage is passed successfully and the Sj considers the GWN is a legitimate as well as the Ui who needs to obtain sensor data.

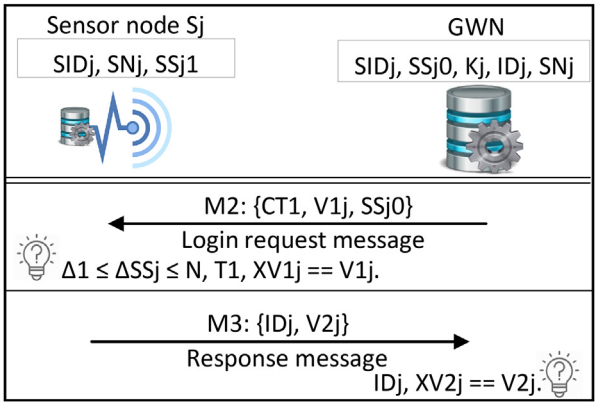

Fig. 6. Authentication (GWN – Sj).

Then, the Sj prepares the response message (M3) and sends it to the GWN according the following steps: (1) sets the initial sequence number M where the value of M is determined according to the application specifications; (2) computes the SKij = SKi ⊕ SKj; (3) stores the Ui record [IDi, SKi, and M]; (4) computes IDj = h2 (SIDj||SNj); (5) updates SSj1 as SSj1 = SSj1 + 1; (6) computes V2j = h4 (SNj||SKij||IDi||IDj); finally, (7) Sj sends the response message M3 including {IDj, and V2j} to GWN through public channel as shown in Fig. 6.

Upon receiving M3 from the Sj, the GWN verifies whether the IDj exists in the sensor nodes database, if not exist, the GWN terminates the login request. Otherwise, the GWN performs the following steps: (1) computes the SKij = SKi ⊕ SKj; (2) computes XV2j = h4 (SNj||SKij||IDi||IDj); (3) checks whether the XV2j matches with the V2j that has received from the Sj. If not satisfied, the GWN terminates the authentication session. Otherwise, the verification stage passes successfully and the GWN considers the Sj is a legitimate sensor node as shown in Fig. 6.

Then, the GWN performs the following steps: (1) computes $CT2 = E_{SKi}$ (SKij||T2||IDip), where the T2 is a current timestamp at GWN entity; (2) computes V2i = h3 (IDi||SKj||r2||SNi||T2); and; (4) sends the response login message (M4) to the Ui that is including {CT2, and V2i} through public channel as shown in Fig. 5.

After receiving the M4 from GWN, the Ui performs the following steps: (1) extracts $SKij||T2||IDip = D_{SKi}$ (CT2); (2) checks the value of ΔT, where $|T2 - T0| \leq \Delta T$, the value of ΔT is determined according to the application specifications. If it holds, the Ui executes next steps, else terminates the authentication session; (3) computes SKj = SKij ⊕ SKi; (4) computes XV2i = h3 (IDi||SKj||r2||SNi||T2); (5) checks whether the XV2i matches with the V2i that has received from the GWN. If not satisfied, the Ui terminates the login request. Otherwise, executes the next steps; (6) V3i = h3 (IDi||SKj||r2||IDip||ΔT); (7) sends the acknowledgment message (M5) to the GWN that is including {IDi, and V3i} through public channel; (8) updates the SNi = h5 (SNi); and (9) updates the IDi = IDip.

Consequently, the verification stage passes successfully and the Ui considers the GWN is a legitimate as well as the Sj. Then, the GWN performs the following steps: (1) computes XV3i = h3 (IDi|| SKj||r2|| IDip||ΔT); (2) checks whether the XV3i matches with the V3i that has received from the Ui. If it not satisfied, terminates the login request. Otherwise, the GWN updates the value of IDis = Φ and considers the Ui is a legitimate as shown in Fig. 5.

4.4. Subsequence authentication stage

In many big data applications, the user is interested to collect the real-time data from the sensor nodes within the same coverage

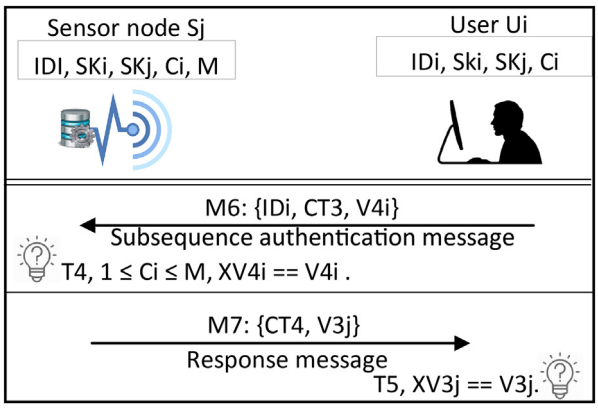

Fig. 7. Subsequence authentication (Ui – Sj).

area as above mentioned. Thus, the Ui should be able to access the data directly from Sj without back to the GWN entity.

In this case, it is important that data are protected from unauthorized access. For this purpose, the subsequence authentication stage is executed between the Ui and Sj after the login authentication stage has been executed between all authentication entities. Fig. 7 illustrates the subsequence authentication stage.

Initially, the Ui performs the following steps to authenticate himself to the Sj: (1) generates random number r4; (2) computes CT3 = (r4||T4||Ci) $\oplus$ h (SKi||SKj||IDi), where the T4 is a current timestamp at Ui entity; (3) computes V4i = h (r4||SKi||T4); and (3) sends the subsequence authentication message {M6} to the Sj that is including the [IDi, CT3, and V4i].

After receiving the M6 from Ui, the Sj performs the following steps: (1) computes r4||T4 = CT3 $\oplus$ h (SKi||SKj||IDi); (2) checks the value of T4, if it hold, the Sj executes next step, else terminates the authentication session; (3) checks if $1 \leq Ci \leq M$, if it hold, the Sj executes next step, else terminates the authentication session and asks the Ui to execute the login authentication stage; (5) computes XV4i = h (r4||SKi||T4); (6) checks whether the XV4i matches with the V4i that has received from the Ui. If not satisfied, the Sj terminates the authentication session. Otherwise, the verification stage passes successfully and the Sj considers the Ui is a legitimate user. Then, the Sj performs the following steps: (1) generates random number r5; (2) computes CT4 = (r5||T5||Ci + 1) $\oplus$ h (SKi||SKj||IDi), where the T5 is a current timestamp at Sj entity; (3) computes V3j = h (r5||SKj||T5); and (3) sends the authentication message {M7} to the Sj that is including the [CT4, and V3j].

Upon receiving the M7 from Sj, the Ui performs the following steps: (1) computes r5||T5 = CT3 $\oplus$ h (SKi||SKj||IDi); (2) checks the value of T5, if it holds, the Sj executes next step, else terminates the authentication session; (3); computes XV3j = h (r5||SKj||T5); (4) checks whether the XV3j matches with the V3j that has received from the Sj. If not satisfied, the Ui terminates the authentication session. Otherwise, updates the Ci = Ci + 1, the verification stage is passed successfully and the Ui considers the Sj is a legitimate sensor node.

4.5. Password change phase

The goal of this phase is to update the password of user securely. In order to reduce the network congestion and to execute this stage in smooth manner, the password of the user is updated without back to the GWN entity. This phase is accomplished between the user Ui and SC entities. when the SC receives the request of password change from the Ui, the SC performs the following Steps: (1) asks the Ui to insert the UIDi, old password PWi0, PWi1, and a new password PW*0i; (2) computes the secret key IDPW = h (UIDi||PWi0||r); (3) computes XV0i = h2 ((SNi||PWi1) $\oplus$ (IDPW||Ki)); (4) checks whether the XV0i matches with the V0i that has stored in the SC. If not hold, the SC rejects the request of password change. Otherwise, (5) computes the IDPW = h (UIDi||PW*0i); (6) computes a new verification code V*0i = h2 ((SNi||PWi1) $\oplus$ (IDPW||Ki)); and (7) updates the verification code V0i = V*0i and then the password change stage passes successfully.

5. Security analysis of the AAA-WSN scheme

While the communication channels between the authentication entities are insecure within unintended environment, an adversary can intercept, delete, capture, modify, and retransmit all authentication messages (request, response, and acknowledgment messages) over these channels. This section illustrates the capabilities of the AAA-WSN scheme to achieve the attractive security features and also to ensure that the AAA-WSN scheme can resist all related popular attacks.

5.1. Mutual authentication

The feature of the mutual authentication means that all authentication entities should authenticate each other to establish a secure communication channel for transmitting the information between them. In AAA-WSN scheme, the GWN represents the trusted entity and the connection bridge between the Ui and Sj entities to authenticate each other.

In order to achieve the mutual authentication feature, the AAA-WSN scheme performs several verification steps during the Login authentication stage as follows: (1) the GWN verifies the authenticity of Ui by checking the login request message {M1} through the values of the T0 and V1i; (2) the Sj verifies the authenticity of GWN by checking the login request message {M2} through the values of the ΔSSj, T1, and V1j; (3) the GWN verifies the authenticity of Sj by checking the response request message {M3} through the value of V2i; (4) the Ui verifies the authenticity of GWN by checking the response message {M4} through the values of the ΔT and V2i; (5) the GWN verifies again the authenticity of Ui by checking the acknowledgment message {M5} through the value of V3i. Furthermore, both of the Ui and Sj authenticate each other in each time the subsequence authentication stage is executed as follows: (1) the Sj verifies the authenticity of Ui by checking the authentication request message {M6} through the values of the Ci, T4, and V4i; (2) the Ui verifies the authenticity of Sj by checking the response authentication message {M7} through the values of the T5, and V3j.

Consequently, the AAA-WSN scheme is able to achieve the mutual authentication feature in all authentication stages.

5.2. User and sensor node anonymity

The user anonymity means two important features: (1) protection of the user identity which means hide the user identity to prevent the unauthorized entity from knowing the user actual identity; and (2) untraceability of the user which means the unauthorized entity can neither distinguish who the user that request to access the sensor node nor the sensor node that the user has accessed. Therefore, unauthorized entity cannot determine if the two authentication sessions are executed by the same user.

The sensor node anonymity is considered an important security service for numerous applications. For example, in healthcare applications, the sensor node identity represents the patient identity. Therefore, if the actual sensor node identity is used within the

communication messages may lead to negative consequences, such as lost the patient his job or his health insurance.

In order to preserve the user and sensor node anonymity in the AAA-WSN scheme, the authentication messages {M1, M2, M3, M4, M5, M6, and M7} are not included the real identity of the user (UIDi) and the real identity of the sensor node (SIDj) as a plaintext whether during the login authentication stage or the subsequence authentication stage.

The AAA-WSN scheme executes a set of one-way hash functions to generate new virtual identities [IDip, IDis, IDi, and IDj] whether instead of the UIDi or SIDj. The values of the virtual identities are changed in each time the login authentication stage is executed. Therefore, it is impossible for unauthorized entity to get or retrieve the real identity from authentication messages that are exchanged between the authentication entities.

Therefore, the AAA-WSN scheme is able to achieve the anonymity feature for both of the user and sensor node entities.

5.3. Perfect forward secrecy

The feature of the perfect forward secrecy is considered one of the most important security requirements to build the schemes based on the AKA principle. The authentication schemes can be support this feature if the disclosing of the long-term keys does not mean an adversary can be able to disclose the previous session keys.

In order to achieve this feature, the AAA-WSN scheme updates the authentication keys by using a set of one-way hash functions for each successful authentication session.

Assume that an adversary has disclosed the long-term keys of the authentication entities [Ki, and Kj], an adversary is still cannot be able to obtain the session keys [SKi, and SKj] of previous session because of the values of SNi, SNj, IDi, and IDj have updated in during the next authentication session as the SNi = h5 (SNi), SNj = h5 (SNj), IDi = IDip = h2 (r2||UIDi), and IDj = h2 (SIDj||SNj), respectively.

Consequently, the AAA-WSN scheme is able to achieve the perfect forward secrecy feature during the login authentication stages.

5.4. Resistance to de-synchronization attack

Most of the proposed authentication schemes that support the anonymity and perfect forward secrecy features always susceptible to de-synchronization attack. These schemes create new identities for the authentication entities to be used within the next authentication session using different methods such as pseudonym identity, cryptographic methods and hash functions, etc. Therefore, the synchronization of these new identities between the authentication entities sides is crucial to successful the execution of their next authentication session. So the adversary always looking to break this synchronization by somehow method to prevent the authorized entities to login ever since.

In order to maintain the synchronization between the Ui and GWN entities, the AAA-WSN scheme employs the following: (1) three identities IDi, IDip, and IDis; (2) two hash functions h2, and h5; and (3) two timestamps T0, and T2. Moreover, for the synchronization between the GWN and Sj, the AAA-WSN scheme employs the following: (1) Sj identity; (2) two serial numbers SSj0, and SSj1; (3) two hash functions h2, and h5; and (4) a timestamp T1.

Suppose the following attack scenarios to illustrate how AAA-WSN scheme maintains the synchronization and resists the de-synchronization attack.

Scenario 1: suppose an adversary prevents the login authentication message {M1} flow. In this scenario, the adversary cannot effect on the synchronization between the Ui and the GWN entities where the values of SNi, and Ki have not even updated.

Scenario 2: suppose an adversary prevents the login authentication message {M2} flow. In this scenario, although an adversary has stopped the current authentication session, but will not be able to effect permanently on the synchronization between the GWN and Sj entities. In the next authentication session, the Sj will update $\Delta SSj - 1$ times the values of SNj as SNj = h5 (SNj), and Kj as Kj = h2 (SIDj $\oplus$ SNj), where the ΔSSj represents the difference between the serial number SSj1 of Sj and the serial number SSj0 of GWN. Thus, the value of IDj that will be computed by the Sj entity will synchronize again with IDj that has stored in GWN entity.

Scenario 3: assume the adversary prevents the response authentication message {M3} flow. In this scenario, although the adversary has stopped the current authentication session, but will not affect completely on the synchronization between the GWN and the Sj entities. The values of SNj, Kj, SS0j, SS1j, and IDj already have been updating in both side.

Scenario 4: assume the adversary prevents the response authentication message {M4} flow. In this scenario, the adversary has stopped the current authentication session, but will not affect permanently on the synchronization between the GWN entity and Ui entity. Since the hash values of SNi have not updated in both of the Ui and GWN entities, only the synchronization of IDip identity is required to consider in next authentication session. Fortunately, when the next login authentication session is launched by the Ui entity using the same IDi of previous session, the GWN entity is still can be able to recognize the IDi of Ui entity through the Value of IDis. In this case the GWN entity needs to computes a new value of IDip as IDip = h2 (r2||UIDi) to continue the authentication session.

Scenario 5: suppose the adversary blocks the response authentication message {M5} flow. In this scenario, the adversary has stopped the current authentication session, but will not affect permanently on the synchronization between the GWN and the Ui entities. Since the IDi and IDip have updated in both of the GWN and Ui entities and the hash value of SNi has updated in Ui side only, the synchronization of hash value SNi is required to consider in next authentication session. When the next login authentication session is launched by the Ui entity using the IDi that has updated in the previous session, the GWN entity is still can be able to recognize the IDi through the Value of IDip. In this case the GWN entity needs to update the session number SNi = h5 (SNi) to continue the authentication session.

In all above scenarios, attacks will be able to make the AAA-WSN scheme unusable temporally but will not have any impact on the next authentication sessions. Therefore, the AAA-WSN scheme is resistant to the de-synchronization attack during all authentication stages.

5.5. Resistance to smart card loss attack

This attack (i.e. smart card loss attack) indicates that an adversary can disclose the actual user identity, and the user password from the smart card using an off-line procedure within polynomial time when the smart card is used illegally.

The Xiong et al. [21] scheme is based on two-factors authentication method, where the scheme is adopted to check the validity of user identity (IDi) and the password (PWi) information. In this scheme, the user Ui chooses the IDi and PWi then generates bi as a random number through the user registration stage. Then the Ui calculates the Ci = h0 (IDi||PWi||bi) to transmit the values of Ui and Ci through a secure channel to GWN. After that, the GWN executes the following steps: (1) generates ui, a, b as random numbers; (2) computes Ki = h1(IDi||x||ui) where x is a secret key of GWN; (3) computes Fi = Ki $\oplus$ Ci; (4) computes V = h2 (h3 (Ki||Ci)); After that the GWN (5) stores values of Fi and V into SC. The

Ui after using the SC that has received from the GWN, Ui stores the value of bi that has generated by himself in the SC.

For Xiong et al. [21] scheme, suppose the following attack scenario to illustrate how such scheme cannot resist the smart card loss attack. If the adversary has obtained the smart card of Ui, and extracts secret information's bi, Fi, and V. Therefore, the adversary can guess the IDi and PWi by checking whether V = h2 (h3 (Fi ⊕ h0 (IDi||PWi||bi))||h0 (IDi||PWi||bi)) holds or not. Since the time that is required to complete all the IDi and PWi space is linear [30]. Therefore, this scheme is still fail to such attack with low entropy.

Since the AAA-WSN scheme is based on three-factors method. If the user SC has stolen by the adversary, and the secret information's have extracted from the SC [SNi, F, V0i, and r], where the Ki = h1 (UIDi||X||SNi), Fi = Ki ⊕ PWi1, IDPW = h (UIDi||PWi0||r), V0i = h2 ((SNi||PWi1) ⊕ (IDPW||Ki)). Consequently, the V0i = h2 ((SNi||PWi1) ⊕ (h (UIDi||PWi0||r)||(Fi ⊕ PWi1))). The adversary cannot guess the password without knowing UIDi and PWi1 together, since the number of candidate passwords is |AUIDi| * |BPWi1| * |CPWi0|, where |AUIDi| is the identity space, |BPWi1| is the space of secret security code, and |CPWi0| is the space of user password. Therefore, AAA-WSN scheme is resistant to the Smart Card Loss Attack.

5.6. Resistance to replay attack

The replay attack means that the adversary attempts to eavesdrop on the authentication messages and retransmit them to the authentication entities without doing any modification. In general, the methods that are used to resist this type of attack: (1) the random numbers; and (2) the current timestamps. The main idea of these methods is to maintain the synchronization between the authentication entities and guarantee the freshness of the authentication messages.

Suppose the following attack scenarios to illustrate how AAA-WSN scheme resists this type of attack.

Scenario 1: if an adversary retransmits the previous eavesdropped message {M1: IDi, CT0, and V1i} to the GWN entity without any modification, where IDi = h2 (SNi||UIDi), SKi = IDi ⊕ Ki, CT0 = ESKi (T0||r2||SIDj), V1i = h3 (T0||SKi||SNi||IDi||r2). So, the GWN entity firstly decrypts the CT0 and checks the value of T0. Therefore, the GWN rejects the login authentication request because T0 is already verified.

Scenario 2: if an adversary retransmits the previous eavesdropped message {M2: CT1, V1j, and SSj0} to the Sj entity without any modification, where CT1 = (T1||SKj||IDi||SKi) ⊕ V0j, V1j = h4 (T1||SKj||SIDj||IDi), and SSj0 is a serial number of the current authentication session that is updated after each successful authentication session as SSj0 = SSj0 + 1. Therefore, the Sj entity will reject the login authentication request because the SSj0 is already verified in previous authentication session.

In first scenario, the AAA-WSN scheme has used timestamp method where validity of T0 is verified before computing any other terms. While in the second scenario, the validity of SSj + 1 are verified before computing any other terms. In addition to, all other authentication messages of login authentication stage are used the challenge-response verification method. Therefore, AAA-WSN scheme is resistant to replay attack.

5.7. Resistance to impersonation attack

The impersonation attack means that an adversary attempts to intercept and forge the request authentication messages to impersonate one of the legal authentication entities.

Suppose the following attack scenarios to illustrate how AAA-WSN scheme resists the impersonation attack.

Scenario 1: suppose an adversary intercepts the login request message {M1: IDi, CT0, and V1i} to impersonate the Ui entity, where IDi = h2 (SNi||UIDi), SKi = IDi ⊕ Ki, CT0 = ESKi (T0||r2||SIDj), V1i = h3 (T0||SKi||SNi ||IDi||r2). It is confirmed that an adversary can generate fresh r2 and timestamp T0. However, an adversary needs to know the Ki, and the current SNi to compute CT0, and V1i.

Scenario 2: suppose the adversary intercepts the login request message {M2: CT1, V1j, and SSj0} to impersonate the GWN entity, where CT1 = (T1||SKj||IDi||SKi) ⊕ V0j, V1j = h4 (T1||SKj||SIDj||IDi), and SSj0 is a serial number of the current authentication session. It is confirmed that an adversary can generate timestamp T1. However, the adversary needs to know the IDi, SKj, SKi, and the current SNi to compute CT1, and V1j.

Scenario 3: Suppose the adversary intercepts the login request message {M3: IDj, and V2j} to impersonate the Sj entity, where IDj = h2 (SIDj||SNj), and V2j = h4 (SNj||SKij||IDi||IDj). However, an adversary needs to know the SKij, IDi, and the current SNj to compute V2j.

In the first scenario, an adversary cannot compute the CT0, and V1i. An adversary cannot compute the CT1, and V1j in second scenario. While in the third scenario, an adversary cannot compute the V2j. So, an adversary not be able to impersonate any of the authentication entities.

Therefore, the AAA-WSN scheme is resistant to impersonation attack.

5.8. Resistance to man-in-the-middle attack

Man-in-the-middle attack means that the adversary attempts to intercept and forge the authentication messages to control the communication messages between authentication entities, then retransmits these messages to make them believe that they are communicate directly with each other.

In the AAA-WSN scheme, all authentication messages that are exchanged between the UI, GWN and Sj entities are protected by a set of secret values such as Ki, SKi, SKj, and SNi. However, the adversary cannot forge the authentication message without knowns these secret values. Therefore, AAA-WSN scheme is resistant to Man-in-the-middle attack.

5.9. Resistance to wrong login information

Practically, wrong login information such as identity, password, and secret security code may be inserted by the user unintentionally into the smart card reader during the login authentication stage. Thus, the authentication scheme may perform unnecessary computations which directly effect on the communication cost and network congestion. Therefore, the detection method of wrong login secret information should be performed at the beginning of login authentication stage, so that the authentication scheme cannot be able to send the login request message without verifying the authenticity of the user. In the AAA-WSN scheme, wrong Login secret information will be detected and rejected by smart card immediately after the verification code V0i that is stored in the smart card is checked at the beginning of login authentication stage. Assume that the user inputs wrong any of secret information (i.e., User identity, password, and secret security code), the values V01 and XV01 will not match. Therefore, AAA-WSN scheme is resistance to Wrong Login information.

5.10. Resistance insider attack

Suppose an insider of the system has obtained the password of the user in somehow, then this person can utilize it to crack other systems that the user has registered with the same identity and password.

In AAA-WSN scheme, the user transmits the {UIDi, IDPW} as registration request message to the GWN entity. An insider has

no ways to get the password of the user, since only the UIDi is stored as a clear text and the IDPW is shielded by the one-way hash function as IDPW = h (UIDi||PWi0||r). Therefore, AAA-WSN scheme is resistance to insider attack.

5.11. Resistance to password Table attack

In AAA-WSN scheme, the GWN entity is not includes any table contains the user's password. Therefore, AAA-WSN scheme is resistance to password table attack.

5.12. Security comparisons

This section presents a comparison between the AAA-WSN scheme with the prior related authentication schemes [19–21] in terms of security features satisfying and the resistance of related popular attacks.

The comparison results that are listed in Table 2 indicate to, the AAA-WSN scheme is the only scheme that can support all security features. By example, the AAA-WSN scheme supports subsequence authentication service, and sensor node anonymity feature, while the other authentication schemes lack of these features. In addition to, both of the Lu et al. [19] and Jung et al. [20] are not be able to support the perfect forward secrecy feature.

Besides, the AAA-WSN scheme is the only scheme that can resist all the related popular attacks, while the other authentication schemes suffer from the smart card Loss attack. In addition to, the Lu et al. [19] and Jung et al. [20] schemes are vulnerable to the de-synchronization attack. Therefore, the AAA-WSN scheme is a fully secure against security attacks and can satisfy all security features.

6. Performance analysis of AAA-WSN scheme

This section discusses different performance analysis features that can be fulfilled by the AAA-WSN scheme, such as a storage space cost, communications cost, and computations cost compared with the prior related authentication schemes [19–21].

Both of the computations cost and communications cost are performed just for the login authentication stage while the storage space cost is performed for the user registration stage, and for the sensor node registration stage.

In order to achieve valid comparisons, the bit length of all authentication parameters will be generalized as follows: (1) identities are 64 bits; (2) passwords are 64 bits; (3) pseudonym identities are 64 bits; (4) current timestamps are 160 bits; (5); sequential and serial numbers are 64 bits; (6) random numbers 256 bits; (7) hash values are 160 bits; (8) the block size of the encryption functions is 128 bits; (9) the plaintext size of the encryption functions is the multiples 128 bits; and (10) the ciphertext size of the decryption functions is the multiples 128 bits.

6.1. Storage space cost analysis

Due to the resource constraints of WSNs, this section concentrates on the storage space cost of the smart card and sensor node. To facilitate analysis, the storage space of the hash functions is not take in account. Table 3 summarizes the storage space cost of smart card and sensor node in the AAA-WSN scheme with the [19–21] authentication schemes.

For the AAA-WSN scheme, the authentication parameters that have stored in the SC {SNi, Fi, V0i, r} require (160 + 160 + 160 + 256) = 736 bits. The authentication parameters that have stored in sensor node {SIDj, SSj1} require (64 + 160 + 64) = 384 bits.

Therefore, the results indicate that the storage cost for smart card in [21] scheme is much higher than other schemes. The smart card storage cost that is required by AAA-WSN scheme is equal to the [19] scheme and much than the [20] scheme. Furthermore, the storage cost for sensor node in the AAA-WSN scheme is equal to the schema in Ref. [21] and higher than other schemes.

6.2. Communications cost analysis

The communications cost is computed based on the size of the authentication messages that are exchanged between entities during the login authentication stage. Table 4 summarizes the communications cost for the AAA-WSN scheme as well as for the authentication schemes of [19–21].

For AAA-WSN scheme, the messages {M1: IDi, CT0, V1i}, {M2: CT1, V1j, SSj0}, {M3: IDj, V2j}, {M4: CT2, V2i}and {M5: IDi, V3i} require (64 (4 8 0) + 160) = (64 + 4 × 128 + 160) = 736 bits, (640 + 160 + 64) = 864 bits, (64 + 160) = 224 bits, ((3 5 2) + 160) = (3 × 128 + 160) = (384 + 160) = 544 bits, (64 + 160) = 224 bits, respectively. Adding the five values, the total communication cost of AAA-WSN scheme is 2592 bits.

The comparison results in term of the communications cost show the following: (1) the number of authentication messages in both of the AAA-WSN and Ref. [21] schemes are five messages, while in the Ref. [19,20] schemes are four messages; (2) the total communications cost that is required for the Ref. [19] scheme is

Table 3
Storage cost analysis.

Schemes	Smart card	Sensor node
[19]	736 bits	224 bits
[20]	576 bits	224 bits
[21]	1088 bits	384 bits
AAA-WSN	736 bits	384 bits

Table 2
Security and functional features comparisons.

Features	[19]	[20]	[21]	AAA-WSN
Mutual Authentication	YES	YES	YES	YES
Subsequence authentication	NO	NO	NO	YES
User Anonymity	YES	YES	YES	YES
Sensor node Anonymity	NO	NO	NO	YES
Perfect Forward secrecy	NO	NO	YES	YES
Resistance de-synchronization Attack	NO	NO	YES	YES
Resistance to Smart Card Loss Attack	NO	NO	NO	YES
Resistance to Replay Attack	YES	YES	YES	YES
Resistance to Impersonation Attack	YES	YES	YES	YES
Resistance to Man-in-the-middle Attack	YES	YES	YES	YES
Resistance to Wrong Login information	YES	YES	YES	YES
Resistance to Insider attack	YES	YES	YES	YES
Resistance to Password table attack	YES	YES	YES	YES

Table 4
Communications cost analysis.

Schemes	M1	M2	M3	M4	M5	Total/bits
[19]	864	800	928	1312	–	3904
[20]	1024	736	320	1056	–	3136
[21]	896	544	224	800	224	2688
AAA-WSN	736	864	224	544	224	2592

Table 5
Computation of authentication entities.

Schemes	Smart Card	GWN	Sensor node
[19]	$7T_h + 2T_{E/D}$	$8T_h + 4T_{E/D}$	$4T_h + 2T_{E/D}$
[20]	$5T_h + 2T_{E/D}$	$5T_h + 2T_{E/D}$	$4T_h$
[21]	$9T_h + 2T_{E/D}$	$11T_h + 2T_{E/D}$	$4T_h$
AAA-WSN	$4T_h + 2T_{E/D}$	$10T_h + 2T_{E/D}$	$4T_h$

Table 6
Total computational cost analysis.

Schemes	Total Functions	Cost/s
[19]	$19T_h + 8T_{E/D}$	≈0.05088
[20]	$14T_h + 4T_{E/D}$	≈0.02688
[21]	$24T_h + 4T_{E/D}$	≈0.03040
AAA-WSN	$18T_h + 4T_{E/D}$	≈0.02744

much higher than other authentication schemes; (3) the AAA-WSN scheme has the lowest communications cost among other authentication schemes.

6.3. Computations cost analysis

This section presents a comparison between the AAA-WSN scheme with the Refs. [19–21] authentication schemes in term of the computations cost analysis. This feature is computed for all authentication schemes according to the operations that performed in each authentication entity.

In order to facilitate the computations cost analysis, the following notations are used: (1) T_h, denotes to the running time; and (2) $T_{E/D}$, denotes to the running time encryption/decryption functions; and (3) the T_h is ≈0.00032 s, and the $T_{E/D}$ is ≈0.0056 s as pointed out in Ref. [21]. Tables 5, and 6 summarize the results of the computations cost analysis for the AAA-WSN scheme as well as for related authentication schemes in Refs. [19–21]. Table 5 shows that the computations cost in each authentication entity of authentication schemes, while Table 6 shows the total computations costs of each authentication scheme.

The results indicate that, the AAA-WSN scheme is relatively has the lowest total computations cost among other authentication schemes. Furthermore, the results indicate to the following: (1) the AAA-WSN scheme has the least computations cost among all schemes at the smart card entity; (2) the computations cost at the sensor node entity in Ref. [19] scheme is much higher than other schemes where the computations cost that is required by AAA-WSN scheme is equal to the computations cost of other authentication schemes; and (3) the computations cost at the GWN entity in Ref. [21] scheme is much higher than the computations cost of other authentication schemes where the computations cost that is required by AAA-WSN scheme is relatively has the least computations cost among other scheme.

7. Conclusion

This paper proposes anonymous access authentication scheme for WSNs in big data environments (AAA-WSN) to achieve appealing security services with high level of efficiency. The AAA-WSN scheme performs strong security features such as the anonymity for the user entity as well for the sensor node entity, full mutual authentication between all authentication entities, and the perfect forward secrecy in all authentication stages. The security analysis shows that the AAA-WSN scheme is resistant to wide range of the popular known attacks, such as the de-synchronization attack, the smart card loss attack, the replay attack, Man-in-the-Middle attack, insider attack, wrong login information attack, password table attack, and impersonate attack. The AAA-WSN scheme uses a set of the hash functions with a lightweight symmetric key cryptography based on the AKA concept to perform all authentication stages. Therefore, the performance analysis in terms of the storage space cost, total computations cost, and total communications cost demonstrates that the proposed scheme achieves high level of security with desirable level of efficiency comparing the recent WSNs authentication schemes. Consequently, the AAA-WSN scheme is applicable to use as authentication scheme in the WSNs as well as is suitable for big data environments that depend on a set of sensor nodes with limited resources during the data capturing and access process.

References

[1] Quan Z, Chunmaing T, Xianghan Z, Chunmaing R. A secure user authentication for sensor network in data capturing. J Cloud Comput 2015;4(6):1–12. 2010.
[2] Farahmandian M, Masdari M, Farahmandian V. Comprehensive analysis of broadcast authentication protocols in wireless sensor networks. J Computer Sci Inf Technol 2014;2(3):107–25.
[3] Giri D, Maitra T, Amin R, Srivastava PD. An efficient and robust RSA-based remote user authentication for telecare medical information systems. J Med Syst 2014;38(145):1–9.
[4] Nashwan S. Secure authentication protocol for NFC mobile payment systems. Int J Comput Sci Network Secur 2017;17(8):256–62.
[5] Nashwan S. SAK-AKA: a secure anonymity key of authentication and key agreement protocol for LTE network. Int Arab J Inf Technol 2017;14(5):790–901.
[6] Nashwan S. SE-H: secure and efficient hash protocol for RFID system. Int J Commun Networks Inf Secur 2017;9(3):358–65.
[7] Yeh H, Chen TH, Liu PC, Kim TH, Wei HW. A secured authentication protocol for wireless sensor networks using elliptic curves cryptography. Sensors 2011;11(5):4767–79.
[8] Choi Y, Lee D, Kim J, Jung J, Nam J, Won D. Security enhanced user authentication protocol for wireless sensor networks using elliptic curves cryptography. Sensors 2014;14:10081–106.
[9] Khan MK, Alghathbar K. Cryptanalysis and security improvements of two-factor user authentication in wireless sensor networks. Sensors 2010;10(3):2450–9.
[10] Das ML. Two-factor user authentication in wireless sensor networks. IEEE Trans Wireless Commun 2009;8(3):1086–90.
[11] Vaidya B, Makrakis D, Mouftah HT. Improved two-factor user authentication in wireless sensor networks. In: Proc, the IEEE 6th Int. conf, wireless and mobile computing, networking and communications (WiMob), Niagara Falls, USA, 11–13 October. p. 600–6.
[12] Nyang D, Lee M. Improvement of Das's two-factor authentication protocol in wireless sensor networks. IACR Cryptol ePrint Arch 2009:631.
[13] Huang H, Chang Y, Liu C. Enhancement of two-factor user authentication in wireless sensor networks. In: Proc, 2010 6th int. conf, intelligent information and multimedia signal processing, Darmstadt, Germany, 15–17 October. p. 27–30.
[14] He D, Gao Y, Chan S, Chen C, Bu J. An enhanced two-factor user authentication scheme in wireless sensor networks. Ad Hoc Sens Wireless Networks 2010;10(3):361–71.
[15] Das A, Sharma P, Chatterjee S, Sing J. A dynamic password-based user authentication scheme for hierarchical wireless sensor networks. J Network Comput Appl 2012;35(5):1646–56.
[16] Turkanovic M, Holbl M. An Improved dynamic password-based user authentication scheme for hierarchical wireless sensor networks. Elektronika Ir Elektrotechnika 2013;19(6):109–15.

[17] Yuan J. An enhanced two-factor user authentication in wireless sensor networks. Telecommun Syst 2014;55(1):105–13.
[18] Amin R, Islam SH, Biswas GP, Khan M, Leng L, Kumar N. Design of an anonymity-preserving three-factor authenticated key exchange protocol for wireless sensor networks. Comput Network 2016;101:42–62.
[19] 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(6):837.
[20] Jung J, Kim J, Choi Y, Won D. An anonymous user authentication and key agreement scheme based on a symmetric cryptosystem in wireless sensor networks. Sensors 2016;16(6):1299.
[21] Xiong L, Peng D, Peng T, Liang H, Liu Z. A lightweight anonymous authentication protocol with perfect forward secrecy for wireless sensor networks. Sensors 2017;17(11):2681.
[22] Wang D, He D, Wang P, Chu C. Anonymous two-factor authentication in distributed systems: certain goals are beyond attainment. IEEE Trans Dependable Secure Comput 2015;12(4):428–42.
[23] Wang D, Wang p. On the anonymity of two-factor authentication schemes for wireless sensor networks: attacks, principle and solutions. Comput Network 2014;73:41–57.
[24] Park D, Boyd C, Moon S. Forward secrecy and its application to future mobile communications security. In: Proc, int. conf. public key cryptography, Melbourne, Australia, 18–20 January. p. 433–45.
[25] Gope P, Hwang T. A realistic lightweight anonymous authentication protocol for securing real-time application data access in wireless sensor networks. IEEE Trans Ind Electron 2016;63(11):7124–32.
[26] Kocher P, Jaffe J, Jun B. Differential power analysis. In: Proc, 19th annual int. cryptology conf on advances in cryptology (CRYPTO '99), Berlin, Germany, 15–19 August. p. 388–97.
[27] Messerges T, Dabbish E, Sloan R. Examining smart-card security under the threat of power analysis attacks. IEEE Trans Comput 2002;51(5):541–52.
[28] Chen, Hua Y, Ge Y, Wang Y, Zeng Z. An improved three-factor user authentication and key agreement scheme for wireless medical sensor networks. IEEE Access 2019;7:85440–51.
[29] Kim B-S, Kim K-I, Shah B, Chow F, Kim KH. Wireless sensor networks for big data systems. Sensors 2019;19(7):1565.
[30] Jiang Q, Zeadally S, Ma J, He D. Lightweight three-factor authentication and key agreement protocol for internet-integrated wireless sensor networks. IEEE Access 2017;5:3376–92.