Novel Secure and Dynamic Clustering Protocols in Wireless Sensor Networks

RUI JIANG, Purdue University, Southeast University
MEHDI AZARMI, Purdue University, ACM Student Member
TAO GONG, Purdue University
BHARAT BHARGAVA, Purdue University, ACM and IEEE Fellow

Secure and dynamic clustering protocols, which are essential in wireless sensor network, are becoming a heated research area. However, the proposed protocols are vulnerable to wormhole attack. Most recently, SecDEACH protocol is proposed and claimed to provide both the resilient cluster head election, preserving a dynamic clustering, and the secure cluster formation. In this paper, we demonstrate that SecDEACH suffers from the wormhole attack. In order to address the wormhole attack, we propose two novel secure and dynamic clustering protocols, the NSDCP1 and the NSDCP2. The NSDCP1 is based on feedback messages from the cluster head, and can be applied in the scenario where the cluster head would not be compromised. The NSDCP2 is based on the feedback messages from the base station, and can be applied in the scenario where the base station is secure. Moreover, we formally prove the confidentiality and the authentication of the NSDCP2 based on the Strand Space model and Authentication Tests. Finally, we conduct an extensive simulation study to compare the performance of the NSDCPs with the SecDEACH. The results show that the proposed protocols hold all the merits of the SecDEACH and are more secure.

Categories and Subject Descriptors: C.2.2 [**Computer-Communication Networks**]: *General—Security and Protection, Network Protocols*

General Terms: Design, Protocol, Security

Additional Key Words and Phrases: Wireless sensor networks, Clustering protocol, Security, Wormhole attack, Formal method

1. INTRODUCTION

Recently, the cluster structures are frequently employed in wireless sensor networks for energy saving and efficiency. Each cluster contains a cluster head (CH) sensor node and many non-CH sensor nodes. During transmission phase, each CH should serve as the aggregator that collects the sensing results from the non-CH nodes, aggregates the result, and reports it to the BS [Dong and Liu 2009]. In clustered sensor network, the CH is more vulnerable to the threats than non-CH sensor nodes. If an adversary compromises a few number of CHs, he could control or disrupt a large area of the network. In addition, the dynamic and periodic clustering makes the design of secure clustering protocol and key distribution more challenging.

After Heinzelman et al. proposed LEACH (Low-Energy Adaptive Clustering Hierarchy), many variants of secure clustering protocols in wireless sensor networks have been proposed [Ferreira et al. 2005; Oliveira et al. 2007; Sirivianos et al. 2007; Han et al. 2010]. Secure LEACH protocol [Ferreira et al. 2005] proposed by Ferreira protected the cluster head election in LEACH which is a classical clustering protocol. But

Author's addresses: R. Jiang, M. Azarmi, T. Gong and B. Bhargava, Computer Science Department, Purdue University; email: {jiangr, mazarmi, tgong, bbshail}@purdue.edu. R. Jiang, School of Information Science and Engineering, Southeast University, Nanjing, China.

it can not prevent the attacker to join in the cluster during the cluster formation phase. Based on secure LEACH protocol, Oliveira et al. [Oliveira et al. 2007] proposed a security mechanism, called SecLEACH, which protected the protocol by using random key pre-distribution and μTESLA broadcast authentication mechanism. However, the central cluster head election method was not suitable to the large-size network and might lead to massive communication and computation overhead. Moreover, the existing CH election schemes cannot prevent a malicious node from fabricating its criterion and transmitting the fabricated criterion. To resolve these problems, Sirivianos et al. [Sirivianos et al. 2007] proposed a random value based scheme to elect the cluster head respectively and randomly. However, it was so easy to choose the node with little energy to be the cluster head that would result in the reduction of the lifetime of the networks. Dong [Dong and Liu 2009] proposed an efficient secure cluster head election scheme which considered the residual energy in nodes and elected the cluster head respectively. However, the scheme could not support dynamic clustering which is a heuristic method for an optimal clustering because the author had an assumption that a cluster sector was pre-defined and could not be changed. Based on Dong's scheme, Han et al. [Han et al. 2010] proposed a secure and resilient dynamic clustering protocol, called SecDEACH, to preserve data privacy in wireless sensor networks. It provides both the resilient CH election preserving a dynamic clustering and the secure cluster formation. In the paper, the authors provided an authentication mechanism to prevent the unauthorized nodes from joining in the cluster and made some suggestions on verification methods to prevent the compromised nodes being a cluster head continually. In addition, the authors considered not only the distance between the node and the CHs but also the distance between the CHs and the BS in order to balance the energy among the nodes in the network. Unfortunately, we find the mechanism is vulnerable to the wormhole attack [Karlof and Wagner 2003; Khabbazian et al. 2009], which commonly involves two or more distant malicious nodes colluding to understate their distance from each other by relaying packets along an out-of-bound channel available only to the attacker.

To the best of our knowledge, there is little research done to deal with the wormhole attack in wireless sensor networks, especially in the clustered structure. The existing methods against the wormhole attack can be divided into the proactive and the reactive countermeasures [Khabbazian et al. 2009]. The proactive countermeasures attempt to prevent wormhole formation, mainly using specialized hardware to achieve accurate time synchronization or time measurement. Among the proactive countermeasures, timing-based solutions attempt to restrict the maximum distance between two neighbors by computing the packet travel time [Hu et al. 2003]. In addition, location information and directional antennas can also be used to defend against the wormhole attack. On the other hand, the reactive countermeasures employ the malicious behavior detection mechanism to detect and withstand the wormhole attack, and may not be able to prevent the wormhole formation.

Khabbazian et al. [Khabbazian et al. 2009] proposed a timing-based countermeasure which is more suitable for practical implementation of solutions based on packet travel time measurements. Using this asynchronous method, the nodes do not need to have synchronized clocks, and are not required to predict the sending time or to be capable of fast switching between the receive and send modes. However, it is assumed that each node is able to record the time at which a packet is fully sent/received. We improve this countermeasure to be applied to the SecDEACH protocol and suggest an improved version which can prevent wormhole attack easily without adding additional message, which is described in our previous paper [Liu and Jiang 2011]. In our earlier paper [Liu and Jiang 2011], we proposed an improved a secure and dynamic timing-based clustering protocol to withstand the wormhole attack efficiently. This approach is in the

proactive countermeasure category, which attempts to prevent wormhole formation. However, in our earlier scheme, each node needs to maintain a local timetable which leads to great space overhead.

Current methods against the wormhole attack mainly focus on the secure routing protocol. These approaches implicitly assume a trusted relationship among nodes and also a secure data aggregation protocol, which ensure both confidentiality and integrity of the messages. However, these secure data aggregation protocols [Castelluccia et al. 2005; Albath and Madria 2009; Huang et al. 2010] do not take the problem of node identity into consideration, and will lead to the wormhole attack. The scheme proposed in [Castelluccia et al. 2005] is based on additive homomorphic encryption that is not relied on decryption of the ciphertext at the intermediate nodes. This scheme is used for aggregation purposes which provides an end to end security. The SHA mechanism, discussed in [Albath and Madria 2009], provides an end to end confidentiality in addition to data integrity by using additive digital signatures. The drawback of this scheme is that the signature scheme can only verify whether the data has been tampered with or not, but not able to identify the malicious node. Huang et al. [Huang et al. 2010] proposed a novel approach for eliminating duplicate encrypted data during aggregation without decryption. It only considers the comparison of the messages and aggregation of the identical data. However, the main idea in these protocols is to divide the functionalities to the secure routing and secure data aggregation, which is not effective to address some problems such as wormhole attack.

In this paper, we focus on the methods against the wormhole attack mainly in the data transmission phase. We propose two novel secure and dynamic clustering protocols (NSDCP), which are NSDCP1 and NSDCP2, based on feedback messages to handle the wormhole attack. The NSDCP1 can be applied in the scenario where the cluster head would not be compromised. The NSDCP2 can be applied in the scenario where the base station is secure and would not be compromised, which is the basic assumption and the least secure requirement in the wireless sensor networks. Moreover, the node in our new protocols should not maintain any timetable. The paper is organized as follows. Section 2 is the brief review of the SecDEACH. Section 3 introduces the wormhole attack in detail. We propose our novel protocols in Section 4 and carry out the formal analysis of one of our new protocols during Section 5. Section 6 is the performance simulation on our protocols vs. SecDEACH. Finally, we conclude the paper in Section 7.

2. OVERVIEW OF THE SECDEACH PROTOCOL

The SecDEACH protocol [Han et al. 2010] is the security framework of the DEACH clustering protocol. As a secure clustering protocol, it contains five phases, which are the *initiation before deployment*, the *preparation after deployment*, the *secure cluster head election*, the *secure Cluster Formation* and the *secure data aggregation*. Below is a brief review of each phase.

— During the *initiation before deployment* phase, the key information in both sensor nodes and BS is pre-assigned, which will be introduced in details in section 3.1.
— The protocol makes the verification of the nodes and produces the pairwise key with neighbor nodes in the *preparation after deployment* phase. This phase is critical to the the security of the following protocol which relies on the authentication of nodes in this phase. In detail, it adopts the Blom's key pre-distribution [Blom 1985; Blundo et al. 1993] and special node IDs generation method to tie the ID of a sensor node to its cryptographic key. The successful decryption and verification of messages allows the sensor node to convince that the message is originated from a legitimate node.

— The proposed scheme claims that each sensor node becomes a CH one per $\frac{1}{P\prime_{opt}}$ rounds on average which is controlled in the Secure Cluster Head Election phase and the Secure Cluster Formation phase. Although every node can be self-elected as a CH, the non cluster head nodes are able to verify whether the CH is compromised or not and then choose a credible CH as their cluster head. In other words, the malicious node cannot be the CH continually due to the detection from the legitimate non cluster head sensor node.
— In the Secure Data Aggregation phase, double homomorphic encryption algorithm [Kumar and Madria 2010] is used to achieve end to end security in the protocol. It is the BS that can decrypt the message while the CHs only perform concealed data aggregation and relay the message to the BS.

3. WORMHOLE ATTACK

3.1. System Model

The notations used in this paper are described as follows:

S_x: Node x
$S_x(id)$: Node x's ID
A key chain: Key chain which includes the keys $A_{i,0}, A_{i,1}, A_{i,2}$
G key chain: Key chain which includes the keys $G_{i,0}, G_{i,1}$
v key chain: Key chain which includes the keys v_0, v_1, v_2
$f(i,.)$: Polynomial share pre-assigned by Node i
P_{opt}: The optimal probability of cluster head in the entire network
P'_{opt}: The optimal probability of being a cluster head of a sensor node based on the distance from itself to the base station
A: Set of compromised nodes
B: Set of legitimate nodes
N: The total number of nodes in network
R: The maximum transmission range

N resource-constrained sensor nodes distributed uniformly on the square area of size $A = a \times a$. Each node S_i is assigned a polynomial share $f(i,.)$ that is used to create symmetric pairwise keys between other nodes. In addition, every node S_i should be assigned two hash key chains: A key and G key chains. These chains are used to verify the identity of sensor nodes. Each sensor nodes ID and $S_i(id)$ can be computed through $S_i(id) = H(A_{i,0} \| G_{i,0})$.

Based on the Bloms key pre-distribution [Blom 1985], each node should save $\lambda + 1$ numbers and its node ID, $S_i(id)$, to manage the key distribution with λ secure property. In other word, if the adversary compromises less than or equal to λ nodes, the rest of nodes are still secure; otherwise, if the adversary compromises more than λ nodes, all pairwise keys of entire network are captured. Moreover, because the sensor nodes have a limited communication radius, each node S_i can reach its neighbor nodes only within S_i's radio range.

3.2. Threat Model

Sensor networks are deployed in hostile environment and each sensor node lacks physical protection. So it is reasonable for the attacker to compromise a minority of the nodes. We should assume that an adversary can perform a series of attacks including capturing a small number of nodes to obtain their pairwise keys and verification information. In addition, we assume malicious nodes can collude to attack by exchanging the key information with other compromised nodes.

The wormhole attack can be launched in two different modes [Khabbazian et al. 2009]. In the hidden mode, the pair attacker just act as two simple transceivers to

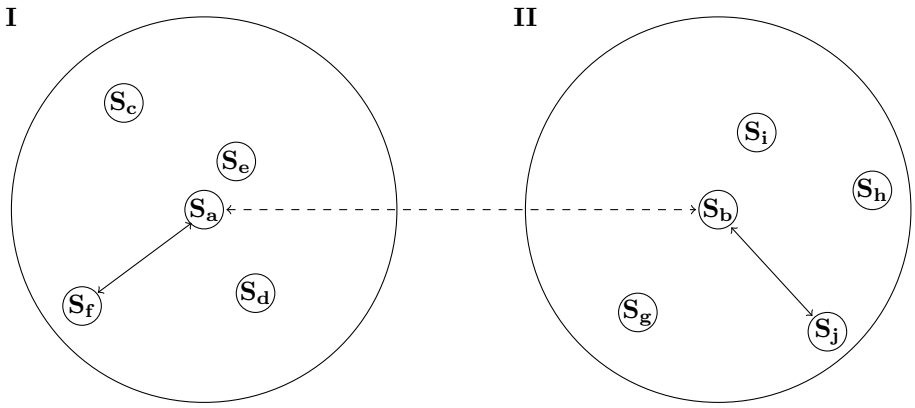

Fig. 1. The wormhole attack in the far-distance model.

eavesdrop the messages at one end of the wormhole and to replay them at the other end. After being included in the route between the source and the destination, the attackers can drop data packets to disrupt the network. In the participation mode, the attackers commonly possess valid cryptographic keys after capturing nodes and can participate in the routing as legitimate nodes which can launch a more powerful attack.

Clearly, the adversary can perform the wormhole attack in two modes, threatening not only the confidentiality but also the integrity and availability of the network. The former does not need compromise nodes, and the latter does.

3.3. Attack Description

Based on SecDEACH protocol, we describe a wormhole attack which poses a great danger to the entire network. Consider two malicious nodes, $S_a, S_b \in A$. If S_a tells S_b its secrets, then S_b can masquerade as S_a to all of S_b's legitimate neighbors, and vice versa. Due to the limited transmission range, the legitimate nodes do not determine the true identity of the compromised node, so may be easily controlled by the attacker. In SecDEACH protocol [Han et al. 2010], the adversary may choose one of the modes to perform the wormhole attack, in the hidden mode or in the participation mode. Whichever mode is chosen, the attack will lead to a devastating consequence to the system.

Figure 1 shows the wormhole attack where the compromised node is far away from each other. Consider the example of the wormhole attack of far-distance colluding in Figure 1. $S_a, S_b \in A$ and $S_c, S_d, S_e, S_f, S_g, S_h, S_i, S_j \in B$. In area I, node S_a's neighbors are S_c, S_d, S_e, S_f, while in Area II node S_b's neighbors are S_i, Sg, S_h and S_j. The distance between S_a and S_b is far longer than the transmission range. Thus, S_a can communicate legitimately with S_c, S_d, S_e, and S_f and S_b can communicate legitimately with S_g, S_h, S_i, and S_j. But, if S_a and S_b collude to exchange each other's information, then S_a can masquerade as S_b or all of S_b's neighbors to communicate with S_a's neighbors. Unfortunately, the existing routing mechanism in the protocol considers only the reliability of the nodes rather than the verification of nodes within the transmission range. Therefore, colluding with S_b enables S_a to present multiple identities to S_a's neighbors.

Following is the table of multiple identities which S_a and S_b can masquerade when S_a and S_b collude:

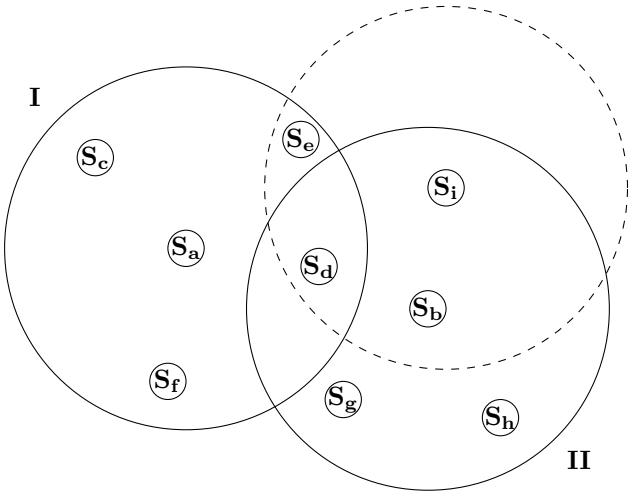

Fig. 2. The Wormhole attack in the near-distance model.

	Fake identites impersonated
S_a	S_b, S_g, S_h, S_i, S_j
S_b	S_a, S_c, S_d, S_e, S_f

Once the attack begins, S_a eavesdrops an announcement of a cluster head from other nodes in Area I. Then, S_b, who colludes with S_a, impersonates the CH in area I and spoofs the non-CH nodes in area II. Furthermore, S_b claims S_a is the best candidate CH by broadcasting the messages to a longer range. Therefore, in this round of CH election, the attackers, S_a and S_b, control area II. In a similar way, S_a may collude with S_b to cheat and control the non-CH nodes in area I. In the following rounds, the adversary masquerades different CH identities to continuously control the network.

Figure 2 shows the situation in near-distance between compromised nodes. In this figure, node S_a may still masquerade S_b, S_g, S_h, S_i and S_d. However, the attackers, S_a and S_b, cannot control the whole network by impersonating each other. In this case, if S_a colludes with S_b to impersonate S_i who announces to be a CH in area II . However, S_e and S_d may receive the same announcement from real S_i, because S_e and S_d are within the radio range of S_i. The impersonation of S_i will fail. Therefore, S_a can only attract S_c and S_f to join the CH with the identity of S_i and control them through impersonation, while S_e and S_d can still transmit the sensing data to real CH, S_i.

Clearly, the adversary can control more nodes in far-distance colluding attack than in the near-distance. Thus the attacker actually prefers to perform far-distance colluding attack. After the adversary controls the network, they commonly discard the data packet or selectively forward the data.

4. PROPOSED PROTOCOL

In our early paper [Liu and Jiang 2011], we proposed a proactive countermeasure based on packet travel time measurements which can prevent wormhole attack easily without adding additional message. However, each node needs to maintain the timetable which leads to high space overhead.

In this paper, we get to consider the countermeasure of the wormhole attack mainly in the transmission phase, and propose two kinds of methods based on the feedback messages to deal with the wormhole attack. Considering the resource scarcity in the wireless sensor networks, we improve the traditional message-reply method, and pro-

vide two improved methods against the wormhole attack based on SecDEACH clustered protocol.

4.1. The NSDCP1

The SecDEACH protocol [Han et al. 2010] cannot prevent the cluster head sensor node from being compromised by itself, but it can make sure that even if the CH is compromised, the adversary cannot continue to control properly the cluster. Thus, based on SecDEACH protocol, we focus on the countermeasure against the wormhole attack in data transmission phase.

Our novel protocol, the NSDCP1 which is based on feedback message from the cluster head, also has five phases. The *initiation before deployment* phase, the preparation after deployment phase and the secure cluster head election phase are same as the ones in SecDEACH protocol. The secure cluster formation phase and the secure data aggregation phase are improved to withstand the wormhole attack. The whole protocol is described in details as follows.

4.1.1. Initiation Before Deployment. Before deployment, the BS creates the $BROADCAST$ key chain $\left\{K^0, K^1, \ldots, K^{br_{max}}\right\}$, where br_{max} II of broadcasts from the BS, using one-way hash key chain method in order to apply μTESLA for the BS's authenticated broadcast. In addition, the BS should create a pairwise symmetric key, K_i, and $counter_i$ for all sensor nodes, where $i = 1, 2, \ldots, N$. And then, BS should maintain a symmetric bivariate polynomial which is a key material for establishing a pairwise key between sensor nodes after deploying.

In terms of sensor nodes, prior to deployment, each sensor node S_i is assigned with two random one-way key chains, the A-key chain and the G-key chain, which are both generated by using one-way hash key chain method. The A-key chain assigned in sensor node S_i contains $A_{i,0}$, $\left\{A_{i,1}, A_{i,2}, A_{i,3} \ldots A_{i,\delta \times P_{opt} \times er_{max}}\right\}$. where er_{max} is the maximum number of rounds for CH election, and the value of δ is 2. The G-key chain in sensor node S_i contains $G_{i,0}, G_{i,1}$. Moreover, each sensor node S_i is assigned a $nodeID$ which is computed by $S_i(id) = H(A_{i,0} \| G_{i,0})$ and is pre-distributed a polynomial share $f(i, .)$ that is used to create a symmetric pairwise key between nodes. Finally, the sensor nodes should also create a pairwise symmetric key, K_i, and $counter_i$ with the BS.

4.1.2. Preparation After Deployment. Fig.3 shows the message flow during the preparation after deployment. Firstly, BS sends a broadcast authentication message to sensor field by using μTESLA in order to tell every node to start the clustering protocol. After receiving this message, each sensor node S_i verifies whether this message is sent from the BS. Whenever a sensor node S_i receives the verified broadcast authentication message from the BS, S_i computes the distance between the BS and the sensor node based on the received signal strength of the broadcast message, saves it, and broadcasts its $nodeID$, $S_i(id)$,within their transmission range R.

Secondly, after receiving other node's ID, $S_i(id)$, each sensor node S_j adds $S_i(id)$ to its neighborhood sensor nodes list $S_{neighbor}(j)$, and establishes a pairwise key $K_{j,i}$ with node S_i by using Blundo's pairwise key pre-distribution scheme [Blom 1985]. According to Blundos scheme, the pairwise key between S_i and S_j is $K_{i,j} = K_{j,i}$. In addition, each sensor node S_j computes the optimal probability being a CH of a sensor node based on the distance from itself to the BS with the equation (1):

$$P'_{opt}(j) = \frac{3}{2} P_{opt} \times \frac{d(S_j, BS) - d_{min}}{d_{max} - d_{min}} \tag{1}$$

Where P_{opt} is the optimal probability of being a CH which is a system parameter. $P'_{opt}(j)$ is the optimal probability of being a CH of sensor node j. The $d(S_j, BS)$ is the

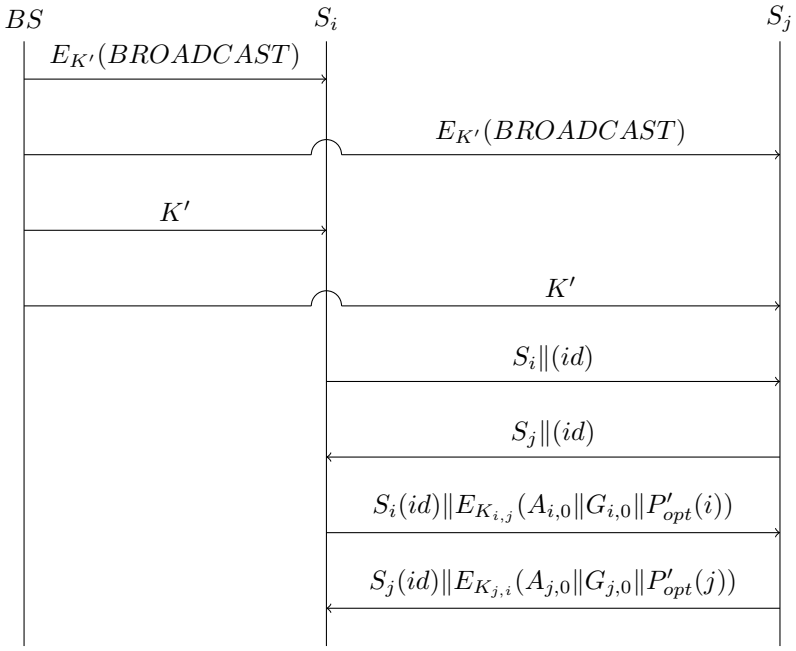

Fig. 3. The message flow in preparation after deployment phase in NSDCP1.

distance between node S_j and BS, d_{max} represents the distance of the farthest sensor node from the BS and d_{min} represents the distance of the closest sensor node.

After each sensor node S_j receives all nodes' ID, it computes $E_{K_{j,i}}(A_{j,0}\|G_{j,0}\|P'_{opt}(j))$ for each sensor node S_i and then sends it out with $S_j(id)$ to that node.

Thirdly, after receiving this message, each sensor node S_i can easily verify these keys according to the ID. Whenever a sensor node S_i receives the verified keys from a node S_j, S_i creates $Q_i(j)$ which means S_j has a qualification being a CH of S_i. The $Q_i(j)$ contains $S_i(id)$, $\frac{1}{P'_{opt}}$ rounds in which the last round the node S_j was a CH, the number of announcements within last $\frac{1}{P'_{opt}}$ rounds, and the last A-key. After collecting all those authenticated A-chain key from other node, every sensor node S_i creates a qualification list $S_q(i)$ in which it consists of array of $Q_i(j)$.

4.1.3. Secure Cluster Head Election. Figure 4 shows the message flow in secure cluster head election phase.

Firstly, each sensor node S_i should compare the residual energy $E_{res}(i)$ with the minimum energy threshold E_{th} which is a pre-determined system value. If $E_{res}(i) < E_{th}$, the sensor node S_i broadcasts $S_i(id)$ and the key of its G-key chain to announce the others that the he can not serve as the CH. If $E_{res}(i) > E_{th}$, node S_i computes the probability of being a CH, P_i, with the equation (2):

$$P_i = \frac{P'_{opt}(i)}{1 - P'_{opt}(i) \times (r \bmod \frac{1}{P'_{opt}})} \times \frac{E_{res}(i)}{E_{init}(i)} \tag{2}$$

Where r is the current round of the CH election, and $E_{init}(i)$ is the initial energy of the sensor node S_i. The node S_i then chooses a random number between 0 and 1. If random number is less than P_i, this node becomes the CH, and broadcasts $S_i(id)$

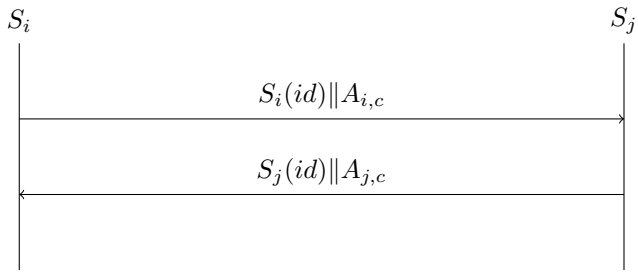

Fig. 4. The message flow in secure cluster head election phase in NSDCP1.

and the last key of its A-key chain $A_{i,c}$ to announce the others that the node S_i is self-elected as CH. The cluster head announcements are commonly sent β times to tolerate the channel loss within its transmission range.

Secondly, when a sensor node S_j receives a key in the G-key chain of S_i, node S_j can verify it by computing and comparing $G_{i,0} = H(G_{i,1})$ with the key $G_{i,0}$ received earlier. If the verification succeeds, node S_j will remove S_i from $S_q(j)$. Otherwise, this announcement is ignored. When a sensor node S_j receives a key in the A-key chain of S_i, node S_j can verify it by computing $A_{i,c-1} = H(A_{i,c})$ and comparing it with the key $A_{i,c-1}$ received earlier. If the verification failed, this announcement is ignored. Otherwise, S_j determines whether S_i has a qualification being a CH by using the equation (3):

$$V(i) = \begin{cases} 1 & \text{if } r - r' \geq r \bmod (\frac{1}{P'_{opt}(i)}) \\ 0 & \text{Otherwise} \end{cases} \tag{3}$$

Where r is the current round, r' is the last round in $Q_j(i)$. If $V(i) = 1$, S_j adds S_i into cluster head candidate list $S_{CH}(j)$ because it means that S_i is a non-compromised node. Otherwise, this announcement is ignored. If this announcement continues over α times within $\frac{1}{P'_{opt}(i)}$ rounds, S_i is removed permanently from $S_q(j)$. Finally, after $S_{CH}(j)$ is created, S_j chooses the closest CH_i among the CHs in $S_{CH}(j)$ and joins to the CH_i as a member. After joining in, it updates the A-key and the last round in $Q_j(i)$.

4.1.4. Secure Cluster Formation Phase. After electing the CH_j to be its cluster head, the S_i sensor node sends $joinMsg(S_i, CH_j)$, which is equal to $S_i(id)\|S_j(id)\|nonce\|MAC_{K_{i,j}}(S_i(id)\|S_j(id)\|nonce)$, to the CH in the same way as the original protocol does.

In detail, node S_i chooses a random number, nonce, and computes the $MAC_{K_{i,j}}(S_i(id)\|S_j(id)\|nonce)$ using the pairwise key with CH_j. And then S_i sends $S_i(id)$, $S_j(id)$, $nonce$, and $MAC_{K_{i,j}}(S_i(id)\|S_j(id)\|nonce)$ to the designated CH.

When the cluster head sensor node CH_i receives the $joinMsg(S_i, CH_j)$ from the node S_i, node CH_j can verify it by computing with the $MAC_{K_{i,j}}$ and comparing with the $MAC_{K_{i,j}}$ received earlier. If the verification succeeds, node CH_j creates a group key key and a one-way key chain $\{v_0, V_1, \ldots, V_R\}$ which is generated by iteratively performing the one-way hash function $H(.)$ on the last key v_R in the chain. Then, node CH_j sends the time slot schedule, key and v_0 to their cluster member nodes by using their pairwise keys.

4.1.5. Secure Data Aggregation Phase. Non-cluster head sensor node transmits $reportMsg(S_i, CH_j)$, which equals to $S_i(id)\|S_j(id)\|Enc_{K_i}(d_i)\|nonce + 1\|MAC_{K_{i,j}}(S_i(id)\|S_j(id)\|Enc_{K_i}(d_i)\|nonce + 1)$. This process is the same as the original protocol, where d_i is the secret data from

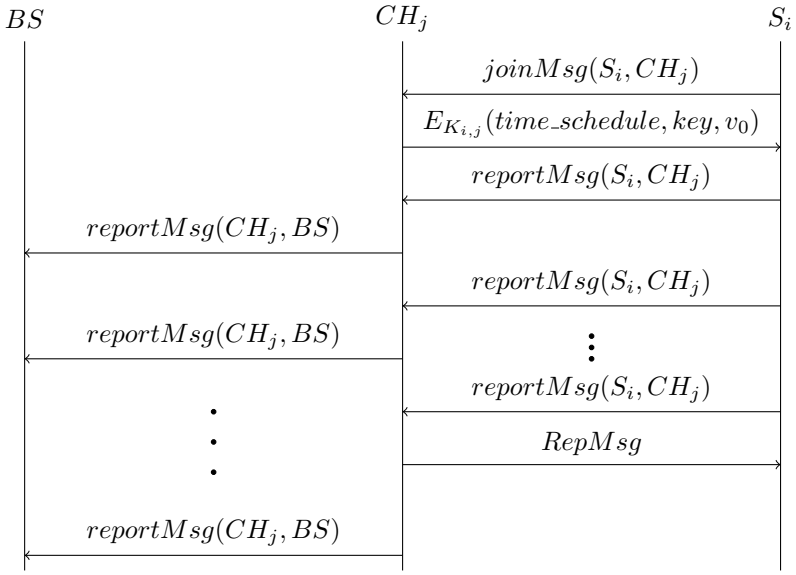

Fig. 5. The message flow in secure cluster formation phase and secure data aggregation phase in NSDCP1.

S_i. Then the CH_j transmits $reportMsg(CH_j, BS)$, which equals to $S_j(id)\|L_{members}(j)\|d_{sum}\|counter\|MAC_{K_j}(S_j(id)\|L_{members}(j)\|d_{sum}\|counter)$, to the BS. $L_{member}(j)$ is the ID's list of the reporting member nodes of CH_j. d_{sum} is the sum of all secret data from reporting sensor nodes. In NSDCP1, both the cluster head node CH_j and the non-sensor node S_i need to record the number when S_i transmits its message. For example, when CH_j receives the message from S_i, CH_j should add 1 to $count_i$, which is saved in CH_j, to record the number of messages transmitted from S_i earlier. Similarly, When S_i sends a message, S_i should add 1 to $count'_i$ which is saved in S_i. After the sensor nodes within the cluster transmit $five$ rounds of messages, the cluster head needs to reply $RepMsg = \{v_1\|H(K_{j,m}\|count_m)\| \ldots \|H(K_{j,i}\|count_i)\|rand\|MAC_{key}(v_1\|H(K_{j,m}\|$ $count_m)\| \ldots \|H(K_{j,i}\|count_i)\|rand)\}$ within its transmission range to notify these non-CH nodes whether the cluster head has received their messages. The sensor node receiving this message firstly verifies the integrity of the message through MAC with the key. Then it can easily verify the message authenticity by computing $v_0 = H(v_1)$ and comparing with the key v_0 earlier. The freshness is guaranteed by $rand$. If the verification succeeds, node S_i computes $VerS_i = H(K_{j,i}\|count'_i)$ where $K_{j,i}$ is the pairwise key between CH_j and S_i. Then, node S_i needs to look inside $RepMsg$ to find the value same as the $VerS_i$. If S_i could find it, which shows that CH_j indeed have received the above five rounds of messages transmitted from S_i and the identity of CH_j is not fabricated. Otherwise, if S_i could not find it, which shows that CH_j is not his authentic cluster head or the messages transmitted/received earlier are attacked by the adversary. Therefore, the non-cluster head node will not send messages to this CH any more and will reselect the new CH_j later. The figure 5 shows the message flow in secure cluster formation phase and secure data aggregation phase.

In our novel method, although the adversary is able to build the wormhole and then lunch the collusion attack, the sensor nodes can detect whether the system is attacked by the adversary through verifying the feedback message from the CH. The CH replies

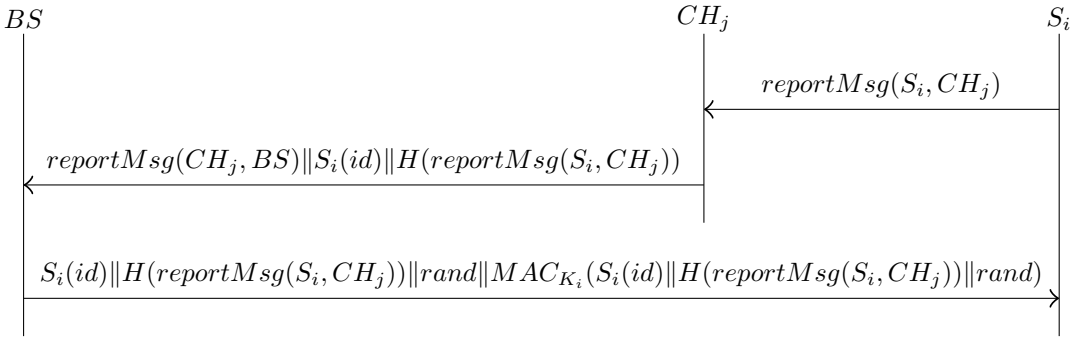

Fig. 6. The message flow in secure data aggregation phase in NSDCP2.

$RepMsg$ every $five$ rounds to save energy. Flexibly, the CH could be configured to reply $RepMsg$ every ten or more rounds for more power efficiency.

However, in this method, when the CH is attacked and compromised, the adversary could disguise the identity to the cluster nodes while selectively forwarding the messages to the BS. In order to deal with the problem, we propose another method based on feedback message from the base station which bypasses the CH and keeps end to end authentication.

4.2. The NSDCP2

The NSDCP2, which is based on the feedback message from the base station, need not assume that the CH would not be compromised. It only assumes the base station is secure, which is the basic assumption and the least secure requirement in the wireless sensor networks. Through this method, the sensor node could judge whether the message it sends is received by the BS and prevent the wormhole attack at utmost. In cluster sensor network like SecDEACH protocol, the base station is able to reply a message to every non-cluster head sensor node after all cluster heads transmit the sensing data to BS in current round.

In NSDCP2, the *initiation before deployment*, the *preparation after deployment*, the secure cluster head election and the secure cluster formation are the same as the ones in NSDCP1. Only the secure data aggregation is changed for defending wormhole attack. We describe the secure data aggregation phase in detail.

The figure 6 shows the message flow in secure data aggregation phase in NSDCP2.In Secure data aggregation phase, non-cluster head sensor node transmits the sensing data in the same way that the original protocol does. After receiving all of the sensing data within their cluster, every cluster head node not only transmits $reportMsg(CH_j, BS)$, but also sends the hash value of every message, which is $H(reportMsg(S_i, CH_j))$, and the node id $S_i(id)$. When the base station receives all of the sensing data, it should send $RepMsg = S_i(id) \| H(reportMsg(S_i, CH_j)) \| rand \| MAC_{K_i}(S_i(i) \| H(reportMsg(S_i, CH_j) \| rand)$ to every non-cluster head sensor node S_i, where $rand$ is used to prevent the replay attack and K_i is the pairwise symmetric key for all sensor nodes to protect the node-to-BS communication. After receiving $RepMsg$, the non-cluster head node S_i can verify it by computing MAC_{K_i} and comparing with the MAC_{K_i} received earlier. If the verification succeeds, it shows that the BS has received the sensing data which is sent earlier. Otherwise, if the node does not receive the message or the verification fails, it shows that CH_j is compromised or it is not his authentic cluster head or the messages transmitted/received earlier is attacked and forged by the adversary.

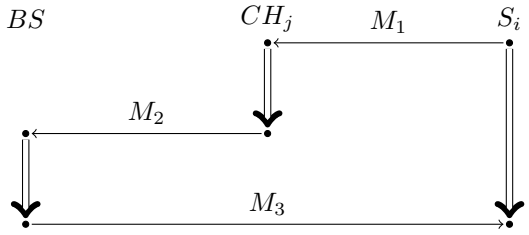

Fig. 7. Formal description of the fifth phase in NSDCP2.

Moreover, in NSDCP2, the base station is not resource constrained and can have the energy to dissipate. The BS should reply a feedback message to all non-cluster head nodes every round of the message, hence the sensor node do not cost much more energy while protecting well against the wormhole attack.

5. FORMAL ANALYSIS OF NSDCP

NSDCP1 and NSDCP2 are the same in the *initiation before deployment* phase, the *preparation after deployment* phase, the *secure cluster head election* phase and the *secure cluster formation* phase. Both of the NSDCP1 and the NSDCP2 are similar to the SecLEACH in the first three phases. The security of the first three phases in SecDEACH is proved [Han et al. 2010]. Therefore, we focus on the NSDCP2 and formally prove it in confidentiality and authentication based on the strand space model [Thayer et al. 1999] and authentication test [Guttman and Thayer 2002] in this paper. The proof of the NSDCP1 is similar. We prove the security of the NSDCP2 as follows.

5.1. Confidentiality Analysis

According to the strand space model [Thayer et al. 1999] and authentication test [Guttman and Thayer 2002], the formal description of the fifth phase in the NSDCP2 is detailed in figure 7. The message M_1 is formalized as $S_iS_j\{|d_i|\}_{K_i}nonce+1\{|S_iS_j\{|d_i|\}_{K_i}nonce+1|\}^{K_{i,j}}$. The message M_2 is formalized as $S_jL_j\{|d_i|\}_{K_i}counter\{|S_jL_j\{|d_i|\}_{K_i}counter|\}^{K_j}H(M_1)$, and finally, the message M_3 is formalized as $S_iH(M_1)rand\{|S_iH(M_1)rand|\}^{K_j}$.

Theorem 1. Suppose Σ is a strand space; C is a bundle of Σ; $s_B \in BS[S_j(id), S_i(id), \ldots, \Sigma d_i, counter]$ with C-height 2; $s_C \in CH_j[S_j(id), S_i(id), \ldots, d_i, nonce + 1, \Sigma d_i, counter]$ with C-height 2; $s_S \in S_i[S_i(id), rand, s_j(id), \ldots, d_i, nonce+1]$ with C-height 2; $K = K^{-1} \notin \mathbf{K}_P$, $K_i \notin \mathbf{K}_p$; let $S = \{d_i, K_i\}$ and $\underline{K} = (K \ / \ S)^{-1}$, for every regular node $n \in C, term(n) \notin I_{\underline{K}}[S]$.

PROOF BY CONTRADICTION. According to [thayer1999strand], if there is a regular node n where $term(n) \in I_{\underline{K}}[S]$, then one of the members d_i or K_i should be a sub-term of $term(n)$. As there is no regular node contains any key of the K_i as a sub-term, so d_i should be the sub-term of $term(n)$.

If n is a positive regular node on a strand s, then $d_i \in term(n)$ implies $n =< s_S, 2 >$. By the unique origination of K_{ASME}, $term(n) = S_i(id)randS_j(id)d_inonce + 1 \in I_{\underline{K}}[S]$, and $K_i \in \underline{K}$, which is conflicted with the suppose $\underline{K} = (K \ / \ S)^{-1}$. Therefore, d_i is confident. □

Theorem 1 proves the secret value d_i can be ensured and will not be compromised in NSDCP2. Hence, the confidentiality of K_{ASME} , which is predicted from d_i, can be ensured. Therefore, the confidentiality of NSDCP2 is ensured. The NSDCP2 can withstand all kinds of secret leaking attack such as fabrication attack.

5.2. Authentication Analysis

Theorem 2. Suppose Σ is a strand space; C is a bundle of Σ; $s_B \in BS[S_j(id), S_i(id), \ldots, \Sigma d_i, counter]$ with C-height 2; if $K_i \notin \mathbf{K}_P$ and the counter is uniquely generating, then there is a regular strand $s_S \in S_i[S_i(id), rand, S_j(id), d_i, nonce + 1]$ with C-height 2.

PROOF . We show first the node on s_B form an ingoing test for d_i. $\{|d_i|\}_{K_i}$ is a test component for d_i, because it contains d_i, and no regular node has any term of this form as a proper sub-term. For $K_i \notin \mathbf{K}_P$ in the assumption, so $< s_B, 1 > \Longrightarrow < s_B, 2 >$ is an ingoing test for d_i in $\{|d_i|\}_{K_i}$. By ingoing test, there exist regular nodes $n_0, n1 \in C$, such that $\{|S_j L_j \{|d_i|\}_{K_i} counter|\}^{K_j}$ is a component of n_1 and $n_0 \implies n_1$ is a transforming edge for $\{|d_i|\}_{K_i}$.

Because n_1 is a positive regular node and $term(n_1) = S_i H(M_1) rand \{|S_i H(M_1) rand|\}^{k_j}$, d_i is uniquely originated in $< s_S, 1 >$, then there exists a negative regular node n_0 to receive d_i. For n_0 is a negative node, it is at $< s_S, 1 >$ for some $s_S \in S_i[S_i(id), rand', S_j(id)']$, Since $< s_S, 1 > \Longrightarrow < s_S, 2 >$ and $term(< s_B, 2 >) = S_i(id) rand S_j(id) d_i nonce + 1$, we see that $S_i(id) = S_i(id)$, $rand' = rand$, $S_j(id)' = S_j(id)$, $d_i = d_i$, $nonce + 1 = nonce + 1$ and the C-height of s_S is 2. □

Theorem 3. Suppose Σ is a strand space; C is a bundle of Σ; $s_S \in S_i[S_i(id), rand, S_j(id), d_i, nonce + 1]$ with C-height 2.; if $K_i \notin K_P$, then there is a regular strand $s_B \in BS[S_j(id), \ldots, \Sigma d_i, counter]$ with C-height 2.

PROOF . We show first the node on s_S form an outgoing test for d_i. $\{|d_i|\}_{K_i}$ is a test component for d_i, because it contains d_i, and no regular node has any term of this form as a proper sub-term. For $K_i \notin K_P$ in the assumption, so $< s_S, 1 > \Longrightarrow < s_S, 2 >$ is an outgoing test for d_i in$\{|d_i|\}_{K_i}$.

By outgoing test, there exist regular nodes $n_0, n_1 \in C$, such that $\{|d_i|\}_{K_i}$ is a component of n_0 and $n_0 \implies n_1$ is a transforming edge for d_i.

Because n_1 is a positive regular node and $term(n_1) = S_i S_j \{|d_i|\}_{K_i} nounce + 1 \{|S_i S_j \{|d_i|\}_{K_i} nounce + 1|\}^{K_j}$, d_i is uniquely originated in $< s_S, 1 >$, then there exists a negative regular node n_0 to receive d_i. For n_0 is a negative node, it is at $< s_B, 1 >$ for some BS strand $s_B \in BS[S_j(id), S_i(id), \ldots, \Sigma d_i, counter]$. Since $< s_B, 1 > \Longrightarrow < s_B, 2 >$ and $term(< s_B, 2 >) = S_i(id)\ rand\ S_j(id)\ d_i\ nonce + 1$, we see that $S_i(id) = S_i(id)$, $rand' = rand$, $S_j(id) = S_j(id)$, $d_i = d_i$, $nonce + 1 = nonce + 1$ and the C-height of s_B is 2.

□

Theorem 2 and Theorem 3 prove that the NSDCP2 can ensure the mutual authentication between BS and non-cluster head nodes when $K_i \notin \mathbf{K}_P$. Therefore, The NSDCP2 can withstand all kinds of authentication attack such as impersonation attack and reply attack. Moreover, if the adversaries want to launch the wormhole attack, they cannot forge the right messages to the BS because of the $K_i \notin \mathbf{K}_P$, which means the K_i is not compromised. The BS and non-cluster head nodes will find the wrong messages and abort them, therefore the wormhole attack will be defeated.

6. PERFORMANCE EVALUATION

We conducted a simulation study using NS2 simulator to evaluate the performance of our proposed protocols. We considered the total number of sensor nodes to be $N = 1000$, and the optimal probability of being a CH, $P_{opt} = 0.05$ (50 CHs in the network on average). Based on this assumption, the number of non-CH sensor nodes within each

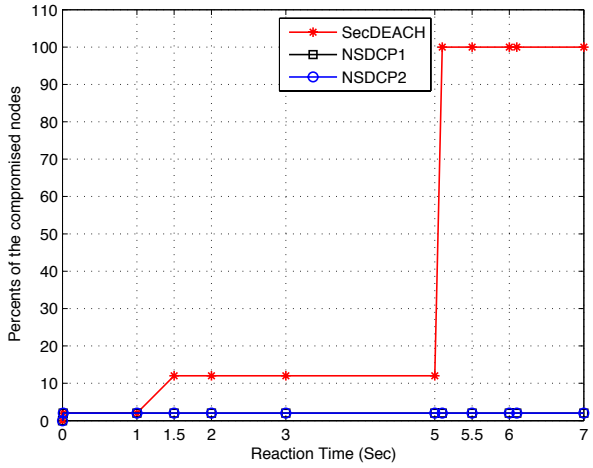

Fig. 8. Nodes controlled by the attacker with the compromised nodes in SecDEACH, NSDCP1 and NSDCP2.

CH is 19. All sensor nodes are randomly distributed in a $100meters \times 100meters$ area with a transmission range of 8 meters.

6.1. The Wormhole Attack

To evaluate the impact of wormhole attacks, we assume the compromised nodes are uniformly distributed over the total area. Specifically, we have one compromised node in every CH region. The goal of this scenario is to evaluate the vulnerability of each protocol in facing a wormhole attack. The simulation results are shown in figure 8.

Figure 8, it demonstrate that the SecDEACH is highly vulnerable to the wormhole attack. If the reaction time is less than $1sec$, all protocols are robust and other uncompromised nodes remain secure. But, when the reaction time increases to $1.5sec$, the SecDEACH protocol reveals its vulnerability and more than 10% of the nodes are getting compromised. This behavior can be observed whenever the reaction time is less than $5.2sec$. Whenever the reaction times passes this cutoff value, all other nodes are getting compromised. On the contrary, the NSDCP1 and the NSDCP2 can withstand the wormhole attack during the simulation study and all other nodes remain uncompromised.

6.2. Fault Tolerance

A compromised node may show a normal behavior similar to a benign node and be elected as the new CH. In such scenarios, the effect of wormhole attack could be severe. Figure 9 demonstrates the resiliency of the NSDCP protocols and SecDEACH protocol in these type of scenarios. This figure shows that the SecDEACH protocol can easily be interrupted. Even if there is a single compromised CH node in the network, all other nodes in the network will be compromised. But, the behavior of the NSDCP1 is linear. In another word, the total number of compromised nodes are linearly proportional to the probability of compromised CHs. The NSDCP2 can withstand the wormhole attack and can protect all the nodes being attacked. As we can see in figure 9, our protocols, especially the NSDCP2, can provide resilient CH election.

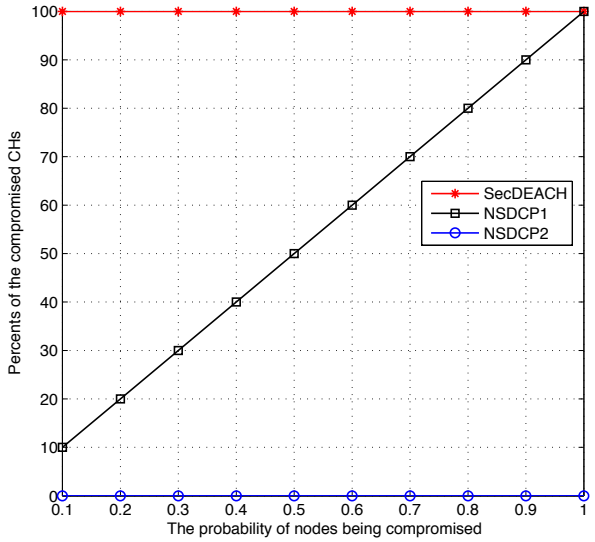

Fig. 9. The number of compromised CHs in SecDEACH, NSDCP1 and NSDCP2.

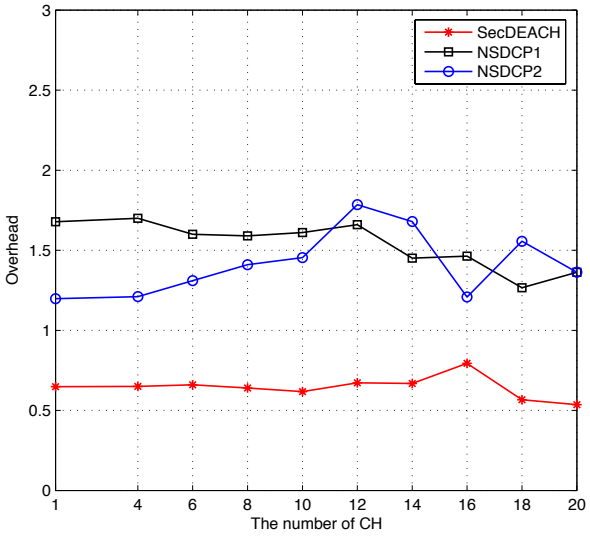

Fig. 10. The overhead of SecDEACH, NSDCP1 and NSDCP2.

6.3. Normalized Overhead

Figure 10 compares the normalized communication overhead of the proposed protocols to the SecDEACH protocol. We define the normalized overhead metric as a total control information *bytes* sent for sending a single *byte* of data. As an example, if the normalized overhead of a protocol is 2, it shows this protocol needs to send in average two

bytes of control data for sending a singe byte of data. In this simulation scenario we consider the $Key_Size = 8Bytes$. This figure shows that for large number of CH, both of the NSDCP protocols have similar overheads. As we can see in figure 10, the NSDCP protocols have more overhead compare to the SecDEACH. However, this overhead is acceptable for a category of sensor applications that are critical but not data-intensive.

7. CONCLUSION

In this paper, we first presented a wormhole attack on the secure and dynamic clustering routing protocol (SecDEACH) and investigated the severe effects of wormhole attacks in sensor networks. In order to withstand the wormhole attack and hold all the merits of the SecDEACH, we proposed two novel secure and dynamic clustering protocols, the NSDCP1 and the NSDCP2, to deal with wormhole attacks. We considered the countermeasure of a wormhole attack in the transmission phase rather than in the secure routing phase. The NSDCP1 is based on the feedback messages from the cluster head, and can be applied to a scenario where the cluster head would not be compromised. The NSDCP2 is based on the feedback messages from the base station, and can be applied in the scenario where the base station is secure and would not be compromised. Based on the Strand Space model and Authentication Tests, We formally analyze the confidentiality and the authentication of our novel protocols, and prove the correctness and the security of our novel protocols. Finally, we conducted a simulation study to compare the performance of the proposed protocols with the SecDEACH, which shown that the proposed protocols hold all the advantages of the SecDEACH and are more secure.

REFERENCES

ALBATH, J. AND MADRIA, S. 2009. Secure hierarchical aggregation in sensor networks. In *Proceedings of IEEE Wireless Communications and Networking Conference*.

BLOM, R. 1985. An optimal class of symmetric key generation systems. In *Advances in Cryptology: Proceedings of EUROCRYPT 84-A Workshop on the Theory and Application of Cryptographic Techniques, Paris, France, April 1984*. Springer, 335.

BLUNDO, C., DE SANTIS, A., HERZBERG, A., KUTTEN, S., VACCARO, U., AND YUNG, M. 1993. Perfectly-secure key distribution for dynamic conferences. In *Advances in cryptologyCRYPTO92*. Springer, 471–486.

CASTELLUCCIA, C., MYKLETUN, E., AND TSUDIK, G. 2005. E. cient aggregation of encrypted data in wireless sensor networks.

DONG, Q. AND LIU, D. 2009. Resilient cluster leader election for wireless sensor networks. In *Sensor, Mesh and Ad Hoc Communications and Networks, 2009. SECON'09. 6th Annual IEEE Communications Society Conference on*. IEEE, 1–9.

FERREIRA, A., VILAÇA, M., OLIVEIRA, L., HABIB, E., WONG, H., AND LOUREIRO, A. 2005. On the security of cluster-based communication protocols for wireless sensor networks. *Networking-ICN 2005*, 449–458.

GUTTMAN, J. AND THAYER, F. 2002. Authentication tests and the structure of bundles* 1. *Theoretical Computer Science 283,* 2, 333–380.

HAN, Y., PARK, M., AND CHUNG, T. 2010. Secdeach: Secure and resilient dynamic clustering protocol preserving data privacy in wsns. *Computational Science and Its Applications–ICCSA 2010*, 142–157.

HU, Y., PERRIG, A., AND JOHNSON, D. 2003. Packet leashes: a defense against wormhole attacks in wireless networks. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*. Vol. 3. Ieee, 1976–1986.

HUANG, S., SHIEH, S., AND TYGAR, J. 2010. Secure encrypted-data aggregation for wireless sensor networks. *Wireless Networks 16,* 4, 915–927.

KARLOF, C. AND WAGNER, D. 2003. Secure routing in wireless sensor networks: Attacks and countermeasures. *Ad hoc networks 1,* 2-3, 293–315.

KHABBAZIAN, M., MERCIER, H., AND BHARGAVA, V. 2009. Severity analysis and countermeasure for the wormhole attack in wireless ad hoc networks. *Wireless Communications, IEEE Transactions on 8,* 2, 736–745.

KUMAR, V. AND MADRIA, S. 2010. Secure data aggregation in wireless sensor networks. *Wireless Sensor Network Technologies for the Information Explosion Era*, 77–107.

LIU, S. AND JIANG, R. 2011. Security analysis and improvement of secure and dynamic clustering protocol. *Applied Mechanics and Materials 48*, 1014–1017.

OLIVEIRA, L., FERREIRA, A., VILAÇA, M., WONG, H., BERN, M., DAHAB, R., AND LOUREIRO, A. 2007. Secleach–on the security of clustered sensor networks. *Signal Processing 87,* 12, 2882–2895.

SIMULATOR, N. 2005. 2 (ns2). *URL: http://www. isi. edu/nsnam*.

SIRIVIANOS, M., WESTHOFF, D., ARMKNECHT, F., AND GIRAO, J. 2007. Non-manipulable aggregator node election protocols for wireless sensor networks. In *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks and Workshops, 2007. WiOpt 2007. 5th International Symposium on*. IEEE, 1–10.

THAYER, F., HERZOG, J., AND GUTTMAN, J. 1999. Strand spaces: Proving security protocols correct. *Journal of Computer Security 7,* 2/3, 191–230.