http://dx.doi.org/10.14400/JDPM.2013.11.12.367

Secure Data Gathering Protocol over Wireless Sensor Network

Hae-Won Choi[*], Myung-Chun Ryoo[*], Chae-Soo Lee[**], Hyun-Sung Kim[***]

Dept. of Computer Engineering Kyungwoon University[*]

Dept. of Mobile Engineering Kyungwoon University[**]

Dept. of Computer Engineering Kyungil University[***]

무선센서네트워크에서 안전한 데이터 수집 프로토콜

최해원[*], 류명춘[*], 이채수[**], 김현성[***]

경운대학교 컴퓨터공학과[*], 경운대학교 모바일 공학과[**], 경일대학교 컴퓨터공학과[***]

요 약 무선센서 네트워크에서 안전한 데이터 수집은 중요한 보안이슈 중에 하나이다. 일반적으로 안전한 데이터 수집이란 데이터 자체의 보안과 안전한 전송 경로 확보를 의미한다. 본 논문은 이와 같은 관점에서 무선센서 네트워크에서 안전한 데이터 수집 프로토콜을 제안한다. 제안하는 프로토콜은 계층형 센서네트워크를 고려한 계층형 키 보안 기법과 안전을 보장할 수 있는 전송경로 설정을 핵심적으로 제시한다. 프로토콜은 네트워크 부하를 최소화 할 수 있도록 최적화 되었으며 네트워크 공격으로 인해 발생하는 문제점을 효과적으로 차단한다. 성능평가 결과 제안하는 프로토콜은 네트워크 퍼포먼스를 고려한 데이터 수집에 효율적이다. 데이터 수집 시 안전을 확보하기 위한 보안 분석 역시 검증해 보았다.

주제어 : 데이터 수집 프로토콜, 네트워크 보안, 센서네트워크, 클러스터 프로토콜

Abstract A secure data gathering in a Wireless Sensor Network(WSN) has given attention to one of security issues. In general, the process of secure data gathering causes difficulties: one process is exchanging the secured data and the other is constructing secured data path. The previous studies have been resolving the difficulties in terms of two problems: security and data gathering in WSNs. However, a WSN requires a protocol that has to guarantee a security of path between sensors and sink, or a cluster head. Thus how to gather data securely is an important issue. In this paper, we propose a secure data gathering protocol over WSNs, which consists of hierarchical key settlement and secure path construction, and aims at tackling two problems. The proposed protocol causes little overhead to sensor nodes for secured key settlement and path construction. This work provides security analysis focused on the key settlement protocol and evaluates network performance for the proposed data gathering protocol through simulation.

Key Words : Network Security, Cluster Protocol, Data Gathering, Wireless Sensor Network

Received 23 October 2013, Revised 16 November 2013
Accepted 20 December 2013
Corresponding Author: Hyun-Sung Kim(Kyungil University)
Email: kim@kiu.ac.kr

ISSN: 1738-1916

1. Introduction

Wireless Sensor Networks(WSN) have been widely applied in applications for health, military, home, commercial industries, etc. These applications periodically need data upcoming from sensors [1, 2]. In a hierarchical WSN, a sensory data is periodically gathered in cluster head and then forwarded to the sink. This method for collecting data makes them very vulnerable to adversary's malicious attacks [3]. The security threats to a WSN are different from the threats of the other wireless networks due to the broadcast nature. In practical, sensor nodes are very fragile since the data may also be physically captured or easily destroyed by the attackers. This is because WSNs are consists of large numbers of wirelessly connected heterogeneous sensor nodes, and which are spatially distributed across a large field of interest [4]. If, for example, a sensor node on a path is compromised by an attacker, illegal nodes, such as the node with detouring routing paths or with depleted energy, in a routing path will be included. This means that the data requires to be passed a slightly longer path to reach at destination for includes compromised information. A case in point, an adversary can physically compromise a subset of sensor nodes in a WSN to eavesdrop information [5].

Research into security and routing confidentiality mechanisms designed specifically for sensor data has been challenging, thus such tackled protocol usually has been focused into an avoidance- and a detection-based study that are defense method and pre-computation, respectively, against attackers. Although the avoidance methods are more powerful than the detection, it is hard to apply sensor domains due to the limited hardware capability; in most cases they need pre-computation processes. There was a new security challenge to cope with, named multiple-stage attack. Attack with multiple compromised nodes is a typical multiple-stage attack to acquire network information and transmit false data. The multiple-stage attack is generally composed of three stages. First, an adversary captures some sensor nodes, and then compromises these nodes using various tools, such as exploits; and then, these compromised nodes are redeployed into the network; lastly, the adversary can use these compromised nodes to attempt various security attacks, such as false data injection, selective forwarding, wormholes and Sybil attacks [6], to jeopardize the whole networks. Since sensor nodes are randomly deployed and unattended, nodes easily are compromised. When an adversary launches a compromise attack, the adversary can simply use a programming board and a serial cable to easily compromise the sensor node in the first stage. The detection of compromised nodes in this case is very difficult task. So, there are some avoidance methods to deal with the attacks using compromised nodes [7]. Sang et al. in [8] addressed that security protocol using a globally shared key on link layer is completely ineffective in presence of insider attacks or compromised nodes [9]. Secure routing methods in [10] for WSN applications are proposed, but none of them are designed with the security considerations, especially focused in terms of compromised nodes.

In order to solve problems related to multiple compromised nodes, this paper proposes a Secure Data Gathering Protocol(SDGP) over WSN, which is consists of three phases: hierarchical key settlement, secure path construction, and secure path based data gathering. Hierarchical key settlement phase, at first, is for setting up network parameters and key generation procedure. And then a secure path construction phase is to establish multiple secure paths between cluster heads and sink node by using data encryption and message digest to support confidentiality and integrity, based on the established keys from the key settlement. Finally, secure path based data gathering phase is to collect data in WSN by relaying data packets from the source node to the sink based on the multiple secure

paths. The contributions of this paper are as follows:

First, it identifies the security problem which is occurred by multiple compromised nodes in WSNs. Second, it effectively proposes a secure data gathering protocol through the hierarchical non-interactive identity-based authenticated key agreement, and gives a clue of problems related to multiple compromised nodes. Lastly, it provides an analytical model for evaluating a network security performance from the compromised nodes.

The paper is organized as follows: in section 2, we briefly review related works in WSNs. In section 3, we propose a secure data gathering protocol. Security analyses and performance evaluations are following in Sections 4 and 5, respectively. We conclude the paper in Section 6.

2. Related Works

Multipath routing in WSNs have received considerable attentions and have used for different goals, such as load balancing, energy efficiency, security, and so on. However, multipath routing has problems focused on the encoding issue and key agreement for resolving security problem and data gathering for load balancing and energy efficiency [1-2].

Packet encoding and key agreement issues in WSN have been intensively researched to improve security in data collection. Shamir [17] proposed an algorithm to break a data packet into a few shares by using the (t, n)-threshold secret sharing algorithm and then deliver shares via different routing paths. A packet is broken into shares, which are sent to sink through randomly generated paths. However, the algorithm looks like it could effectively keep data in safe from the compromised nodes but it has security problem when they could collect t data shares, which could form the original data from the shares. The secret-sharing and routing parameters in [11] are optimized to minimize the energy cost for a given packet delivery probability constraint. Recently, Yuxin *et al.* [3] proposed a Feedback-Based Secure Path approach (FBSP) based on Shamir's algorithm to establish multiple secure paths. The secure paths in the FBSP are not always secure and furthermore, the approach shares the security problems in Simmer's algorithm.

To achieve secure routing in WSNs, Shamir's algorithm was applied to ad hoc on-demand multipath distance vector routing in [12] and data are split into multiple shares and transmitted to multipath by using maximally disjoint paths in [13]. A non-interactive hierarchical key agreement protocol using Pairings in [14] is similar to other key agreement protocols for authentication. Bilinear map captures an important cryptographic problem, i.e., the Bilinear Diffie-Hellman (BDH) problem, which was introduced by Boneh and Franklin in [15]. The authors in [16] addressed that cryptographic operation is required while any attacking challenge is detected, and proposed the secure aggregation tree that is built in such a way that the child is able to listen all the incoming data from its sibling to its father so that the child nodes can observe the behavior of its father, then the cheating activity of any non-leaf node can be detected. However, it does not provide data confidentiality. Ozdemir [7] proposed, to develop trustworthiness for environments and neighboring nodes, a secure and reliable data aggregation in WSNs. If an aggregating node is under attack (e.g. denial of service attack), it is detected by using the monitoring system. This approach provides data integrity and source authentication, but does not provide data confidentiality.

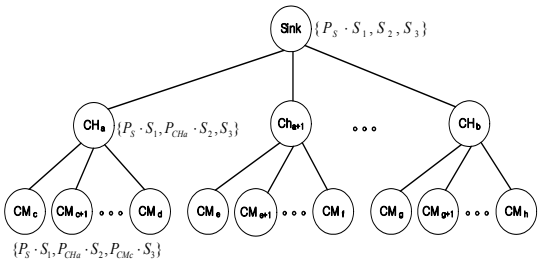
3. Secure Data Gathering Protocol

This section presents a Secure Data Gathering Protocol(SDGP) with a hierarchical structure, which is

composed of two phases: hierarchical key settlement and secure path construction, which are referred to as the Phases I and II respectively. The Phase I is for setting up network parameters and key generation procedure. Multiple secure paths between source nodes toward sink are established in the Phase II by using data encryption and the message digest based on the established keys from the key settlement. In this paper, the proposed protocol assumes a hierarchical WSN, which is represented to a hierarchical tree with depth 3. The degrees of the sink and cluster heads are j and k, respectively, which are determined by the number of nodes, n, in a WSN.

3.1 Phase I: Hierarchical Key Settlement

In the Phase I, in order to establish a shared key between two nodes in a WSN, we use the pre-established secret keys method for the simplicity to establish credential between two entities. After this phase, all nodes have a pair of keys, public key and private key, for the public key cryptosystem. In general, all sensor nodes in WSN belong to the same administrative entity, such as the sink (or cluster head from member sensor nodes point of view), but the Internet has multi-administrative entities.



[Fig. 1] Hierarchical key settlement for hierarchical WSN

In security applications for WSNs, it is reasonable that sensor nodes in the initial state know each other, and thus trust as they can exchange information in the plaintext mode. Fig. 1 shows a procedure of a hierarchical key settlement for hierarchical WSN. In Fig. 1, the sink, cluster head, and cluster member node are denoted as the black colored square and circle and the white colored circle, respectively.

Phase I uses an identity of each node ID_S for the public key, which is based on the ID-based cryptosystem. Sink, as an administrator that is responsible for distributing a security key, performs key generation by starting setups with its private key set, denoted as $\{S_1, S_2, S_3\}$ where S_i $(1 \le i \le 3)$ are from Z_q^*. We in this paper assume that each entity plays a different role in a network; for example, sink has more abilities, such as setting up keys and gathering data over a WSN, cluster heads have neutral abilities, such as governing sensor nodes deployed in their realm, and sensor nodes have least abilities, such as sensing. The privilege of each entity is determined with the degree of how many elements does the entity have from the private key set of the sink. To set up keys, Phase I performs the following operations.

Step 1. The sink with identities ID_S creates it's private key set, $\{S_1, S_2, S_3\} \in Z_q^*$, for a WSN and computes $P_S=h(ID_S)$ and $P_S \bullet S_1$, where $h(\bullet)$ is a one-way hash function. After that, sink stores the information in it's memory and sends $\{\{P_S \bullet S_1, S_2, S_3\}, P_S\}$ to cluster heads.

Step 2. When a cluster head with identities ID_{CH} receives the message, it computes $P_{CH}=h(ID_{CH})$ and $P_{CH} \bullet S_2$. After that, it stores the information in it's memory and sends $\{\{P_S \bullet S_1, P_{CH} \bullet S_2, S_3\}, P_S, P_{CH}\}$ to it's member nodes.

Step 3. When a member node with identities ID_{CM} receives the message, it computes $P_{CM}=h(ID_{CM})$ and $P_{CM} \bullet S_3$ and then member nodes store the computed results in it's memory.

The proposed key settlement phase has a good advantage in the perspective of leaf nodes. These nodes

do not need any other information to set up session keys with their ancestor nodes in Fig. 1 because they already have the required information after Phase I. Thereby, they do not need to have communication with the counterpart node to set up session key. It is very important aspect in WSN due to their limitations.

〈Table 1〉 Notations

Symbol	Description
ID_i	Identities of entity i
$\{S_1,S_2,S_3\}$	Set of private key for sink node, $S_i \in Z_q^*$
$h(\cdot)$	One way hash function $h:\{0,1\}^* \rightarrow Z_q$
$\cdot$	Multiplication operation
G	Additive group of prime order q
G_T	Multiplicative group of prime order q
ê	bilinear map $\hat{e}:G \times G \rightarrow G_T$
P_i	Amplified identities by applying $h(ID_i)$
sk	Session key established between two entities
L_R	Path's set of identities
C	Path's credential level
α	Permitted transmission delay

3.2 Phase II: Secure Path Construction

The purpose of SDGP is to reduce the probability that the compromised nodes are in the secure paths and also to be not effected on the confidentiality of data even if they are in the paths. This phase uses encryption/decryption, *MAC*, and (t, n) threshold mechanism based on the keys from the Phase I. Notations used in the Phase II are listed in Table 1.

The goal of this phase for the SDGP is to establish multiple secure paths between source nodes and sink using the settled keys in Phase I. Encrypted and fragmented data are transmitted to the sink over the secured multi paths. After processing this Phase II, the sink would determine secure paths. The Phase II for constructing secure path is shown in Fig. 2. The symbols of the sink, cluster heads, and member nodes use identical symbols with those of Fig. 1, and it operates to as follows:

Step 1. A member node CM_k sends an encrypted data packet with it's identity set $\{P_S', P_{CHj}', P_{CMk}'\}$ and the message digest $MAC=h(sk||$the encrypted data packet) to it's cluster head CH_j, which is encrypted by using a session key sk in the Phase I. The sk of CM_k is computed as $sk=\hat{e}(P_S \cdot S_1, P_S') \cdot \hat{e}(P_{CHj} \cdot S_2, P_{CHj}') \cdot \hat{e}(P_{CMk} \cdot S_3, P_{CHj}')$, which includes it's own identity set $\{P_S', P_{CHj}', P_{CMk}'\}$.

Step 2. If two nodes, CM_k and CH_j, want to establish a secure channel, they need to perform key agreement. After receiving the encrypted data, the cluster head CH_j checks the validity of *MAC*. The shared session key between CM_k and CH_j should be the same because $sk=\hat{e}(P_S \cdot S_1, P_S') \cdot \hat{e}(P_{CHj} \cdot S_2, P_{CHj}') \cdot \hat{e}(P_{CHj} \cdot S_3, P_{CMk}')=\hat{e}(P_S, P_S' \cdot S_1) \cdot \hat{e}(P_{CHj}, P_{CHj}' \cdot S_2) \cdot \hat{e}(P_{CHj}, P_{CMk}' \cdot S_3)=\hat{e}(P_S, P_S')^{S1} \cdot \hat{e}(P_{CHj}, P_{CHj}')S_2 \cdot \hat{e}(P_{CHj}, P_{CMk}')^{S3}$. Only if the MAC validation is successful, CH_j breaks the data into n shares according to the (t, n)-threshold algorithm, computes $MAC=h(sk||$the data share) for each share, and forwards them to the neighbor heads after it adds its identity CH_j to L_R.

Step 3. When a cluster head CH_m receives a share, it adds its identity CH_m to L_R. The share of data is forwarded by a collection of relay heads until it reaches sink.

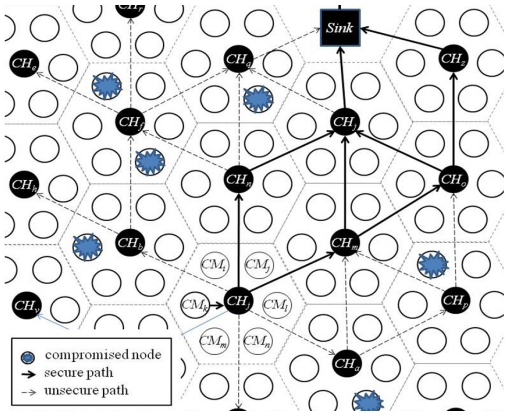
Step 4. On the arrival of the share, sink decrypts the share by establishing $sk=\hat{e}(P_S, P_S' \cdot S_1) \cdot \hat{e}(P_S, P_{CHj}' \cdot S_2) \cdot \hat{e}(P_S, P_{CHj}' \cdot S_3)$ with the source node using it's private key and the received identity set $\{P_S', P_{CHj}', P_{CMk}'\}$. When sink successfully finishes the check of *MAC*, it extracts $L_R=\{CH_j, CH_m, \cdots, CH_z\}$ from the share and stores it in its local database. Here, L_R is called a secure path.

Step 5. Sink adds the secure path L_R to a notification packet and sends the packet with the trust

value C to the source node by using the route in L_R. Here, C is used as a counter with an initial value t (t>0), which represents credibility level of the route. The notification packet contains the secure path for data collection and the credibility of the path, C.

Step 6. When a cluster head CH_o receives the packet, it extracts a sub-path Po={ CH_{o+1}, CH_{o+2}, …, CH_n } from L_R and stores it into its local cache only if its identity is within L_R. CH_o extracts its next-hop cluster head CH_{o-1} from L_R and forwards the packet to the head.

Step 7. When the cluster head CH_j receives the packet, it extracts L_R from the packet, and stores it in its local cache.



[Fig. 2] Secure path construction

Fig. 2 exemplifies establishing secure paths from CH_j to the sink using Phase II. It is assumed that CM_b and CH_r are compromised by the adversary mentioned in the above. When CH_j receives the encrypted message with MAC from it's member node, CH_j validates the message digest and divides the data into shares only if the validation is successful, adds it's ID into L_R and the message digest MAC for each share, and broadcasts the shares to CH_m, CH_k, and CH_r. After receiving the share, each cluster head adds it's ID into L_R, and re-broadcasts the shares to neighbor cluster heads. Finally, the sink receives the shares from various paths and gets L_R. Note that the sink will drop the share if it arrives beyond the delay threshold or the validation of MAC is failed. After deciding the credibility level of multi-paths and setting value C, the sink feedbacks C and L_R by following the path stored in L_R.

3.3 Phase III: Secure-path based Data Gathering

This sub-section proposes a secure-path based data gathering scheme by using the multi-path from the Phase II. In this Phase III, each cluster head keeps the secure paths information in their local database and selects a path with the highest value from C among multiple secure paths. We use a (t, n) threshold mechanism to encode a sensed data packet. When a sensor node wants to send a packet to a destination node, it first breaks the packet into m shares according to the threshold mechanism. Each share is then encrypted and transmitted to the neighbor from the multi-path. The overall steps for setting up the secure-path based data gathering are as follows.

Step 1. When a source member node CH_k intends to send a data share from the (t, n) threshold mechanism to the sink, it first checks whether the application requires data aggregation at the cluster head or not.

- If it does not require data aggregation, then CH_k generates a session key sk with the sink and sends the required data and ID set related information to CH_j after encrypting the data with the key sk and adding the digest MAC=h(sk||the encrypted data). For this, it performs the following sub steps.

- CH_k generates a session key sk=$\hat{e}(P_S \cdot S_1, P_S')$ • $\hat{e}(P_{CHj} \cdot S_2, P_S')$ • $\hat{e}(P_{CMk} \cdot S_3, P_S')$ with the sink, where $\{P_S', P_{CHj}', P_{CMk}'\}$ is CM_k's own

ID set.

- CM_k encrypts the data with sk and sends the encrypted data with the *ID* set $\{P_S', P_{CHj}', P_{CMk}'\}$ to CH_j.

• Else if it requires data aggregation, some additional steps from the case above are necessary to be involved for safely transferred to the cluster head. CM_k should first send encrypted data to CH_j. Then, CH_j needs to generate a new session key *sk* with the sink, encrypts the aggregated data with the session key, attaches *MAC* of the encrypted data with the session key, and sends the encrypted data with the ID set related information to the sink. To perform this, they perform the following steps

- CM_k generates a session key $sk=\hat{e}(P_S \cdot S_1, P_S') \cdot \hat{e}(P_{CHj} \cdot S_2, P_{CHj}') \cdot \hat{e}(P_{CMk} \cdot S_3, P_{CHj}')$ with CH_j, where $\{P_S', P_{CHj}', P_{CMk}'\}$ is CM_k's own ID set.
- CM_k encrypts the data with sk, attaches $MAC=h(sk||$the encrypted data$)$ to the message, and sends the message with the ID set $\{P_S', P_{CHj}', P_{CMk}'\}$ to CH_j.
- After receiving the encrypted data from CM_k, CH_j generates the session key $sk=\hat{e}(P_S', P_S \cdot S_1) \cdot \hat{e}(P_{CHj}', P_{CHj} \cdot S2) \cdot \hat{e}(P_{CMk}', P_{CHj})^{S3}$ for CM_k, where $\{P_S, P_{CHj}\}$ is it's own ID set and $\{P_S', P_{CHj}', P_{CMk}'\}$ is the received ID set.
- After verifies *MAC* and collects whole data from member nodes, CH_j performs data aggregation, generates a new session key $sk=\hat{e}(P_S \cdot S_1, P_S) \cdot \hat{e}(P_{CHj} \cdot S_2, P_S) \cdot \hat{e}(P_{CHj}, P_S)^{S3}$ for the sink, encrypts the data with the generated *sk* and attaches *MAC* to the message.

Step 2. When CH_j intends to send the message to the sink, it first checks its local cache. If there are secure paths, it selects a secure path $P=<CH_j, L_R, C>$ with the largest value *C* from its local data repository. CH_j attaches $L_R=\{CH_m, CH_{m+1}, \cdots, CH_z\}$ to the head of the data share. If there are no secure paths in the local cache of the relay nodes, it just performs the secure path construction phase as in Phase II.

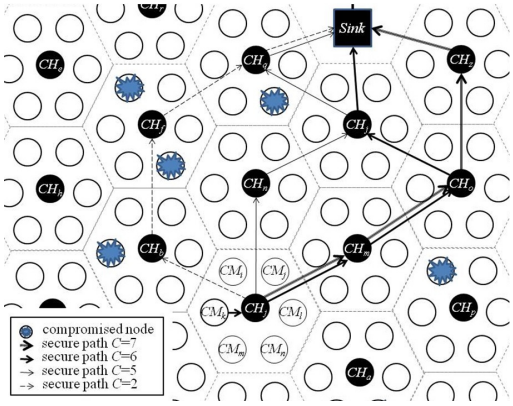
Step 3. When a cluster head CH_i receives the share, it first checks if the cluster head CH_{i+1} in L_R is in its neighbor list. If the head is not in the list, it just performs random multipath routing and path construction. Otherwise, it sends the share to the head CH_{i+1} in L_R.

Step 4. On the arrival of the share to the sink successfully, the sink generates a session key sk with CM_k or CH_j depending on the included ID set in the message. If the ID set is $\{P_S', P_{CHj}', P_{CMk}'\}$ from CM_k, it generates $sk=\hat{e}(P_S \cdot S_1, P_S) \cdot \hat{e}(P_{CHj}', P_S)S_2 \cdot \hat{e}(PCMk', P_S)S_3$. Otherwise, it generates $sk=\hat{e}(P_S \cdot S_1, P_S) \cdot \hat{e}(P_{CHj} \cdot S_2, P_S) \cdot \hat{e}(P_{CHj}, P_S)S_3$ by using the ID set $\{P_S, P_{CHj}\}$ from CH_j. After that, the sink validates *MAC* of each share, decrypts the message and extracts $L_R=\{CH_j, CH_m, CH_{m+1}, \cdots, CH_z\}$ only if the validations are successful. If there is a secure path in the share, it means every relay cluster heads have used the path and the sink just sends back an empty notification to CM_k or CH_j. Otherwise, the sink extracts the identity set as a new secure path from the share, updates its local database, and sends back a notification with the newly-constructed secure path to CM_kor CH_j. The relay cluster heads on the path update their local cache with the sub-paths. On the arrival of the notification, CM_k and CH_j extract new secure path from the packet, and store it in its local cache. If the share is dropped or does not reach to the sink within the allowed time span, CM_k or CH_j does not receive a notification from the sink, and then it just decreases the credential counter *C* by 1 of the path. If the counter of a secure path is cleared, each node will remove it from its local cache.

The secure-path based data gathering phase performs differently based on the application whether it requires data aggregation or not. If the application does not require data aggregation, CM_k establishes a session key sk with the sink and sends the encrypted data with the message authentication code to the sink by following the pre-established secure paths. Otherwise, CM_k requires additional processing, which needs more operations through it's cluster head. CM_k should first establish a secure channel with CH_j and then send the data with the message digest. Then, CH_j also needs to establish secure channel with the sink using a new session key sk with the sink.

Fig. 3 exemplifies a process of the secure-path based data gathering from CM_a to the sink, which uses the multi-secure paths established in Phase II. CM_a establishes a session key sk with CH_j or the sink depending on the application requires data aggregation, encrypts data using *sk*, adds the message authentication code *MAC*, and sends the message to CH_j. In the case that the application requires data aggregation, CH_j requires additional operations that it decrypts and collects data from it's member nodes, and after that it aggregates the collected data and encrypts the data by establishing a new session key sk with the sink. Otherwise, CH_j sends the encrypted data with the *MAC* to the sink by using the secure paths stored in its local cache. When a cluster head CHi receives the share, it also checks its local cache to find out a secure path and sends the share to the CH_{i+1} in L_R bounded to the sink. As soon as the share arrives, it manages the credential counter C depending on the credibility.

By using these procedures, SDGP could safely cope with the attacks from the multiple compromised nodes even if more than t data shares are exposed to the compromised nodes. Furthermore, to reduce the probability that paths contain the compromised nodes, the sink assigns path's credibility level C ($C>\alpha$) depending on the transmission delay, where α is the permitted transmission delay determined by considering the network characteristics. This means that a path with longer latency than the other paths considers that the path has the compromised nodes and in that case, SDGP drops the path because there is high possibility that the transmission delay could be increased due to the interrupts for the compromised nodes. If the share is arrived in a certain delay threshold, the sink extracts the path and assigns the proper path credibility level C. The secure path is set C depending on the credibility level, which means that higher value represents more secure than the lower value. The credibility level could be synchronized between sink and every cluster heads in the feed-back steps.



[Fig. 3] Secure path based data collection

4. Security Analysis

Since the core mechanism in SDGP is based on the proposed hierarchical key settlement described in Section 3, this Section provides the security analysis for the proposed hierarchical key settlement. Until now, a simple, efficient and convincing formal methodology for correctness analysis on security protocols is still an important subject of research and, moreover, an open problem. Because of these reasons, most security protocols have been demonstrated with a simple proof.

Therefore, we follow the approaches used in [18] for comparison purpose. This Section first gives insights about the computational problems and security analysis in sub-sections 4.1 and 4.2, respectively, for SDGP. Sub-section 4.3 investigates a security operation overhead whether they are vulnerable to types of various attacks or not and shows the compared analysis results in terms of security operation and security cost.

4.1 Computational Problems

Bilinear map captures an important cryptographic problem, i.e., the Bilinear Diffie-Hellman (BDH) problem, which was introduced by Boneh and Franklin in [15]. The security of SDGP relies on a variant of the BDH assumption.

Let G and G_T be two groups of a prime order q. Suppose that there exists a bilinear map $\hat{e}$: $G \times G \rightarrow G_T$. We consider the following computational assumptions

- Bilinear Diffie-Hellman (BDH) : For a, b, and $c \in {}_R Z_q^*$ and given aP, bP, and cP, computing $\hat{e}(P, P)^{abc}$ is hard.
- Decisional Bilinear Diffie-Hellman (DBDH) : For a, b, and $c \in {}_R Z_q^*$, differentiating $(aP, bP, cP, \hat{e}(P, P)^{abc})$ and $(aP, bP, cP, \hat{e}(P, P)^{r})$ is hard.

4.2 Security Analysis

Our security analysis is focused on verifying the overall security requirements for the proposed hierarchical key settlement including passive and active attacks as follows.

Proposition 1. The proposed hierarchical key settlement is secure against passive and active attacks.

Proof: Assuming that an adversary is success if the adversary could learn some useful information from the intercepted messages. We show that probability to succeed in learning them is negligible due to the difficulties of the underlying cryptosystem, the BDH problem, and the DBDH problem.

1. A completeness of the key agreement protocol is already proven by describing the run of the protocol in section 3.

2. If the adversary is passive adversary, all information the adversary can gather are as follows: the ID set $\{P_S', P_{CHj}', P_{CMk}'\}$, the path set $L_R = \{CH_m, CH_{m+1}, \cdots, CH_z\}$, and the message digest MAC. However, it is negligible to find the key related information from them due to the difficulties of the underlying cryptosystem, the BDH problem, and the DBDH problem.

Finally, we could say our protocol is secure against passive attack.

Proposition 2. The proposed hierarchical key settlement is secure against active attack.

Proof: Assuming that an adversary is success if the adversary finds the session key sk or the session key related information $\{S_1, S_2, S_3\}$. Therefore, we show that probability to succeed in finding them is negligible due to the difficulties of the underlying cryptosystem, the BDH problem, and the DBDH problem.

1. The acceptance by all entities means that each MAC in the corresponding message is successfully verified. That is, MAC is decrypted and verified successfully by using the correct session key sk. We show that if it is the case that entities accept the messages and continue the session, then the probability that the adversary has modified the messages being transmitted is negligible. And the only way for the adversary to find the session key or security related information is to solve the difficulties of the underlying cryptosystem, the BDH problem, and the DBDH problem.

2. Now, we consider the active adversary with following cases.

(a) There is no way that an adversary could get the

secret information $\{S_1, S_2, S_3\}$ due to the difficulties of the BDH problem and the DBDH problem.

(b) An adversary cannot impersonate CM_i or CH_i to cheat the sink. That is due to the attacker cannot generate valid messages without deriving the correct session key sk, since the attacker cannot pass the verification of MAC in the protocol.

(c) An adversary cannot impersonate the sink to cheat CM_i or CH_i. As described in the above, only the legal sink can form the legal messages by including the proper check sum, which needs to be properly matched with the information from CM_i or CH_i in the protocol steps. Even if the attacker could pass the verifications at the protocol steps, the attacker still cannot get any useful information from the encrypted messages due to the difficulty of the underlying public key cryptosystem and cannot generate the consequent valid messages.

Finally, we could say our protocol is secure against active attack.

4.3 Security Operation Overhead

Security operation in SDGP is affected to the overall performance of the target network. In this subsection, we will compare SDGP with FBSP in the perspective of security [3]. For the convenience of evaluating the computational cost of security related operation only, we define some notations as the same as in [19] as follows.

- TG_e : the time of executing a bilinear map operation
- TG_{mul} : the time of executing a scalar multiplication operation of point
- T_{hash} : the time of executing a hash function
- $TS_{e/d}$: the time of executing a symmetric encryption or decryption
- T_{th} : the time of executing a (t, n) threshold mechanism

In Table 2, we summarize the security operation evaluation of SDGP. The advantage of SDGP compared to the other secure routing protocols is it does not require any additional message exchanges to establish session keys to secure their communication channel. In Table 2, we know that each side requires about $6TG_e+5TG_{mul}+mTS_{e/d}+mT_{hash}$ operations except the sink with additional $1T_{th}$ for the (t, n) threshold mechanism.

〈Table 2〉 Security operation evaluation of SDGP

Phases		Member Node CM_i	Cluster Head of CM_i	Other Cluster Heads	Sink
Phase I		$TG_{mul}+T_{hash}$	$TG_{mul}+T_{hash}$	$TG_{mul}+T_{hash}$	$TG_{mul}+T_{hash}$
Phase II		$3TG_e+2TG_{mul}+T_{hash}$	$3TG_e+2TG_{mul}+T_{hash}$	–	$3TG_e+2TG_{mul}+T_{hash}$
Phase III	aggregation (X)	$3TG_e+2TG_{mul}+mTS_{e/d}+mT_{hash}$	–	–	$3TG_e+2TG_{mul}+mTS_{e/d}+mT_{hash}+T_{th}$
	aggregation (O)		$3TG_e+2TG_{mul}+mTS_{e/d}+mT_{hash}$		
Total		$6TG_e+5TG_{mul}+mTS_{e/d}+(m+2)T_{hash}$	$6TG_e+5TG_{mul}+mTS_{e/d}+(m+2)T_{hash}$	$TG_{mul}+T_{hash}$	$6TG_e+5TG_{mul}+mTS_{e/d}+(m+2)T_{hash}+T_{th}$

In Table 3, we demonstrate the security comparisons between SDGP and FBSP in terms of security operation requirements and various security aspects. It is well known that the pairing operation is currently inefficient and cost-consuming but there are recent tendencies to adopt the pairing operation over WSNs due to the security reasons [20-23]. We can conclude SDGP supports better security than FBSP but requires more operations due to the security reasons. Security of data in FBSP is only depending on the (t, n) threshold mechanism. Thereby, FBSP could not support confidentiality, integrity and privacy and unsecure against colluding attack. However, SDGP could solve the problems in FBSP by additionally adopting encryption to the data, which is actually to support

confidentiality and cope with against colluding attack. Furthermore, SDGP could support partial privacy by using amplified identities in LR instead of identities in Phases II and III, respectively.

〈Table 3〉 Security comparisons between SDGP and FBSP

Properties	Value	Value
Security computational cost (member or head)	-	$6TG_e+5TG_{mul}+mTS_{e/d}+(m+2)T_{hash}$
Security computational cost (sink)	T_{th}	$6TG_e+5TG_{mul}+mTS_{e/d}+(m+2)T_{hash}+T_{th}$
Confidentiality	Not support	Support
Integrity	Not support	Support
Privacy	Not support	Partially support
Colluding attack	Not secure	Secure

5. Performance Evaluations

To validate the proposed protocol, this section assumes that compromised nodes on network layer attempts an attack that drops or retransmits packets. We provide comparisons between the proposed protocol, SDGP and feedback-based secure path approach (FBSP) from [3] in terms of a ratio of receiving packets and energy consumption.

5.1 Metrics and Simulation Setting

We provide a comprehensive analysis of the performance of SDGP through extensive simulation. In this simulation, WSNs are composed of five clusters; each cluster has 9 member nodes. Assuming that a transmit range of cluster head is 40m and the distance between cluster heads and sink which is located in center is 140m. Member sensors of a cluster transmit a sensed data to its cluster head and the cluster head again transmits aggregated data to sink or near cluster head through using multipath. Compromised nodes randomly deployed on the multi-path. Each simulation is repeated 100 times and the average results are plotted. Parameters used in the simulation are listed in Table 4.

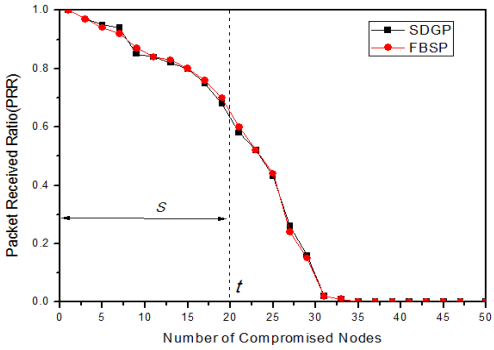
〈Table 4〉 Simulation parameters

Parameter	Value
Size of WSN	$100m \times 100m$
Number of sensor nodes(n)	100
Number of sensor nodes(n)	$n \times 10\%$
Position of sink	(50, 50)
Beginning of energy	1 J
Packet size(L)	2,000 bit
Amplification mode power	$0.1nJ/bit/m^2$
Idle mode power(E_{elec})	$50nJ/bit$
Receive mode power	$50nJ/bit$
Transmit mode power	$50nJ/bit$
Transmission range between head and head	$140m$
Transmission range between head and sink	$140m$
Transmission range between head and node	$40m$
Transmission collision ratio	20%
Simulation time(1 round)	5 second

5.2 Packet Receiving Ratio and Energy Consumption

This sub-section shows a comparison between SDGP and FBSP in terms of packet receiving ratio. In case of FBSP, the ratio of receiving packet, denoted as ε, increases, because all packets transmitted by compromised nodes for attack is received, but in our scheme the ratio of receiving packet is lower than that of FBSP. We conduct our simulation in the environment listed above. Packets arrival time follows the Poission distribution, which is 0.001 to 3 seconds, and assumed that in the Poission distribution the ratio of packet collision is 20% of transmitted packet. Packet receiving ratio, denoted as PRR, is an important factor in secure data gathering protocols, which is suffered from compromised nodes. This problem leads to resulting in a decrease of PRR. In this experiment, PRR can be obtained as follows.

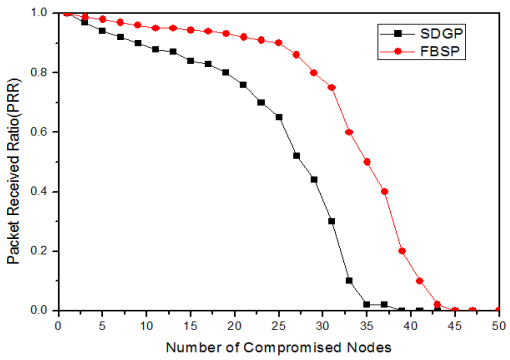
$$PRR = \frac{The\ number\ of\ packets\ received}{The\ number\ of\ packets\ tranmitted}$$



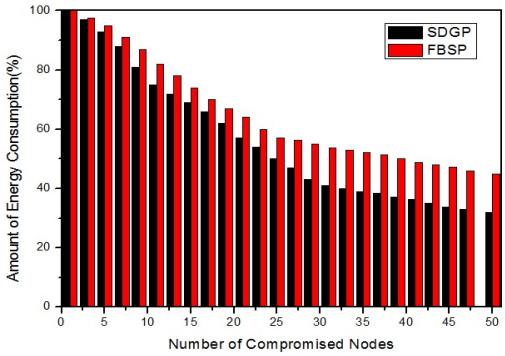
[Fig. 4] Performance of PRR with packet drop by compromised node

Fig. 4 shows the effect of increasing the number of compromised nodes on the PRR. The drop ratio of message is plotted, as shown in Fig. 4, when the number of compromised nodes increases. In both FBSP and SDGP, PRR decreases in case of increasing the number of message. This is because the number of multi-secure paths decreases when the number of compromised node increases. Two schemes are similar with respect to the numbers of compromised nodes and multi path. The content of entire message will be not leaked in case of that the number of compromised nodes is less than 20, the part of the S between 0 to 20 until time t in Fig. 4. In Fig. 4, two PRRs ratios are approximately the same in ratio because it only considers the received PRR in FBSP. However, in Fig. 5, when the number of the retransmitted packets is considered, the PRR of the proposed SDGP outperforms the one of FBSP.

When a wireless sensor network is attacked by compromised nodes, the compromised node tries to send a false message to sensor's the destination. Fig. 5 represents the case of retransmitting packets again after packet is intercepted by compromised nodes. This is to disperse altered information in the entire networks. In our proposed scheme, altered information can be transmitted to sink. The ratio of receiving packets in proposed scheme decreases slower than that of FBSP due to the process of security, as shown in Fig. 5. In SDGP, the use of secure keys based on the hierarchical key settlement over secure paths allow safety authentication and avoid message coming from compromised nodes.



[Fig. 5] Performance of PRR with retransmitting packets by compromised node



[Fig. 6] The amount of energy consumption depending on increasing the number of compromised nodes

Fig. 6 plotted the amount of energy consumption depending on increasing the number of compromised nodes. FBSP consumes much more energy comparing with SDGP due to increasing the number of messages sent by compromised nodes. On the contrary, the proposed SDGP relatively reduces dispersing the message by compromised nodes, and can decrease many useless transmissions between member sensors and the parties concerned. Thus, we reduce energy consumption.

6. Conclusion

This work has been proposed a secure data gathering protocol, or SDGP, which is composed of three phases, referred to as Phase I and II, to solve the jeopardizing problem under the attack of multiple compromised nodes over hierarchical WSN. In addition, SDGP can provide a cost effective secure key settlement to secure the paths. Thereby, SDGP does not require interactive key agreement between communication parties. The main advantages of SDGP fall into two capabilities: the first would establish multiple paths and collect data in secure with low overhead than the previous protocols and the second could keep security even if more than t data shares are exposed to the compromised nodes.

The results from the security analyses and network performance evaluations show that SDGP supports better security and good performance.

REFERENCES

[1] Mao, G.; Fidan, B.; Anderson, B. Wireless sensor network location techniques. Computer Networks: The International Journal of Computer and Telecommunications Networking. 2007, 29, pp. 2529-2553.

[2] Arampatzis, T.; Lygeros, J.; Manesis, S. A survey of applications of wireless sensor and wireless sensor networks. In Proceedings of IEEE International Symposium on, Mediterrean Conference on Control and Automation, Limassol, Cyprus, 2005, 6, pp. 719-724.

[3] Yuxin, M.; Guiyi, W. A Feedback-based Multipath Approach for Secure Data Collection in Wireless Sensor Networks. Sensors. 2010, pp. 9529-9540.

[4] Culler, D.; Estrin, D.; Srivastava, M. Overview of sensor networks. IEEE Comput. 2004, 8, pp. 41-49.

[5] Tiwari, M.; Arya, K.V.; Choudhari, R.; Choudhary, K.S. Designing Intrusion Detection to Detect Black Hole and Selective Forwarding Attack in WSN Based on Local Information. 2009 Fourth International Conference on Computer Sciences and Convergence Information Technology. 2009, pp. 824-828.

[6] Giruka, V. C.; Singhal, M.; Royalty J.; Varanasi, S. Security in wireless sensor networks. Wireless Communications and Mobile Computing. 2008, 8, pp. 1-24.

[7] Ozdemir, S. Secure and reliable data aggregation for wireless sensor networks. Lecture Notes in Computer Science. 2007, 4836, pp.102-109.

[8] Sang, Y.; Sang, H.; Shen, Y.; Inoguchi, Y.; Tan, Y.; Xiong, N. Secure data aggregation in wireless sensor networks: A survey. Proc. of the Seventh International Conference on Parallel and Distributed Computing, Applications and Technologies. 2006, pp. 315-320.

[9] Karlof, C. Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures. Proc. of IEEE SNPA 2003. 2003, pp. 113-127.

[10] Yang, Y.; Wang, X.; Zhu, S.; Cao, G. SDAP: A secure hop-by-Hop data aggregation protocol for sensor networks. Proc. of ACM MobiHoc 2006, 2006.

[11] Shu, T.; Liu, S.; KrunzSecure, M. Data collection in wireless sensor networks using randomized dispersive routes. Proc. of IEEE INFOCOM 2009, 2009, pp. 2846-2850.

[12] Mahesh, K.M.; Samir, R.D. Ad hoc on-demand multipath distance vector routing. ACM SIGMOBILE Mobile Comput. Commun. Rev. 2002, pp.92-93.

[13] Lee, S. J.; Gerla, M. Split multipath routing with maximally disjoint paths in Ad hoc networks. Proc. of ICC 2001. 2001, pp. 3201-3205.

[14] Guo, H.; Mu, Y.; Lin, Z.; Zhang, X. An efficient and non-interactive hierarchical key agreement protocol. Computers & Security. 2011, 30, pp. 28-34.

[15] Boneh, D.; Franklin, M. Identity based encryption from the Weil pairing. Lecture Notes in Computer Science. 2001, 2139, pp. 213-229.

[16] Wu, K.; Dreef, D.; Sun, B.; Xiao, Y. Secure data aggregation without persistent cryptographic operations in wireless sensor networks. Ad Hoc Networks. 2007, 5, 1, pp.100-111.

[17] Shamir, A. How to share a secret. Comm. ACM, 1979, 22, pp. 612-613.

[18] Kim, H. S. Location-based authentication protocol for first cognitive radio networking standard. Journal of Network and Computer Applications. 2011, 34, pp. 1160-1167.

[19] He, D. An efficient remote user authentication and key agreement protocol for mobile client-server environment from pairings. Ad Hoc Networks. 2012, 10, pp. 1009-1016.

[20] Gouvea, C. P. L; Lopez, J. Software implementation of pairing-based cryptography on sensor networks using the MSP430 microcontroller. Lecture Notes in Computer Science. 2009, 5922, pp. 248-262.

[21] Rosli, R.; Yusoff, Y. M.; Hashim, H. A review on pairing based cryptography in wireless sensor networks. Proc. of ISWTA 2011. 2011, pp. 48-51.

[22] Szczechowiak, P.; Oliveira, L. B.; Scott, M.; Collier, M.; Ricardo, D. NanoECC:Testing the limits of Elliptic Curve Cryptography in Sensor Networks. Lecture Notes in Computer Science. 2008, 4913, pp. 305-320.

[23] Xiong, X.; Wong, D. S.; Deng, X. TinyPairing: A fast and lightweight pairing-based cryptographic library for wireless sensor networks. Proc. of WCNC 2010. 2010, pp. 1-6.

최 해 원(Choi, Hae Won)

・1996년 2월 : 경일대학교 컴퓨터공학과(공학사)
・2000년 2월 : 경북대학교 컴퓨터공학과(공학석사)
・2009년 2월 : 경북대학교 컴퓨터공학과(공학박사)
・2006년 3월 ~ 현재 : 경운대학교 컴퓨터공학과 조교수
・관심분야 : 알고리즘, 유비쿼터스 컴퓨팅
・E-Mail : happy9950@hotmail.com

이 채 수(Chae-Soo Lee)

・1994년 2월 : 경북대학교 전자공학과 (공학사)
・1996년 2월 : 경북대학교 전자공학과 (공학석사)
・2000년 2월 : 경북대학교 전자공학과 (공학박사)
・1999년 3월 ~ 현재 : 경운대학교 모바일공학과 교수
・관심분야 : Color Image Processing, 모바일 컨텐츠
・E-Mail : csl@kw.ac.kr

류 명 춘(Myung-Chun Ryoo)

・1989년 2월 : 영남대학교 컴퓨터학과(공학사)
・1991년 2월 : 영남대학교 컴퓨터공학과(공학석사)
・2009년 2월 : 영남대학교 컴퓨터공학과(공학박사)
・1997년 3월 ~ 현재 : 경운대학교 컴퓨터공학과 교수
・관심분야 : 지능정보시스템, Bioinformatics
・E-Mail : mcr@ikw.ac.kr

김 현 성(Hyung-Sung Kim)

・1996년 2월 : 경일대학교 컴퓨터공학과 (공학사)
・1998년 2월 : 경북대학교 컴퓨터공학과 (공학석사)
・2002년 2월 : 경북대학교 컴퓨터공학과 (공학박사)
・2002년 3월 ~ 현재 : 경일대학교 컴퓨터공학과 교수.
・관심분야 : 정보보호, 병렬처리
・E-Mail : kim@kiu.ac.kr