Composable Secure Roaming Authentication Protocol for Cloud-assisted Body Sensor Networks

Qing-Qing Xie[1], Shun-Rong Jiang[2], Liang-Min Wang[1], and Chin-Chen Chang[3,4]
(Corresponding author: Chin-Chen Chang)

School of Computer Science and Technology, Anhui University[1]
NO. 111, Jiu Long Rd., Hefei, Anhui 230601, China
School of Computer Science and Communication Engineering, Jiangsu University[2]
Zhenjiang, Jiangsu, 212013, China
Department of Information Engineering and Computer Science, Feng Chia University[3]
No. 100, Wenhwa Rd., Seatwen, Taichung, Taiwan 40724 (R.O.C.)
Department of Computer Science and Information Engineering, Asia University[4]
No. 500, Lioufeng Rd., Wufeng, Taichung, Taiwan 41354 (R.O.C.)
(Email: alan3c@gmail.com)
(Received May 23, 2014; revised and accepted Sept. 12 & Nov. 24, 2014)

Abstract

The cloud-assisted Body Sensor Networks (BSN) often has an architecture of Multi-hop Wireless Networks (MWN) model, in which both the body sensors and the users must be secure to protect the whole infrastructure. Unfortunately, both the information providers and the users are movable and resource-constrained in communication and computation. Thus some new security problems are proposed, such as the light weight-secure authentication caused by limited resource, re-authentication in foreign zone caused by mobility, and composablilty security caused by heterogeneity between the transmission subnet, many BSN subnets. We propose a Random Roaming Authentication Protocol (RanRAP) for BSNs with these cloud-assisted infrastructure. We test the composable security at an AP/cluster head/gateway node by using strand spaces theory, and analyze the performance of RanRAP protocol in both the theoretical analysis and experiment simulations. It was shown that RanRAP has some advantages of composable security, computation and communication overheads over some related protocols.

Keywords: Authentication protocol, body sensor networks, composable security, internet of things

1 Introduction

Body Sensor Networks (BSNs) [4, 14] have emerged as a promising technology for medical and non-medical applications, which are also called Wireless Body Area Sensor Networks (WBANs). BSNs consist of a number of miniaturized, portable, and autonomous sensor nodes that are used to monitor the body function and the surrounding environment. These sensor nodes continuously collect vital signs of patients, which are used for ubiquitous health monitoring including real-time diagnosis and prescription. In addition, BSNs may be used for managing catastrophic events and increasing the effectiveness and performance of rescue forces. The huge amount of data collected by WBAN nodes should be saved and preceded in a scalable, on-demand, powerful, and secure manner. Cloud-assisted BSNs are emerged and expected to satisfy these requirements [8]. Typical Cloud-assisted BSN works in the architecture of Multi-hop Wireless Network (MWN) Model [24], [25] as shown in Figure 1, in which a backbone transmission subnet is employed to connect the BSN clusters with cloud.

In Figure 1, the sensor clusters are formed by the body sensors located in a near place. These sensors are weak in computing and communication, but they are movable with the worn person. Thus the sensor can roam from cluster A to cluster B. The transmission subnet is a fixed infrastructure, e.g., the internet, wire networks, and some other steady wireless devices connected to a powerful cloud computing center. Each cluster has an access point (AP) to the backbone network. The AP is powerful in computation and communication, and also serves as the head of the cluster (CH) and the gateway node of the BSN cluster. We take each cluster as a BSN subnet, and the cluster head (CH, also AP) as the base station of the located subnet. Then BSNs with the cloud-assisted

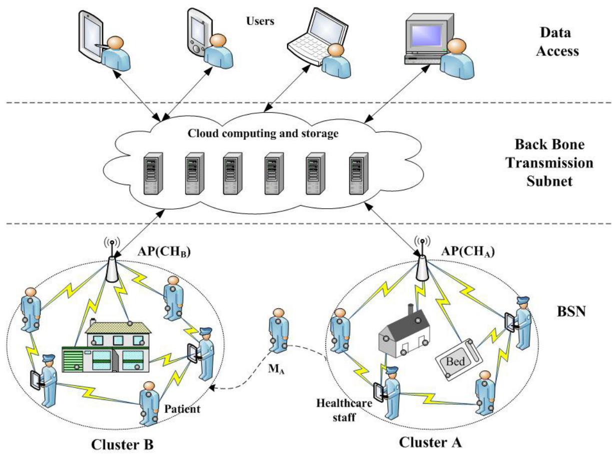

Figure 1: Cloud-assisted BSNs

infrastructure become scalable, for each AP has a cluster and the number of the AP is assumed no limitation in the fixed transmission subnet. The storage and processing of big data collected by BSN isn't a problem, either. Because AP sends the collected information to the cloud computing center through the transmission backbone network, and the cloud will save and process the big data.

Now we discuss the security of cloud-assisted BSN. We won't consider the internal security questions in a BSN cluster, the transmission backbone network and the cloud center with the assumption that they are solved in their areas respectively. We focus on the new security question on the node roaming authentication. For example, node M_A moves away from his home cluster A, and joins a foreign cluster B. M_A cooperates with the node in the cluster B, and the BSN also requires CH_B to collect continually the information of M_A.

In the remainder part of this paper, we will study the roaming identity authentication between CH_B and M_A. This identity authentication has several features. The first is the requirement of the lightweight. The mobile nodes in the BSN are resource-limited in communication, computation and even power supply. Secondly, the selection of the visiting foreign cluster is usually on demand and random. That is to say, a node often moves from one cluster to another in a random manner after the node registers in the initial home cluster. Besides, the node is unwilling to go back to the home cluster to obtain the trusted recommendation certification before joining a new cluster. Finally, when the mobile nodes join in a foreign cluster and obtain their legal identities, they want to access all the network resources of the foreign cluster. Therefore, the authentication protocol should have both the self-security and the composable security [30, 31], and shouldn't affect the security of the existed protocols in the foreign cluster. A typical composable security scenario is that the running of the identity authentication protocol in the cluster head shouldn't reduce the security of other protocols running in that head. Currently, papers about the roaming authentication protocol for this Cloud-assisted BSN architecture are very limited. Up to now, there are no references on the authentication protocol with the random roaming and composable security.

2 Related Work

It seems that the traditional research area of the Secure Wireless Roaming [12, 33, 34, 39] is related to our topic. However, these protocols are realized by the session keys negotiation based on the public key mechanisms. The computation of the public key algorithm costs too much. Thus it is difficult to complete the computation in the node of the BSN. There is also no composable and secure protocol in this area.

The universal composable (UC) security [30, 31] refers to the situation that the protocol guarantees the security when it is in the following cases: running alone, composed of an arbitrary set of protocols, and more generally, used as a component of an arbitrary protocol. Some protocols [1, 3, 19] are designed or analyzed by using the UC formal approach. Unfortunately, the present formal protocol design method for the composable security is combined with a strong security, which fails to guarantee the lightweight property. Later UC security is integrated into the design of roaming authentication protocol, such as [7], which, however, did not attend to the lightweight property.

The typical lightweight authentication protocols in the area of wireless sensor networks is proposed by Perrig et al. [28] who presented the lightweight secure structure SPINS and the broadcast authentication protocol μTESLA. The μTESLA used a reverse hash chain to replace the public-cryptography-based heavy algorithms. Du et al. [6] reduced the computation and communication overheads by adopting the Merkle Tree to construct an authentication path. Further, the whole network was divided into some subareas to reduce the Merkle Tree height, and protocol authentication hops were also reduced. Only the static nodes were considered in [6, 28]. Later many security studies took mobile nodes into consideration, such as the mobile authentication [2, 18, 38], and roaming authentication for wireless communication [11, 23, 36]. But they are not lightweight enough for wireless sensor networks.

The most related work of BSN security is reported in [13, 14, 21, 29]. Huang et al. [14] and Li et al. [21] present a survey on secure access and data security respectively, but they didn't talk about the roaming authentication. References [13, 29] are discussing the lightweight roaming authentication schemes for the wireless sensor networks. Han [13] considered the re-authentication issue on the mobile nodes moving among different sink nodes. The sink in the home cluster is assumed as a trusted third party, and the adjacent relation of the clusters is assumed as the pre-known information. Then the authentication materials are pre-stored in the adjacent clusters. Thus the credible information is also assumed to be transferred to the adjacent clusters. That is to say, the foreign cluster is limited as one of a neighbor of the home cluster. In this way, the communication and computation expenses of the re-authentication are reduced by the neighbor roaming assumption and pre-transferred information. Here this binding relation of neighboring clusters loosed for the cluster heads are connected by the fixed infrastructure of the transmission subnet.

Qiu [29] presented a roaming authentication protocol, in which a mobile node roams within a very large and distributed wireless sensor network, such as the application of the BSN in the healthcare field. When the dynamic sensor moves to a new area (foreign cluster), it sends a request message to the base station before connecting with the router (cluster head) of the area. After verifying the validity of the request message, the base station generates the session key for the mobile node and the router, and sends it to the router. Then the router sends the material of the session key to the mobile node. In Qiu's protocol, the overheads of the base station are too heavy and the communication band near the base station becomes the bottleneck of the system.

We also studied the re-authentication protocol in heterogeneous wireless sensor networks with some mobile sink. In literature [16], we focus on the wireless sensor network based on the classic structure of Voronoi graph, and deduce the computation and storage cost of the presented protocol by using the knowledge of Voronoi graph. In literature [17], we consider a mobile wireless network with a base station, which presented as an on-line trusted authority.

But the scenario of this paper is different from the reported work. At first, there is no base station on-line taking care of the body sensors in networks. Furthermore, the BSN based on a cloud-assisted infrastructure has a MWN model, in which the communication among routers(Cluster Heads) is transferred to the transmission subnet and the computation of the base station was run by the cloud computing center. Thus, the main contribution of this paper is that we focus on a new case that the BSN is connected with a cloud computing center and a backbone transmission network. The presented RanRAP satisfies the random roaming, lightweight and composable security. To the best of our knowledge, our RanRAP is the first reported authentication protocol for the roaming scenario of the presented cloud-assisted BSN.

3 Roaming Authentication Protocol

Our *RanRAP* can be divided into two phases, Phase 1 and Phase 2. In Phase 1, the mobile node registers in the initial home cluster. The secret materials are set and preloaded on the mobile node, such as initial key and authentication information. In Phase 2, the mobile node and the foreign cluster head authenticate each other, and then generate a session key.

Table 1: Notation and description

Notation	Description
t_i	Timestamp
M_A	Mobile node
CH_A	Home cluster
CH_B	Foreign cluster
K_{A_B}	Session key between A and B
$E_K(\cdot)$	Encrypt the plaint message by K
$D_K(\cdot)$	Decrypt the ciphertext by K
$MAC_K(\cdot)$	Message authentication code used K
R_1, R_2	Random number
$H(\cdot)$	hash function
$\|\|$	connect
$\oplus$	xor

In the cloud-assisted BSN, there are three characteristics:

- The nodes are mobile, and they often move from one cluster to another.
- Each cluster has a head, which is the gateway node connected with the BSN cluster and the transmission subnet. The head has the non-limited communication band and is assumed to be secure as the traditional base station.
- Each cluster is like a traditional sensor network. The head has the same assumed abilities as the base station, and all the heads are connected with the transmission subnet and the cloud computing center.

We assume that the cluster has the security structures of the traditional WSN, such as SPINS [28], and the transmission subnet has the public key infrastructure just like the Internet. Here we focus on the authentication scheme for nodes' random cross-cluster roaming. Table 1 shows the notation used in the protocol.

3.1 Phase 1: Mobile Node Initial Registration

In the BSN, the mobile node M_A belongs to the home cluster A with a head CH_A (cluster head A), and registers in this local cluster. In the initial registration phase, M_A sends the registration request to CH_A. Then CH_A randomly selects a symmetric session key $K_{CH_A-M_A}$, a random number r and an identity authentication material $E_{sk_{CH_A}}(CH_A, M_A, t_b, t_e)$, where sk_{CH_A} is the private key of CH_A, t_b and t_e are the predefined beginning and ending time of the identity authentication, respectively. Thus, $t_e - t_b$ is the effective time of the identity authentication. As a reply, CH_A sends $K_{CH_A-M_A}$, r and

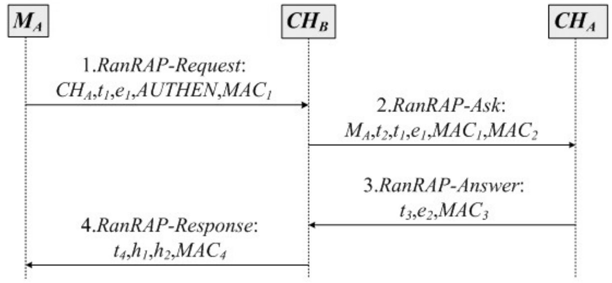

Figure 2: Random roaming authentication protocol

$E_{sk_{CH_A}}(CH_A, M_A, t_b, t_e)$ to M_A. M_A stores the information into its main memory. At the same time, CH_A saves the registration information. The initial registration is finished offline, and assumed to be secure.

3.2 Phase 2: Random Mobile Node Roaming Authentication Protocol (RanRAP)

In Phase 2, M_A moves to a new cluster CH_B, and acquires a legal identity in this foreign cluster. Besides, a new session key is generated between M_A and CH_B. The *RanRAP* protocol is described in Figure 2.

1) *RanRAP-Request phase.* M_A randomly selects a random number $R_1 \in \{0,1\}^a$, computes $e_1 = E_{K_{CH_A-M_A}}(R_1)$, $MAC_{K_{CH_A-M_A}}(CH_A,\ t_1,\ e_1,\ AUTHEN)$ and sends the *RanRAP-Request* message to CH_B,

$$M_A \longrightarrow CH_B : CH_A, t_1, e_1, AUTHEN, MAC_1 \quad (1)$$

where $AUTHEN = E_{sk_{CH_A}}(CH_A,\ M_A,\ t_b,\ t_e)$, $MAC_1 = MAC_{K_{CH_A-M_A}}(CH_A,\ t_1,\ e_1,\ AUTHEN)$, and t_1 is the presented time.

2) *RanRAP-Ask phase.* After receiving the *RanRAP-Request* message at time t^*, the foreign cluster CH_B checks whether $(t^* - t_1) \leq \Delta t$, Δt is a predefined threshold of time slot. If it is in a valid time interval, CH_B uses the public key $K_{PK_{CH_A}}$ of CH_A to decrypt AUTHEN and get CH_A^* and M_A^* in the cipher-text. The accuracy and the reliability of M_A are authenticated and some illegal messages are dropped. We can also obtain t_e which can resist the expired authentication non-limited reused by the adversary.

   After the correctnesses of $CH_A^* = CH_A$ and $M_A^* = M_A$ are verified, CH_B sends the message *RanRAP-Ask* to CH_A and the home cluster of M_A,

$$CH_B \longrightarrow CH_A : M_A, t_2, t_1, e_1, MAC_1, MAC_2 \quad (2)$$

   where t_2 is the message sending time, and $MAC_2 = MAC_{K_{CH_B-CH_A}}(M_A, t_2, t_1, e_1, MAC_1)$.

3) *RanRAP-Answer phase.* After receiving the message *RanRAP-Ask*, CH_A verifies the legitimacy by using the verification algorithm *VerifyA* shown in Figure 3. Therefore CH_A obtains R_1, and sends the *RanRAP-Answer* message to CH_B,

VerifyA algorithm

---

if $((t^*\text{-}t_2) \leqq \triangle t)$
  then compute $MAC_2^* = MAC_{K_{CH_A-CH_B}}(M_A, t_2, t_1, e_1, MAC_1)$;
  if $MAC_2^* = MAC_2$
    then CH_A find $AUTHRN$ and $K_{CH_A-M_A}$ by M_A;
      compute $MAC_1^* = MAC_{K_{CH_A-M_A}}(CH_A, t_1, e_1, ATUEHEN)$;
    if $MAC_1^* = MAC_1$
      then compute $R_1 = D_{K_{CH_A-M_A}}(e_1)$;
    end if
  end if
end if

Figure 3: *VerifyA* algorithm

$$CH_A \longrightarrow CH_B : t_3, e_2, MAC_3 \quad (3)$$

   where $e_2 = E_{K_{CH_A-CH_B}}(r, R_1)$, and $MAC_3 = MAC_{K_{CH_A-CH_B}}(t_3, e_2)$.

4) *RanRAP-Response phase.* When CH_B receives the message *RanRAP-Answer*, the first step is to verify the correctness of t_3 and MAC_3. If the validation fails, the session ends. Otherwise r and R_1 are extracted, and a random number $R_2 \in \{0,1\}^a$ is chosen. The session key is computed according to Equation (4).

$$K_{CH_B-M_A} = H(R_1||R_2) \quad (4)$$

   Finally, CH_B sends the following *RanRAP-Response* message to M_A,

$$CH_B \rightarrow M_A : t_4, h_1, h_2, MAC_4 \quad (5)$$

   where $h_1 = r \oplus H(R_1), h_2 = H(r) \oplus R_2$, and $MAC_4 = MAC_{K_{CH_B-M_A}}(t_4, h_1, h_2)$.

5) M_A receives the *RanRAP-Response* message, and verifies whether t_4 is within the threshold time. If not, the session ends. Otherwise, $r* = h_1 \oplus H(R_1)$ is computed. If $r* = r$, CH_B identity is proved, then M_A computes $R_2 = H(r) \oplus h_2$. Hence M_A can obtain the new session key from Equation (4). Then M_A checks the correctness of MAC_4, if MAC_4 is correct, the authentication completes, and the new session key is valid.

After completing the authentication and generating the session key, CH_B immediately distributes a new identity authentication and r' to M_A, and informs CH_A to delete the identity authentication material r of M_A. Thus M_A becomes a member of CH_B, and can take CH_B as the home cluster and move to another new foreign cluster.

Taking into account the issues of traceability and tracking, when M_A joins CH_B, CH_B redistributes a new ID to the mobile node. We assume that each cluster head has 2^{16} IDs. When the mobile node obtains the trust of the new cluster, the cluster head selects a new unused ID for the mobile node. In this way, we can prevent the outside nodes to track the trajectory of the mobile node. At the same time, in order to let the lawful authority trace the movement of the mobile node, each cluster head maintains a source ID table which is like Table 2. The table includes the ID of the previous home cluster head, the ID of the mobile node in previous home cluster, the redistribution ID of the mobile node in the new cluster, and the time taken to join the cluster.

3.3 Protocol Security Analysis

Proposition 1. RanRAP *satisfies the forward security.*

Based on the Table 2, even if the attackers acquire the session key $K_{CH_C-M_A}$ between the mobile node M_A and the cluster node CH_C, it is still difficult to derive the session key used before, i.e. $K_{CH_B-M_A}$ .The session key between M_A and CH_B is determined by two random numbers R_1 and R_2, and they are separately transmitted by the ciphertext e_1 in Equation (1) and the XOR value h_2 in Equation (5).

If the attackers want to obtain the plaintext R_1 from e_1, they must know the session key $K_{CH_C-M_A}$ between M_A and CH_A. However $K_{CH_C-M_A}$ is a preloaded value and is assumed to be completely secure. Thus it is impossible to obtain the value of R_1 in our *RanRAP*.

R_2 is also difficult to know because it is only used in $h_2 = H(r) \oplus R_2$. If the attackers want to deduce R_2 from h_2, they should know the hash value $H(r)$. However r is also a preloaded value and it is as secure as $K_{CH_C-M_A}$.

Even if the attackers acquire the current session key of M_A, they can not derive the previous session key of M_A without R_1 and R_2. Thus the protocol satisfies the forward security.

The forward security also provides a privacy protection for the roaming node. When the roaming node joins the new cluster, other nodes and the physical capture attackers do not know which cluster the roaming node comes from. However the cluster head that acts as the AP of the cluster knows the privacy, thus the roaming node can also be traced by the authorized assistance of the AP.

Proposition 2. RanRAP *obtains the local identity authentication.*

In the roaming protocol *RanRAP*, there is no pre-shared information between CH_B and M_A. However, the *RanRAP-Request* message in Equation (1) contains the identity authentication $AUTHEN$ which is encrypted by the private key of CH_A. After CH_B receives $AUTHEN$, CHB decrypts the ciphertext by using the public key of $K_{pk_{CH_A}}$ and obtains CH_A* and M_A*. If $CH_A* \neq CH_A$ and $M_A* \neq M_A$, the mobile node is judged to be illegitimate. The *RanRAP-Ask* message is not sent to CH_A in the BSN. In this way, the performance of the resistance against the forgery attack can be improved. CH_B can also acquire t_e, to resist the non-limitation of reusing the expired identity.

With the support of the MWN-based architecture, we assume that the authentication materials are securely transmitted by the transmission subnet. In the BSN, the authentication protocol serves as the local authentication schemes between the mobile node and the foreign header. That is to say, the MWN-based IoT architecture makes all the heads like the neighbors, which saves the computation and communication over the BSN.

Proposition 3. RanRAP *completes the mutual identity authentication.*

In the *RanRAP* protocol, M_A applies to join a new cluster by sending the *RanRAP-Request.* This message contains the authentication content $AUTHEN$. According to Proposition 2, $AUTHEN$ can achieve the initial identity authentication of M_A in the cluster CH_B. The completed identify authentication is realized by the algorithm *VerifyA* after CH_A receives the message *RanRAP-Ask* in Equation (2).

As CH_B shows his identity to M_A, it is mainly deterred by the random value r, which exists in the *RanRAP-Response* message. The random value is delivered through the XOR value $h_1.r$ is generated in the home cluster and is re-generated after the authentication is realized in the foreign cluster.

Proposition 4. RanRAP *has the ability of preventing man-in-the-middle attacks.*

According to the analysis of the *RanRAP* protocol, we find that the attackers are able to trick or truncate *RanRAP-Request* messages to imitate the mobile node M_A and communicate with the foreign cluster head. Thus the attackers can preserve the protocol and eventually be able to extract the session key material from the feedback message *RanRAP-Response.* However, according to Proposition 1, R_1 and R_2 are not sent in the plaintext. In order to attack the protocol, the previous session key should be known. The whole problem is back to the question of Proposition 1. For the intermediary tampering attacks, as mentioned in Proposition 3, the bilateral identity authentication ensures the correctness of the identity of the message sender, and the MAC used in every message ensures the message integrity. The security of the MAC depends on the security of the hash function. The recommended MAC size in our protocol is 4 bytes for the practical application, since only 40 forgery attempts per second are available in a 19.2 kb/s channel and 2^{31} trials are required for a successful forgery. The intermediary can not construct a valid message to realize the communication, thus the protocol is secure against the man-in-the-middle attacks.

Proposition 5. RanRAP *blocks the replay attack.*

The current timestamp is bound in every message of *RanRAP*. They are noted as t_1, t_2, t_3 and t_4 in Request, Ask, Answer and Response messages, respectively. If the received messages is not in the valid time slots Δt, it will

Table 2: Source ID table in the cluster head

Mobile Node	Source Cluster Head	Local ID	Time
M_A	CH_A	M'_A	t_a
M_B	CH_C	M'_B	t_b
...	...	...	...
M_I	CH_J	M'_I	t_k

be dropped to resist the replay attacks. The random number used to generate the pair-wise key is updated when the mobile node joins the new cluster according to Equation (5). Thus, the freshness and prevention of the replay attack are guaranteed validly.

3.4 Discussion about the DoS Exhaustion Attacks

The Denial of Service (DoS) attack is a key issue that must be considered in the design of the security network protocols. References [15] and [27] reported that the DoS attack can be efficiently prevented if the authentication is completed by the mobile node and the foreign head. That is to say, local roaming authentication at foreign head is beneficial for the DoS prevention.

Our *RanRAP* designs for the BSN based on the cloud-assisted structure (Figure 1). We assume the security questions on the transmission subnet are solved, and do not discuss the security question within a single BSN cluster with the assumption that is the same as traditional wireless sensor networks. Thus we consider the DoS attack on the *RanRAP* protocol, which is different from the DoS attack in WSNs discussed in [15, 27, 32]. The DoS attacks that we will study in the roaming authentication scenario can be classified into the following two aspects.

1) Attack from inside adversary. The inside attacks are launched by inside nodes. If the mobile node is physically captured by the adversary, it is compromised and replicated. Then a large number of replicas are deployed in the BSN, and the adversary can launch the DoS attack. Due to the fact that the multiple replicas have the same ID in the cluster, the cluster head is easy to find the replica by binding the sensor's relative location and ID. The replica detection is another research area and some papers have reported good results [35, 41]. When the replicas are deployed in different clusters, they are difficult to be detected by the ID recognition. However, in our *RanRAP*, the mobile node has a pair of keys with the cluster head. Only one replica is allowed in a cluster. Thus it is impossible for a simple replicated node to launch the DoS attack in a subnet. Thus the inside DoS attack is resisted by this means.

2) Attack from outside adversary. This kind of DoS attack often depletes the network resources by replaying the forged or overdue packets. *RanRAP* resists this attack by encrypting and authenticating the fresh message with a timestamp. Unfortunately, the cryptology algorithm can recognize the outside attacker, but can't fight against the resources depletion in communication and computation. The attackers can also cheat the sensor node by ceaselessly sending the request message to the header and asking for joining the cluster. Then the relay nodes forward a large amount of packets to the cluster head. The head runs *RanRAP* to authenticate the request.

Our *RanRAP* protocol can't prevent the outside DoS attack, because the sensor node directly sends the *RanRAP-Request* message to the cluster head. Because of the characteristic of the random roaming, the sensor node doesn't know any information about the mobile node when it joins the cluster. Thus it's difficult to authenticate the mobile node. The outside DoS attack is an open problem in this area, and Qiu's [29] and Han's [13] papers didn't consider the energy overhead caused by this attack.

In our cloud-assisted BSNs, *RanRAP* can be improved by dividing each BSN cluster into some small sub-clusters. This method was enlightened by the scalable and clustering scheme presented by Reference [20]. A sub-cluster can vote a chair by some cluster selection algorithms. When the mobile node M_A moves into the cluster CH_B, M_A first communicates with the closest sub-cluster chair CH_{B_i}. The validity of message *RanRAP-Request* is checked by CH_{B_i} as the first step. After check, the chair decides whether the request will be forwarded. In this improvement, we shift the verification process from CH_B to small cluster head CH_{B_i}. Then the bad consequence of the DoS attack is limited in a small sub-cluster.

4 Composable Security Based on Authentication Test

The authentication test was proposed by Guttam [5, 10]. The authentication test is based on the security protocol formalization of the strand spaces theory and the challenge-response mechanism. The instance of authentication test is constructed by a special form of the data transmission characterized with the unique source property. The special form of data transmission completes the proof of the authentication properties of the protocol through proving the existence of the general nodes. The composable authentication test was also proposed by Guttman [9] in 2009, and is used to prove that two protocols used in composition don't undermine the overall security.

4.1 Basic Framework of Composable Protocol's Authentication Test

The basic goal of the composable authentication test is to test whether Π_2 has a composable security. We consider the composition of protocols Π_1 and Π_2 (the composable protocol is denoted by Π_1). If the composable protocol Π is still able to achieve the security goals identified by II_1, it means the operation of Π_2 doesn't affect the security goals identified by Π_1. Thus Π_2 has a composable security based on the authentication test.

When the proposed protocol *RanRAP* runs, the cluster heads CH_A and CH_B have a shared key $K_{CH_A-CH_B}$. It can be assumed that $K_{CH_A-CH_B}$ is generated by the classic protocol TinyPK. Eventually, there are some circumstances of the composable using of TinyPK [37] and the *RanRAP* protocol. We record TinyPK as Π_1 and *RanRAP* as Π_2. Π means that TinyPK is used in the combination with the *RanRAP* protocol, which is used to test whether the *RanRAP* affects the security goals of TinyPK during the running process of Π. In other words, if Π achieves authentication test, then Π_1 is composable secure in this application instance.

The proof of the composable authentication test is generally executed in three steps [9]. First, the strand space directional figure is used to describe the initial protocol. It simplifies the running process of the protocol. And predicate symbol is also used in this protocol. Second, the security goal of Π_1 is deduced on the basis of the protocol logical description, and it is proved that the security goal of the composable protocol Π and Π_1 is homomorphic. At last, the node roles involved during the protocol running on the basis of the protocol logical description are defined and described. By decomposing the node role (Π_1 or Π_2) during the protocol execution of Π, the proof about the strong disjoint encryption of Π_1 and Π. the solution of counterexample of Π-skeleton, and the realization of Π-skeleton are completed. After the three steps, the security goals of Π_1 can be achieved, which means that the composable security authentication test of Π_2 is completed. The three steps are separately described in Subsections 4.2, 4.3, and 4.4.

4.2 Test Strand Space Model and Description of Protocol

As described in Subsection 4.1 the composable security of *RanRAP*(Π_2) is tested by the composable use states of TinyPK(Π_1) and *RanRAP*(Π_2) among nodes M_A, CH_A and CH_B. Thus during the composable protocol's execution, we can see whether the security goals of TinyPK(Π_1) can be realized. The strand space model is used here to simplify the two protocols, and the logical language is used to describe the protocols.

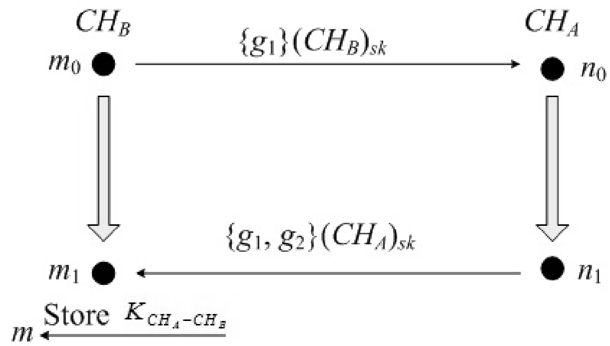

Figure 4: Strand space of TinyPk(Π_1)

The function of TinyPK [37] in the composable protocol is to generate the shared key among clusters. The strand space of TinyPk(Π_1) is shown in Figure 4.The cluster heads CH_B and CH_A are two participants. CH_B, m_0, and m_1 are the nodes of CH_B. n_0 and n_1 are the nods of CH_A. g_1 and g_2 are generated by the Diffie-Hellman key exchange. g_1 is g^x mod P, and g_2 is g^y mod P (x and y are random values). $\{g_1\}(CH_A)_{sk}$ represents that CH_A uses the private key of CH_A to encrypt g_1. Store $K_{CH_A-CH_B}$ means that after CH_B verifies the correctness of the messages sent by CH_A, CH_B computes the shared key $K_{CH_A-CH_B}$ and stores the key. For simplicity, some unnecessary parameters are neglected during the implementation of the protocol. The basic security assumptions of protocol TinyPK(Π_1) are as follows: (1) $(CH_B)_{sk}$, $(CH_A)_{sk} \notin K_P$ (K_P is the key set grasped by the penetrator), (2) x and y are generated uniquely, (3) g is not leaked, and (4) $x \neq y$.

As shown in Figure 4, the strand spaces of protocol Π_1 contain the initiator strands and the responder strands.

The initiator strands of protocol TinyPk(Π_1) are denoted by S_{i1}.which contains two participants CH_B and CH_A. Messages $\{\{g_1\}(CH_B)_{sk}\}$ and $\{\{g_1, g_2\}(CH_A)_{sk}\}$ are transmitted between them. We use $Init[]$ as the identity of the initiator strands

$$S_{i1} \in Init[CH_B, CH_A, \{g_1\}(CH_B)_{sk}, \{g_1, g_2\}(CH_A)_{sk}].$$

The trace is described as Equation (6)

$$\begin{aligned} & << \sigma_1, a_1 >, < \sigma_2, a_2 >> \\ = \; & < +\{\{g_1\}(CH_B)_{sk}\}, -\{\{g_1, g_2\}(CH_A)_{sk}\} > \end{aligned} \quad (6)$$

In Equation (6), $< \sigma_1, a_1 >$ represents the symbolic term of the trace. The symbol term generally are denoted by $< \sigma, a >$, where σ has the positive or negative values, corresponding to the sender or receiver, respectively. a is the strand space trajectory, which is the message transmission path. For example, the trace of the initiator $< \sigma_1, a_1 >$ corresponding to $+\{\{g_1\}(CH_B)_{sk}\}$, means that CH_B sends a message $\{g_1\}(CH_B)_{sk}$.

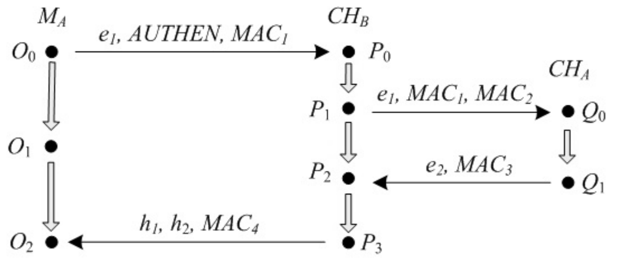

Figure 5: Strand space of TinyPk(Π_2)

$\forall$ $m, x, y, g, (CH_A)_{pk}, (CH_B)_{pk}, (CH_B)_{sk}, (CH_A)_{sk}$
if Store1($m, (CH_A, CH_B)_K$)
$\wedge$Non(pk(CH_A)) $\wedge$Non(pk(CH_B)) $\wedge$Non(g) (φ1)
$\wedge$Non(sk(CH_A)) $\wedge$Non(sk(CH_B)) $\wedge$ Non(g) (φ2)
then $\exists$ m_0, n_0
$CH_B1(m_0, x, g_1, (CH_B)_{sk})$ $\wedge CH_A1(n_0, g_1, (CH_B)_{pk})$
and $\exists$ m_1, n_0
$CH_A2(n_1, g_1, y, g_2, (CH_B)_{pk}, (CH_A)_{sk})$ $\wedge$ $CH_B2(m_1, x, g_1, g_2, (CH_B)_{sk}, (CH_A)_{pk})$
$\forall$ $m, n, x, y, g, (CH_A)_{pk}, (CH_B)_{pk}, (CH_B)_{sk}, (CH_A)_{sk}$
if $\varphi 1 \wedge \varphi 2 \wedge$ Lsn1(n, g)
then false

Figure 6: Implementation of Protocol TinyPK(Π_1)

The responder strands of protocol TinyPk(Π_1) are denoted by S_{r1} which contains the same strands as the initiator strands. We used *Resp*[] as the identity of the responder strands, then

$$\begin{aligned} & << \sigma_1, a_1 >, < \sigma_2, a_2 >> \\ = \; & < -\{\{g_1\}(CH_B)_{sk}\}, +\{\{g_1, g_2\}(CH_A)_{sk}\} > \end{aligned} \tag{7}$$

$< \sigma_1, a_1 >$ corresponds to $-\{\{g_1\}(CH_B)_{sk}\}$ in Equation (7), means that CH_A receives a message $\{g_1\}(CH_B)_{sk}$. $< \sigma_2, a_2 >$ in Equation (7) corresponds to $+\{\{g_1, g_2\}(CH_B)_{sk}\}$, and means that CH_A sends a message $\{g_1, g_2\}(CH_A)_{sk}$.

The function of protocol *RanRAP* in the composable protocol is to achieve the mobile node authentication accessing to the new foreign cluster by using the shared key generated by protocol TinyPK. The strand spaces are shown in Figure 5, where O_0, O_1 and O_2 are the nodes of participant M_A, P_0, P_1. P_2 and P_3 are the nodes of participant CH_B. Q_0 and Q_1 are the nodes of participant CH_A. The symbols involved in strand space have the same definition as described in Subsection 3.2. The basic security assumptions of protocol *RanRAP(*Π_2*)* are as follows: (1) $(M_A, CH_A)_k$, $(CH_A, CH_B)_k \notin K_P$, (2) R_1 and R_2 are generated uniquly, and (3) $R_1 \neq R_2$. $(M_A, CH_A)_k$ denotes the session key between M_A and CH_A, and K_P is the key set grasped by the penetrator.

In Figure 5, the basic strand spaces of protocol Π_2 contain the initiator strands, the responder strands, and the server strands.

The initiator strand of protocol *RanRAP(*Π_2*)* is denoted by S_{i2}, which contains two participants, M_A and CH_B. The messages transmitted between them are $e_1, AUTHEN, MAC_1$ and h_1, h_2, MAC_4. We use *Init*[] as the identity of the initiator strands.

$$\begin{aligned} S_{i2} \in Init[&M_A, CH_B, e_1, AUTHEN, MAC_1, \\ &h_1, h_2, MAC_4]. \end{aligned}$$

The trace is described as Equation (8)

$$\begin{aligned} & << \sigma_1, a_1 >, < \sigma_2, a_2 >> \\ = \; & < +\{e_1, AUTHEN, MAC_1\}, -\{h_1, h_2, MAC_4\} > \end{aligned} \tag{8}$$

The responder strands of protocol *RanRAP*(Π_2) is denoted by S_{r2}, which contains three participants, M_A, CH_B and CH_A. The messages transmitted among them are $\{e_1, AUTHEN, MAC_1\}$, $\{e_1, MAC_1, MAC_2\}$, $\{e_2, MAC_3\}$ and $\{h_1, h_2, MAC_4\}$. We use *Resp*[] as the identity of the responder strands, then

$$\begin{aligned} S_{r2} \in Resp[&M_A, CH_B, CH_A, e_1, AUTHEN, MAC_1, \\ &MAC_2, e_2, MAC_3, h_1, h_2, MAC_4]. \end{aligned}$$

The trace is described as Equation (9)

$$\begin{aligned} & << \sigma_1, a_1 >, < \sigma_2, a_2 >, < \sigma_3, a_3 >, < \sigma_4, a_4 >> \\ = \; & < -\{e_1, AUTHEN, MAC_1\}, \\ & \quad +\{e_1, MAC_1, MAC_2\}, -\{e_2, MAC_3\}, \\ & \quad +\{h_1, h_2, MAC_4\} > \end{aligned} \tag{9}$$

The server strands of protocol *RanRAP*(Π_2) is denoted by S_{s2}, which contains two participants, CH_A and CH_B. The messages transmitted between them are $\{e_1, MAC_1, MAC_2\}$ and $\{e_2, MAC_3\}$. We use *Ser*[] as the identity of the server strands.

$$S_{s2} \in S_{er}[CH_A, CH_B, e_1, MAC_1, MAC_2, e_2, MAC_3].$$

The trace is described as Equation (10),

$$\begin{aligned} & << \sigma_1, a_1 >, < \sigma_2, a_2 >> \\ = \; & < -\{e_1, MAC_1, MAC_2\}, +\{e_2, MAC_3\} > \end{aligned} \tag{10}$$

4.3 Security Goals Description of Protocol Π_1 and Homomorphism Security Goals Proof of Π

Guttman [9] defined a goal language $L(\Pi)$ based on the classical first order logic for the strand space security protocol. $L(\Pi)$ is a language for the execution of protocol Π, based on Classic Quantified Language. We use $L(\Pi)$ to describe the execution of protocol Π and define the security goals of the protocol. Figure 6 shows the implementation of protocol TinyPK(Π_1) based on the language grammar, in which the strand space is described in Figure 4. φ_1 and φ_2 are extracted as the security goals.

In Figure 6, m is the storage node where CH_B stores $K_{CH_A - CH_B}$. m_0, m_1, and n_0 correspond to the nodes as shown in Figure 3, which send or receive a message. n is defined as the attackers' monitoring node which is not drawn out in Figure 4. $x, y, g, (CH_A)_{pk}, (CH_B)_{pk}, (CH_B)_{sk}, (CH_A)_{sk}$ and other notations in Figure 6 have the same definitions as Subsection 4.2 and Figure 4.

Figure 6 also includes some new predicate symbols, such as Store1 $(m, (CH_A, CH_B)_K)$ and Non($pk(CH_A)$), which are defined as follows.

In the predicate symbols like Nov(v), v is assumed to be no-originating. It exists in every entity before the implementation of the protocol and is not grasped by the penetrator. Be specific to Non($pk(CH_A)$), Non($pk(CH_B)$), Non($sk(CH_A)$) and Non($sk(CH_B)$) in Figure 6, where $(sk(CH_B))$ is defined as the private key of CH_B, and $(pk(CH_B))$ is the public key of CH_B.

Predicate symbols like RhoJ$(m, v_1, \cdots, v_i)$ are the role predicate. They are defined as follows, In skeleton A when m is the j^{th} node of an instance of the role ρ, with its parameters (in some conventional order) instantiated by the associated values $v_1, \cdots, v_i$. Be specific to Store1$(m, (CH_A, CH_B)_K)$, $CH_B1(m_0,\ x,\ g_1,\ (CH_B)_{sk})$, $CH_A1(n_0,\ g_1,\ (CH_B)_{pk})$, $CH_{A2}(n_1,\ g_1,\ y,\ g_2,\ (CH_B)_{pk},\ (CH_A)_{sk})$, $CH_{B2}(m_1,\ x,\ g_1,\ g_2,\ (CH_B)_{sk},\ (CH_A)_{pk})$ in Figure 6, where Store1$(m,\ (CH_A, CH_B)_K)$ means that Store role stores $(CH_A, CH_B)_K$ in node m. $CH_B1(m_0,\ x,\ g_1,\ (CH_B)_{sk})$ means that CH_B produces variable x, g_1, $(CH_B)_{sk}$ at its first node m_0. The same explanation can be used in the other role predicates.

The security goals based on predicates φ_1 and φ_2 can be explained as follows. Before $K_{CH_A-CH_B}$ is generated, CH_A receives the message $\{g_1\}(CH_B)_{sk}$ and successfully obtains g_1. CH_B receives the message $\{g_1, g_2\}(CH_A)_{sk}$ and successfully obtains g_2. Security assumptions: $(CH_B)_{sk}, (CH_A)_{sk}$ and g are not leaked.

The mapping relation is based on the strand space and classic quantity language. We can verify the security goals claimed by φ_1 and φ_2 in Figure 6 and the security goals in Figure 4 are homomorphic.

Theorem 1. ***The security goals based on strand space model in Figure 4 can be expressed as*** φ_1 ***and*** φ_2.

Proof. Suppose A as the skeleton of Π. σ is the function mapping of skeleton A from variables of Π_1 to the strand space of Π. To deduce $A \models \varphi(\varphi_1 and \varphi_2)$, all the function mapping of A should satisfy $A,\ \sigma \models \varphi$.

The propositional logic is defined via the standard Tarski inductive clauses for the classical first order logical constants, and the basic clauses are as follows:

$A, \sigma \models Nov(v)$, iff $\sigma(v) \in non_A$ (non_A means passive, and it exists in the role node before the implementation of the protocol);

$A, \sigma \models RhoJ(m, v_1, \cdots, v_k)$, iff $\sigma(m) \in nodes(A)$ (nodes(A) represents the nodes belonging to skeleton A), and $\sigma(m)$ is the j^{th} node of the role ρ with its parameters (in some conventional order) instantiated by the associated values $\sigma(v_1), \cdots, \sigma(v_k)$.

The predicates $pk(CH_A)$, $pk(CH_B)$, $sk(CH_A)$, and $sk(CH_B)$ in Figure 6 are no-originating in skeleton A. All the four predicates meet $\sigma(v) \in non_A$, and then $A, \sigma \models Nov(v)$.

Same as the role nodes in Figure 6, Store1$(m,\ (CH_A, CH_B)_K)$, $CH_B1(m_0,\ x,\ g_1,\ (CH_B)_{sk})$, $CH_A1(n_0,\ g_1,\ (CH_B)_{pk})$, $CH_{A2}(n_1,\ g_1,\ y,\ g_2,\ (CH_B)_{pk},\ (CH_A)_{sk})$, $CH_{B2}(m_1,\ x,\ g_1,\ g_2,\ (CH_B)_{sk},\ (CH_A)_{pk})$ are the role nodes and meets the parameter relationship, which satisfies $A, \sigma \models RhoJ(m, v_1, \cdots, v_k)$.

We define $A \models \varphi$ when $A, \sigma \models \varphi$ for all σ. Theorem 1 verifies that the security goals based on the strand space model in Figure 4 can be expressed as φ_1 and φ_2. □

4.4 Composable Security Proof of the RanRAP Protocol

According to the definition of the composable security protocol in [9], the realization is divided into the following steps. First, it is proved that Π and TinyPK(Π_1) have a strong disjoint encryption, which is defined here as Proposition 6. Second, we give the solution to the counterexample of Π_1-skeleton of Π and the realization of Π-skeleton as Proposition 7. Finally, after drawing the above two conclusions, with the composable security definition described in [9], Theorem 2 is concluded, which means that RanRAP is a composable security.

Proposition 6. *Protocol* Π *and TinyPK(*Π_1*) satisfy the Strong Disjoint Encryption.*

The strong disjoint encryption requires that when *RanRAP*(Π_2) constructs the protocol, there should be no creation conflicts and extraction conflicts with TinyPK(Π_1). The creation conflicts mean that *RanRAP*(Π_2) can not create encryptions which are specified in TinyPK(Π_1). *RanRAP*(Π_2) can be used, but can not construct a similar encryption, which can leak the contents constructed by TinyPK(Π_1). The extraction conflicts mean that the encrypted content of TinyPK(Π_1) are transmitted to the *RanRAP*(Π_2) protocol, which can not be re-transmitted the plaintext outside these encrypted contents again.

Proof. According to the definition of the strong disjoint encryption in [9], the proof is divided into three steps. First, the primary and secondary nodes of the combination protocol Π should be found out. Second, the message related to the creation conflicts and extraction conflicts should be found out, which is based on the definition of the secondary nodes. Finally, combined with the secondary node, the content encrypted by the creation conflicts and extraction conflicts should be found out, and the conclusion should be drawn. □

1) Determining the primary and secondary nodes. The primary nodes are defined as the role nodes, appeared in TinyPK(Π_1) when using the composable protocol Π. The role nodes are defined as the secondary nodes, which are used in protocol Π but not in an instance of the role nodes of TinyPK(Π_1). According to the definition of the role nodes and traces in Equations (6) and (9), it can be found that CH_A and CH_B are the primary nodes, and M_A is the secondary node.

2) Determining the contents of the creation conflicts and extraction conflicts about the secondary node. The main purpose of this step is to identify all the involved encrypted and decrypted contents of the secondary node, which is prepared for the creation conflicts and extraction conflicts of the next step. According to trace in Equation (8), the encrypted contents are e_1, $MAC_1$and $AUTHEN$, and

$$e_1 \in E \subseteq (\Pi_2) = \{e_1 : \exists O_0, \alpha . e_1 \subseteq msg\,(\alpha(O_0)) \wedge (O_0) \text{ is a role node of } \Pi_2\}, \tag{11}$$

$$MAC_1 \in E \subseteq (\Pi_2) = \{MAC_1 : \exists O_0, \alpha . MAC_1 \subseteq msg\,(\alpha(O_0)) \wedge (O_0) \text{ is a role node of } \Pi_2\}, \tag{12}$$

$$AUTHEN \in E \subseteq (\Pi_2) = \{e_AUTHEN : \exists O_0, \alpha . e_{AUTHEN} \subseteq msg\,(\alpha(O_0)) \wedge (O_0) \text{ is a role node of } \Pi_2\}. \tag{13}$$

From Equations (11), (12), and (13), it can be known that the encrypted contents e_1, MAC_1 and $AUTHEN$ are implemented in protocol Π_2. In the strand space model, there is a corresponding homomorphism at node O_0 that generates the encrypted content, and O_0 is the role node of protocol Π_2.

3) Determining the strong disjoint encryption. The strong disjoint encryption requires the secondary node having no creation conflicts or extraction conflicts with the TinyPK(Π_1) protocol. According to Equation (11), e_1 has the creation encryption related with protocol *RanRAP*(Π_2) referred to the specific encryption content of e_1. It is not relevant to TinyPK(Π_1), and there is no creation conflicts. MAC_1 is the same as e_1. But for $AUTHEN$, it does not belong to TinyPK(Π_1) or RanRAP(Π_2), and the decrypted contents of $AUTHEN$ do not flow in the trace, which is only used as the middle validation. Even if the decrypted contents combined with the message which generates $AUTHEN$, there are no creation conflicts or extraction conflicts.

From the above analysis, we can find that the secondary node of the combination protocol Π does not cause the creation conflicts or extraction conflicts. Therefore Π and TinyPK(Π_1) satisfy the strong disjoint encryption.

Proposition 7. *The cluster counterexample of A of protocol Π in protocol Π_1(TinyPK) and the realization proof of cluster A in protocol Π.*

For any goal $G_1 \in L(\Pi_1)$, the TinyPK(Π_1)-counterexample A_1 from a Π-counterexample B should be squeezed. This can be achieved by the following two steps. First, B is restricted to its primary node $B \uparrow \Pi_1$(represented by cluster A). Then, all the non-primary encryptions $e \notin E \ll (\Pi_1)$ are removed from A, thus the rest is A_1.

B is first restricted to its primary node skeleton $B \uparrow \Pi_1$ form traces in Equations (6)-(10): $[CH_B, CH_A, \{g_1\}(CH_B)_{sk}, \{g_1, g_2\}(CH_A)_{sk}, e_1, MAC_1, MAC_2, e_2, MAC_3]$.

After all the non-primary encryptions $e \notin E \ll (\Pi_1)$ are removed from A, skeleton A_1 is $[CH_B, CH_A, \{g^x\}(CH_B)_{sk}, \{g_1, g_2\}(CH_A)_{sk}]$.

Proof. The realization of skeleton A is achieved through the authentication test in [9]. There is a new proposed authentication test. Thus we first describe the definition of the authentication test as Lemma 1. □

Lemma 1. *Let c be an atom or an encryption, and S be a set of encryptions. If $\exists n \subseteq nodes(A)$, $Cut(c, S, A)$, is the test cut for c and S in A, we formalize*

$$\begin{aligned} & Cut(c, S, A) \\ = \; & \{n \subseteq nodes(A) : \exists m . m \leq_A n \wedge c\dagger^S msg(m)\}. \end{aligned} \tag{14}$$

According to the new definition of the authentication test in Lemma 1, two important cuts Cut(g_1, S_1, A) and Cut(g_2, S_2, A) should be solved in skeleton A. Cut($g_1, \{g_1, g_2\}(CH_A)_{sk}, A$) is solved at node n1, and Cut($g_2, \{g_1, g_2\}(CH_A)_{sk}, A$) is solved at node m_1. Thus skeleton A is realized.

The final judge of the composable protocol Π_2 is based on the composable theorem in [9]. We describe it as Lemma 2 here.

Lemma 2. *Let Π and Π_1 have the strong disjoint encryption, and let $G_1 \in L(\Pi_1)$ be a security goal. If $A \models \neg G_1$ can be realized, for some realized Π_1-skeleton A_1, $A_1 | = \neg G_1$.*

Theorem 2. ***RanRAP is a composable security***.

Proof. In Theorem 1, $A \models \varphi$ (φ is expressed as two secure claims, φ_1 and φ_2), and Proposition 7 has proved that the skeleton A of protocol Π is realized. Combined with the definition of the counterexample realized in [9], the conclusion $A| = \neg G_1$ is drawn.

According to Lemma 2, the first requirement of protocols Π and Π_1 is that they should have a strong disjoint encryption, which has been described in detail in Proposition 6. Another requirement of Lemma 2 is that $A| = \neg G_1$ should be met, which has also been deduced from Theorem 1 and Proposition 7. Hence Π and Π_1 also satisfy another premise of Lemma 2. A_1 has been given as the counterexample of Π_1 in Π. According to Lemma 2, the conclude $A_1| = \neg G_1$ is drawn. The composable protocol Π does not affect the security goals of protocol TinyPK(Π_1). Thus the composability of the *RanRAP* protocol is concluded. □

5 Protocol Performance Analysis and Comparison

The roaming authentication protocol of the cloud assisted BSN has three aspects of security needs: lightweight, random roaming and composable security. In this section, our work is compared with the related work in terms of these three aspects both qualitatively and quantitatively.

5.1 Comparison with the Related Works

Table 3 lists the comparison between *RanRAP* and related protocols in the aspects of lightweight, randomly roaming and security. The computation overhead is measured by CPU's number of revolutions in a 8 MHz CPU. The message size is measured in the unit of byte. In the following, h_n is the average hops when the mobile node in a cluster reaches the cluster head, n_c is the average number of the neighboring clusters, and " $-$ " means that it is not considered in the security part.

The lightweight is compared in three aspects: communication times, computation overhead and message size. In the comparison of communication times, Han 2010 [13] and our *RanRAP* protocol consider the entire authentication process. The communication times include the transmission time of the authentication materials. The roaming protocols of Yang 2010 [39] and He 2011 [12] used the identity-based cryptography and group signature to realize the local authentication of the roaming protocol. The communication times of the mobile node in their protocols do not contain the transmission of the authentication materials. The former total communication times are greater than communication times involved in the mobile nodes. In the roaming protocol, the mobile node is limited by the resource. Thus the lightweight focuses on mobile nodes. The sensor node in IoT under the mobile environment is more limited in energy, computation capacity and communication capacity. Thus it has more demands on the lightweight. The communication times of Han 2010 [13] are equal or greater than 4 times, because of the re-authentication process after every moving. The protocol stores all the authentication materials into the neighboring nodes through broadcast, and the broadcast communication computes at least once communication.

The numbers of CPU revolutions are used to calculate the computation overhead. It is mainly based on [26] which proposed to use the energy consumption relationship of each algorithm to estimate the results. In the 8 MHz CPU for the Micaz mote, its encryption algorithm, CPU revolution and energy consumption are shown in Table 4.

Table 5 shows the basic cryptographic operations used in the roaming protocol. The cryptographic algorithms corresponding to the energy consumption is shown in Table 4. BCE represents Block Cipher Encryption, MAC means Message Authentication Code Computer, PKE means Public Key Encryption or Decryption, ECSM means Elliptic Curve Scalar Multiplication, P means El-

Table 3: Comparison of related work

protocol	lightweight						Random roaming	security		
	Communic-ation times		Communication overhead (CPU revolution)		Message size			Local identity authentication	Prevent Dos attack	Composable security
	Whole protocol	Mobile node	Whole protocol	Mobile node	Whole protocol	Mobile node				
Yang 2010[9]	3	3	393.225M	198.45M	$74h_n+72$	$74h_n+72$	√	-	√	-
He 2011[10]	3	3	584.225M	294M	$188h_n+72$	$188h_n+72$	√	-	√	-
Han 2010[21]	≥ 4	3	150.388K	53.71K	$52h_n+88$ $+56n_c$	$52h_n+88$	×	-	√	-
Our *RanRAP*	4	2	25.529M	32.226K	$50h_n+108$	$50h_n+28$	√	√	√	√

Table 4: Energy algorithm consumption

Encryption algorithm	Energy consumption	CPU revolution
$AES(128bits)$	$38\mu J$	10742
$ECDSA(160bits)$	$52mJ$	$14.7M$

liptic Curve Bilinear Pairing, and EXP means Modular Exponentiation. The comparison of the computation overhead does not consider the hash algorithm overhead when the protocols run.

Table 5: Protocol encryption operations

Protocol operation	Energy consumption
BCE	$1AES$
MAC	$1AES$
$ECSM$	$1ECDSA$
P	$6ECDSA$
PKE	$2ECDSA$
EXP	$2ECDSA$

The basic computation times of the protocols in Table 3 are shown in Table 6. From Tables 4, 5 and 6, the estimated calculation of the computation overhead can be obtained.

Table 6: The whole protocol computation overhead

Protocols	Whole computation overhead	Whole energy overhead
Yang 2010 [39]	$8.75ECSM + 3P$	$1391mJ$
He 2011 [12]	$15.75ECSM + 4P$	$2067mJ$
Han 2010 [13]	$4BCE + 8MAC$	$456\mu J$

From Table 3, we can find that computation overhead of Han 2010 [13] and *RanRAP* are lower than the similar protocols in an integer magnitude on the mobile sensor nodes. This is mainly determined by the goal of the protocol design. We can obtain the computation time of each protocol based on the CPU revolutions from Table 3. The computation times of Yang 2010 [39] and He 2011 [12] are 24.08625 s and 36.75 s in the 8 MHz CPU for the Micaz nodes, respectively. The theoretical value of computation times does not include the time overhead of the communication delays and the task waiting, but it is still too long for the roaming service. While Han 2010 [13] and *RanRAP* are 6.7125 ms and 4.0256 ms, respectively, it has an obvious applicability regarding the computation overhead. In the comparison of the lightweight between Han 2010 and *RanRAP*, *RanRAP* is more excellent than Han 2010, because *RanRAP* uses fewer cryptographic operations. This is mainly due to the reduced communication times while the average amounts of the computation time of the two protocols are almost the same. In addition, it should be noted that when the local authentication is activated, the number of the consumed revolutions is 25.529 M which is much bigger than Han's solution. However, the decryption algorithm of the public keys runs on the cluster head, which has enough energy to support. If the function of the local authentication is inactivated, the whole process costs 128.904 K.

Table 3 shows the comparison of the message size. We consider the influence of the average relay hops in the cluster and the average adjacent clusters on the entire message size of the protocol (h_n means the average relay hops in the cluster, and nc means the average neighboring clusters). The message size in Table 3 is measured by the data in Table 7. In addition, in the message size calculation process, the message format is beyond calculation, and the message size of the symmetric encryption is an integer multiple of the key length. The message size of the public encryption is only calculated as the size of the encrypted message. The length of the hash value has the same length as the content in the hash function.

From the message size in Table 3, we can find that the message sizes of Han 2010 and *RanRAP* protocol are the shortest. Despite using the public key algorithms, the length of the protocol message only computes according to the length of the encrypted content without considering the specific public key algorithm. However, with the number of relay hops in the cluster increased, the *RanRAP* protocol has a better performance. In addition, Han 2010 protocol is also related with the average number of the adjacent cluster n_c. When n_c is large, the protocol roaming range and the message size are enlarged. Other protocols can execute the random roaming without relying on the location of the home cluster and foreign cluster.

RanRAP protocol uses $AUTHEN$ to filter some illegal joiners by decrypting parameters in the local foreign cluster before the mobile node is authenticated. The specific instructions can refer to the analysis of Proposition 2 in Subsection 3.3.

For the security comparison, some roaming protocols should go to the fixed home cluster to achieve the identity authentication. Thus the DOS attack is unavoidable. In Han 2010 and *RanRAP* protocols, when a mobile node reaches a new cluster, the mobile node doesn't need to go back to the fixed initial cluster to obtain the authentication material in the next roaming, which reduces the harm caused by the DOS attacks. What's more, we propose

Table 7: Bytes of basic protocol

Protocol content	bytes
MAC	4
Time stamp	8
Random number	8
Key	16
Node Id	2
Index operation	50
Elliptic curve length	20

a solution to the DOS attacks caused by the Multi-hop transmission. The details are shown in Subsection 3.4. The composable security is the most distinctive feature of the *RanRAP* protocol compared with other lightweight protocols. That is because when the lightweight protocol is used by the cluster head, the cluster head acts as the gateway which connects the BSN subnets and the backbone network. The lightweight protocol running in the BSN cluster can not affect the security goals of the original protocol. Section 4 completes the composable security proof of the authentication test.

5.2 Protocol Simulation

We simulate the *RanRAP* protocol by using NS2.29, and the transmission delay is used to quantify the message sizes, because the transmission delay can reveal the availability and efficiency of the *RanRAP* protocol. The simulation uses the mesh network topology. The MAC layer uses the 802.15.4 protocol which Zheng [40]wrote for NS2. The routing layer uses the AODV routing protocol which has the shortest hops. The transportation layer uses the UDP protocol, and the application layer transmits the CBR packet. The message size is set according to Table 7 in Subsection 5.1. The data transmission speed is 250 KB/s, which adopts the recommended beacon mode standard setting in [40].

Supposing the communication radius of the mobile node and the sensor nodes within the cluster is 20 m, the communication radius of the cluster head is 100 m. The delay times of the *RanRAP-Request* and *RanRAP-Response* messages in the node are derived from the computation overhead of the mobile node according to Table 3 (using the Micaz mote with 8 MHz CPU). The delay times are 3 ms and 1.5ms. The time delay of *RanRAP-Ask* and *RanRAP-Answer* in the node is set to 1 ms (using a CPU with a frequency of a few hundreds MHz as the cluster head processor).

To analysis the message size of Table 3 in 6, we design two groups of simulations. The number of each simulation is 100 times.

Simulation 1: When $n_c = 1$, we set $h_n = 2$ and $h_n = 5$.

When the protocol runs, only a specified neighboring cluster can roam, and *RanRAP* degenerates as the Han 2010 [13] protocol which is a re-authentication protocol between clusters. Figures 7 and 8 show 100 sets of data obtained from the simulation.

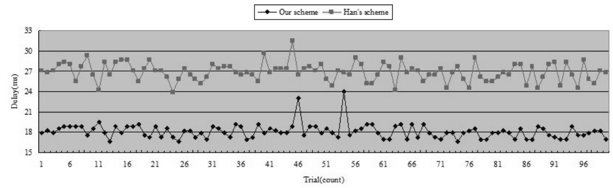

Figure 7: Time delay for $n_c = 1$ and $h_n = 2$

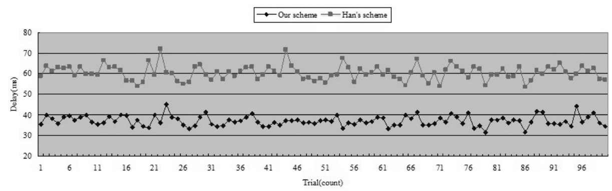

Figure 8: Time delay for $n_c = 1$ and $h_n = 5$

In Figure 7, $n_c = 1$, and $h_n = 2$. The average delay of Han 2010 is 26.864 ms, whereas the average delay of *RanRAP* is 18.1056ms. Contrary to the theoretical analysis of the message size, the delay of the *RanRAP* simulation is less. This is mainly due to the *RanRAP* protocol less once to send a message. When *RanRAP* sends a message, it needs to add the MAC layer header, which makes the simulation delay is not proportional to the message sizes. The fluctuation effect in Figure 7 is mainly caused by $h_n = 2$. During the transmission process of nodes in the cluster, it needs to consume the transmission delay (when the node transmits, it needs to repeat calling, sending and receiving process, and seek the routing table, which needs more delay time). Thus the time is unstable.

In Figure 8, $n_c = 1$ and $h_n = 5$. The average delay of Han 2010 is 60.3689 ms, whereas *RanRAP* is 37.04ms. With the relay hops in the cluster increased, the advantages of the *RanRAP* protocol are more obvious compared with the results in Figure 7. This is due to less communication message sizes of the mobile node. From the comparison between Figures 7 and 8, we can also see that the volatility becomes larger, which is due to $h_n = 5$. This indicates that as the cluster relay hops increase, the instability of the delay is more obvious. Table 3 also shows that the *RanRAP* protocol has the characteristic of the random roaming. In order to reflect the performance advantages by using the network model, we design simulation 2. The simulation assumes that the mobile node randomly roams in a fixed region which has 100 clusters. The average number of the neighboring clusters around each cluster head is $n_c = 4$. In order to reflect the fairness, we assume that the Han 2010 protocol can

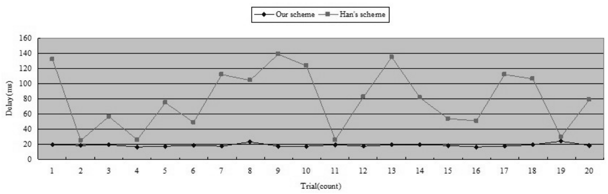

Figure 9: Time delay for $n_c = 4$ and $h_n = 2$

automatically search for the shortest hop to roam to the foreign cluster.

Simulation 2: When $n_c = 4$ and $h_n = 2$, the mobile node roams randomly in the region, and the mobile nodes separately run the *RanRAP* and Han 2010 protocols. We randomly select the foreign cluster in the simulation. The mobile node which runs the *RanRAP* protocol only needs to complete the authentication protocol once, whereas the Han 2010 protocol requires to inquiry the shortest path and then authenticates one cluster by one cluster. Figure 9 shows 20 clusters which are randomly selected to visit, and their simulation times are calculated by selecting the foreign cluster to joining the visit cluster (The maximum roaming hop is 5 and the minimum is 1.).

Figure 9 shows the experimental results of 20 times. The abscissa shows that the crossing cluster roaming hop of the 2nd, 4th, 11th, and 19th experiments is 1. The two protocols have a similar performance in the experiment. The hop of the 3rd, 6th, 15th, and 16th experiments is 2. The hop of the 5th, 12th, 14th, and 20th experiments is 3. The hop of the 7th, 8th, 17th, and 18th experiments is 4. The hop of the 1st, 9th, 10th, and 13th experiments is 5. Under these circumstances, the time delays of Han 2010 are 2, 3, 4, and 5 times that of the *RanRAP* protocol, respectively, because Han 2010 needs to join the neighboring cluster to authenticate several times through the running path.

Note that the message sizes of Han 2010 are proportional to the number of n_c. However, in simulation 2 only one message transmission delay is included. We do not consider the cluster head waiting for the message to send by sequence. In addition, the fluctuation of the same roaming hop is in a wide range as shown in Figure 9. As the hops among the clusters increases during roaming in Han 2010, the direct ratio of hops to the time delays is not obvious. This is due to the accumulation of the transmission fluctuation.

We further demonstrate the random roaming characteristics of the *RanRAP* protocol in Figure 9. It also illustrates the application network model considered by Han 2010, which restricts its advantages in terms of roaming. From the above time delay measured by the simulation, we can find that the whole time delay of the *RanRAP* protocol can be limited within 50 ms. Within the tolerance for the time delay roaming protocol in [22], the normal use and the normal operation of the node itself is not affected, which can achieves a seamless interface in the practical application.

6 Conclusion

With the application and development of the BSN, the BSNs become popular and are distributed widely. Then many BSN clusters are connected with the backbone transmission networks, and the big data collected by BSN require cloud storage and processing. Thus a novel type of cloud-assisted BSNs is presented. We consider the security questions of this type of BSNs with cloud-assisted infrastructure. Especially, we discussed the roaming authentication of the mobile body sensors in this scenario. In this paper, we exploit the advantages of cloud-assisted BSNs based on MWN model, and design an efficient, secure and composable protocol for the mobile nodes roaming randomly in the networks. The security analysis shows that our designed protocol can satisfy the forward security and mutual identity authentication, and can prevent the man-in-the-middle attacks and the replay attacks. The performance analysis shows that the *RanRAP* protocol can achieve lightweight, random roaming and composable security, which is well adapted to the application requirements of the BSN based on cloud-assisted infrastructure.

Acknowledgments

The paper has been supported by Natural Science Foundation of China under No.61003300 and 61272074, and Natural Science Foundation of Jiangsu Province under No.BK2011464. We would like to acknowledge that Prof. QIU Ying and Robert DENG Huijie greatly improved our work and advised us to discuss DoS attack prevention.

References

[1] M. Burmester, T. V. Le, B. D. Medeiros, and G. Tsudik, "Universally composable RFID identification and authentication protocols," *ACM Transactions on Information and System Security (TISSEC)*, vol. 12, no. 4, pp. 21-33, 2009.

[2] C. C. Chang and H. C. Tsai, "An anonymous and self-verified mobile authentication with authenticated key agreement for large-scale wireless networks," *IEEE Transactions on Wireless Communication*, vol. 9, no. 11, pp. 3346–3353, 2010.

[3] S. Chari, C. Jutla, and A. Roy, "Universally composable security analysis of Oauth v2.0," *IACR Cryptology ePrint Archive*, pp. 526, 2011.

[4] M. Chen, S. Gonzalez, A. Vasilakos, H. Cao, and V.Leung, "Body area networks: A survey," *Mobile Networks and Applications*, vol. 16, no. 2, pp. 171–193, 2011.

[5] S. F. Doghmi, J. D. Guttman, and F. J. Thayer, "Completeness of the authentication test," in *The 12th European Symposium on Research in Computer Security (ESORICS 2007)*, LNCS 4734, pp. 106–121, Springer, 2007.

[6] W. Du, R. Wang, and P. Ning, "An efficient scheme for authenticating public keys in sensor networks," in *The 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp.58–67, Chicago, IL, 2005.

[7] T. Feng, W. Zhou, and X. Li, "Anonymous identity authentication scheme in wireless roaming communication," in *2012 8th International Conference on Computing Technology and Information Management (ICCM'12)*, vol. 1, pp.124–129, Seoul, Korea, 2012.

[8] G. Fortino, M. Pathan, and G. Fatta, "BodyCloud: Integration of cloud computing and body sensor networks," in *2012 IEEE 4th International Conference on Cloud Computing Technology and Science (CloudCom)*, pp. 851–856, Taipei, Taiwan, 2012.

[9] J. D. Guttman, "Cryptographic protocol composition via the authentication tests," in *The 12th International Conference on Foundations of Software Science and Computational Structures*, vol. 5504, pp. 303–317, York, UK, 2009.

[10] J. D. Guttman and F. J. Thayer, "Authentication tests and the structure of bundles," *Theoretical Computer Science*, vol. 283, no, 2, pp. 333–380, 2002.

[11] K. Han, K. Kim, and T. Shon, "Untraceable mobile node authentication in WSN," *Sensors*, vol. 10, no. 5, pp.4410–4429, 2010.

[12] D. He, J. Bu, S. Chan, C. Chen, and M. Yin, "Privacy-preserving universal authentication protocol for wireless communications," *IEEE Transactions on Wireless Communications*, vol.10, no.2, pp.431–436, 2011.

[13] D. He, C. Chen, S. Chan, and J. Bu, "Strong roaming authentication technique for wireless and mobile networks," *International Journal of Communication Systems*, vol. 26, no. 8, pp. 1028–1037, 2013.

[14] Y. M. Huang, M. Y. Hsieh, H. C. Chao, S. H. Hung, and J. H. Park, "Pervasive, secure access to a hierarchical sensor-based healthcare monitoring architecture in wireless heterogeneous networks," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 4, pp. 400–411, 2009.

[15] S. Hyun, P. Ning, A. Liu, and W. Du, "Seluge: Secure and DoS-resistant code dissemination in wireless sensor networks," in *The 7th International Conference on Information Processing in Sensor Networks*, pp. 445–456, St. Louis, MO, 2008.

[16] S. Jiang, J. Miao, and L. Wang, "Mobile node authentication protocol for crossing cluster in heterogeneous wireless sensor network," in *2011 IEEE 3rd International Conference on Communication Software and Networks (ICCSN)*, pp. 205–209, Xi'an, China, 2011.

[17] S. Jiang, J. Zhang, J. Miao, and C. Zhou, "A privacy-preserving reauthentication scheme for mobile wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 913782, pp. 1–8, 2013.

[18] D. Kar, R. Tatum, and K. Zejdlik, "MHIP: Effective key management for mobile heterogeneous sensor networks," *International Journal of Network Security*, vol. 15, no. 4, pp. 280–290, 2013.

[19] J. Katz, "Universally composable multi-party computation using tamper-proof hardware," in *26th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, LNCS 4515, pp.115–128, Springer, 2007.

[20] J. Li, B. Bhattacharjee, M. Yu, and R. Levy, "A scalable key management and clustering scheme for wireless ad-hoc and sensor networks," *Future Generation Computer Systems*, vol.24, no. 8, pp. 860–869, 2008.

[21] M. Li, W. Lou, and K. Ren, "Data security and privacy in wireless body area networks," *IEEE Wireless Communications*, vol. 17, no. 1, pp. 51–58, 2010.

[22] X. Li, X. Lu, J. Ma, Z. Zhu, L. Xu, and Y. Park, "Authentication and key management in 3G-WLAN interworking," *Mobile Networks and Applications*, vol. 16, no. 3, pp. 394–407, 2011.

[23] X. Li, Y. Zhang, X. Liu, J. Cao, and Q. Zhao, "A lightweight roaming authentication protocol for anonymous wireless communication," in *2012 Global Communications Conference (GLOBECOM)*, pp.1029–1034, Anaheim, CA, 2012.

[24] M. E. Mahmoud and X. Shen, "An integrated stimulation and punishment mechanism for thwarting packet drop in multihop wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 60, no.8, pp.3947–3962, 2011.

[25] M. E.Mahmoud and X. Shen, "FESCIM: Fair, efficient, and secure cooperation incentive mechanism for multi-hop cellular networks," *IEEE Transactions on Mobile Computing*, vol. 11, no.5, pp.753–766, 2012.

[26] G. de Meulenaer, F. Gosset, F. X. Standaert, and O. Pereira, "On the energy cost of communication and cryptography in wireless sensor networks," in *The 4th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WIMOB'08)*, pp. 580–585, Avignon, France, 2008.

[27] P. Ning, A. Liu, and W. Du, "Mitigating DoS attacks against broadcast authentication in wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 4, no. 1, pp. 1–35, 2008.

[28] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "SPINS: Security protocols for sensor networks," *ACM Wireless Network*, vol. 8, no. 5, pp. 521–534, 2002.

[29] Y. Qiu, J. Zhou, J. Baek, and J. Lopez, "Authentication and key establishment in dynamic wireless sensor networks," *Sensor*, vol. 10, no. 4, pp. 3718–3731, 2010.

[30] C. Ran, "Universally composable security: A new paradigm for cryptographic protocols," in *42nd Annual Symposium on Foundations of Computer Science*, pp. 136–145, Las Vegas, NV, 2001.

[31] C. Ran, "Universally composable security: A new paradigm for cryptographic protocols," *Cryptology ePrint Archive*, Report 2000/067, 2005.

[32] H. Tan, D. Ostry, J. Zic, and S. Jha, "A confidential and DoS-resistant multi-hop code dissemination protocol for wireless sensor networks," in *The Second ACM Conference on Wireless Network Security*, pp. 245–252, Zurich, Switzerland, 2009.

[33] Z. Wan, K. Ren, and B. Preneel, "A secure privacy-preserving roaming protocol based on hierarchical identity-based encryption for mobile networks," in *The First ACM Conference on Wireless Network Security (ACM WiSec'08)*, pp. 62–67, Alexandria, VA, 2008.

[34] J. Wang, Y. Yu, and K. Zhou, "A regular expression matching approach to distributed wireless network security system," *International Journal of Network Security*, vol. 16, no. 5, pp. 382–388, 2014.

[35] L. Wang and Y. Shi, "Patrol detection for replica attacks on wireless sensor networks," *Sensors*, vol.11, no.3, pp. 2496–2504, 2011.

[36] Y. Wang, D. S. Wong, and L. Huang, "One-pass key establishment protocol for wireless roaming with user anonymity," *International Journal of Network Security*, vol. 16, no. 2, pp. 129–142, 2014.

[37] R. Watro, D. Kong, S. Cuti, C. Gardiner, C. Lynn, and P. Kruus, "TinyPK: Securing sensor networks with public key technology," in *The 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks*, pp. 59–64, Washington, DC, 2004.

[38] G. Yang, "Comments on an anonymous and self-verified mobile authentication with authenticated key agreement for large-scale wireless networks," *IEEE Transactions on Wireless Communication*, vol. 10, no. 6, pp. 2015–2016, 2011.

[39] G. Yang, Q. Huang, D. Wong, and X. Deng, "Universal authenticated protocols for anonymous wireless communications," *IEEE Transactions on Wireless Communications*, vol.9, no.1, pp.168–174, 2010.

[40] J. Zheng and M. J. Lee, "A comprehensive performance study of IEEE 802.15.4," *Sensor Network Operations*, Wiley-IEEE Press, pp. 218–237, 2006.

[41] W. Zhu, J. Zhou, R. H. Deng, and F. Bao, "Detecting node replication attacks in mobile sensor networks: theory and approaches," *Security and Communication Networks*, vol. 5, no. 5, pp. 496–507, 2012.

Qing-Qing Xie received her B.S. degree from school of computer science and technology, Anhui University in 2012. Now She is working toward PhD degree in Anhui University, China. Her research interests include cryptology and data security.

Shun-Rong Jiang received his M. S. degree in computer science in Jiangsu University, China, in 2012, and now he is studying for his Ph.D degree in Cryptology at Xidian University China,. His research interests include wireless communication security and cryptographic protocols.

Liang-Min Wang received his B. S. degree in computational mathematics in Jilin University, China, in 1999, and the Ph.D degree in Cryptology from Xidian University, China, in 2007. From 2009 to 2010, he was also a visiting scholar in Nanyang Technological University of Singapore. Now he is an associate professor of Jiangsui University. His research interests include security protocols and wireless sensor networks. Currently, he is a senior member of CCF, and a member of IEEE and ACM.

Chin-Chen Chang received his BS degree in applied mathematics in 1977 and the MS degree in computer and decision sciences in 1979, both from the National Tsing Hua University, Hsinchu, Taiwan. He received his Ph.D in computer engineering in 1982 from the National Chiao Tung University, Hsinchu, Taiwan. During the academic years of 1980-1983, he was on the faculty of the Department of Computer Engineering at the National Chiao Tung University. From 1983-1989, he was on the faculty of the Institute of Applied Mathematics, National Chung Hsing University, Taichung, Taiwan. From August 1989 to July 1992, he was the head of, and a professor in, the Institute of Computer Science and Information Engineering at the National Chung Cheng University, Chiayi, Taiwan. From August 1992 to July 1995, he was the dean of the college of Engineering at the same university. From August 1995 to October 1997, he was the provost at the National Chung Cheng University. From September 1996 to October 1997, Dr. Chang was the Acting President at the National Chung Cheng University. From July 1998 to June 2000, he was the director of Advisory Office of the Ministry of Education of the R.O.C. From 2002 to 2005, he was a Chair Professor of National Chung Cheng University. Since February 2005, he has been a Chair Professor of Feng Chia University. In addition, he has served as a consultant to several research institutes and government departments. His current research interests include database design, computer cryptography, image compression and data structures.