SCIS 2011 The 2011 Symposium on Cryptography and Information Security
Kokura, Japan, Jan. 25-28, 2011
The Institute of Electronics, Information and Communication Engineers

Secure, Fast Rebuilding, and Energy Efficient Routing Protocol for Mission-Critical Application over Wireless Sensor Networks

Made Harta Dwijaksara * Doyoung Chung * Yi Jae Park * Jangseong Kim *
Kwangjo Kim *

Abstract— The mission-critical application over wireless sensor network (WSN) such as fire alarm, radiation leak, and surveillance reconnaissance should support fast, reliable, and fault tolerant on its routing protocol. Otherwise, the application cannot support its own functionality and bring unexpected losses. However, the existing routing protocols are believed neither to consider security issues nor to deal with system reliability. In this paper, we propose a secure, fast rebuilding and energy efficient cluster based routing protocol for mission-critical application. Compared to LEACH and HPEQ, our approach provides reliability while reducing processing time and energy dissipation through cluster-based authentication mechanism and delayed propagation of management messages. According to the NS2 simulation, we can reduce 10%∼15% of overall energy dissipation and 40% of cluster rebuilding time.

Keywords: mission-critical application, WSN, cluster-based authentication, delay propagation.

1 Introduction

Wireless Sensor Network (WSN) is one of the fundamental technologies for building ubiquitous computing environments. The feasible applications of WSN can be classified into environmental, military, health, and home applications, *etc.* Some of these applications should send the sensed data in real-time and be recovered even if the unexpected failures have been occurred. Otherwise, we may suffer from severe damage on economic or environment. Here, we call these applications as "mission-critical" applications over WSN. The typical examples of these applications are fire alarm, monitoring of toxic area, habitat monitoring, radiation leak in nuclear power plant, and surveillance reconnaissance, *etc.*

As the WSN consists of many sensor nodes with limited resources (*i.e.* computational power, storage and battery), it has many security vulnerabilities [1] than other conventional networks (*e.g.* LAN and mesh network). For instance the fire alarm application installed to monitor fire in the forest. In such application, the adversary may inject false data to the network. As a result the appropriate action cannot be executed properly and may endanger the human being. However, to provide a good security mechanism requires high computation overhead which is not suitable for sensor nodes. This trade off, between security and efficiency, becomes a fundamental issue in deploying WSN application for mission-critical application [2].

Due to the limited resources of the sensor nodes, multi-hop communication in WSN is required. From this point, we believe that routing protocol has an important role in order to provide reliable communication for WSN. To extend the network lifetime, an efficient routing protocol is mandatory. That's why cluster based routing protocol (CBRP) is introduced. As the CBRP can support in-network data aggregation, the energy consumption for reporting the sensed data can be reduced. Although several efficient routing protocols have been proposed in the literature, most of them did not consider security issues [3, 4]. Even if some protocols consider security issues, but they are still vulnerable to the insider attacker. When an adversary compromises a sensor node, the adversary can easily obtain the secret information from the sensor node within a few minutes [5]. Therefore, these protocols [6, 7, 8, 9, 10] cannot be applied for mission-critical applications.

A routing protocol for mission-critical applications should consider the following problem: When the cluster head should be changed or an unexpected failure occurs in a cluster head, any cluster members should perform cluster selection procedure. During this procedure, new cluster head should send the notification message to its potential members. As multi-hop communication in WSN is mandatory, in order to deliver the messages this procedure takes several amount of times and consumes a proper amount of energy. This situation can be an issue in mission-critical applications. Such a new selection process of CH is called cluster "rebuilding process".

In this paper, we propose a secure, fast rebuilding and energy efficient routing protocol for mission-critical application over WSN. Through candidate selection and

* KAIST, 373-1 Guseongdong, YuseongGu, Daejeon, 305-701 Korea Computer Science, KAIST South Korea, {made.harta, wordspqr, qkrdlwo, jskim.withkals, kkj@kaist.ac.kr}

delay propagation mechanism, we can reduce energy consumption and time required for cluster rebuilding. Only two-hop distance nodes from new cluster head are required to join rebuilding process and the other remaining nodes can use the previous cluster information. Later, these nodes will be informed with new cluster information using delay propagation mechanism. To illustrate the efficiency of our protocol, we simulate our protocol with the well-known efficient routing protocols, *i.e.*, HPEQ [4]. Compared to HPEQ, we can reduce 10%∼15% of overall energy dissipation and 40% of cluster rebuilding time.

The rest of this paper is organized as follows: In section 2, we describe the related work in the literature and its shortcomings. We present the detail idea behind our proposed approach in Section 3. The simulation result used to evaluate our idea is showed in Section 4 together with the security analysis. Finally, our conclusion and future research are discussed in Section 5.

2 Preliminary and Related Work

We consider cluster based WSN which is comprised by four entities *i.e.* base station (BS) or sink, gateway, cluster head (CH) and sensor node. A BS has the role to collect all the sensing data from the sensor nodes. The gateway only forwards message from CH-to-BS or vice versa (no computation is performed by gateway). By introducing the gateway, we can reduce energy consumption of the intermediate nodes. In addition, we can reduce transmission delay and packet loss due to congestion close to the base station. That's why many prototypes system support gateway [11, 12]. Furthermore, the network will be partitioned into several clusters whereas each cluster has a CH selected from the sensor node. To reduce the burden of sensor node which acts as a CH, the role of CH will be distributed to all cluster members. Therefore, the cluster needs to be rebuilt every particular interval of time to select new a CH. Figure 1 shows the typical system model of cluster-based WSN.

While the CBRP has some advantages in term of energy efficiency, it also has several shortcomings with respect to the security. Due to inherent characteristic of WSN, the attackers are very easy to compromise the network if no strong security mechanisms have been employed. To model the attack, we assume that the attackers know the security mechanism used and can overhear all messages exchanged within the network. The attacker can analyze the security protocol used and launch the attack. The attack on cluster based WSN mostly fall into several types of attack, such as[8]: bogus routing information, hello floods, sinkhole, black hole, select forward and denial of service. The countermeasure for those attacks is by authenticating the CH and sensor node securely. Hence, using cryptography must be mandatory. The most dangerous attack happens when the adversary has compromised the legitimate node by capturing this node physically. The

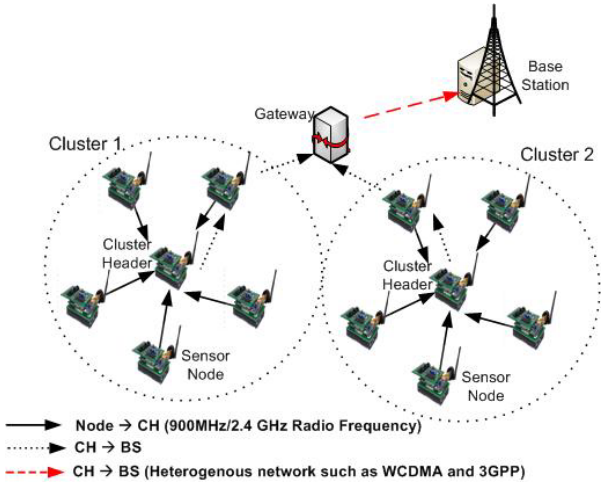


Figure 1: Cluster-based WSN

adversary can extract all the key material stored on the node and exploit this information to execute another attack. This is very likely to happen due to nature of WSN and high cost deploying tamper-resistant sensor node.

2.1 Cluster Based Routing Protocol

There are several studies which tried to address the problem of secure and energy efficient CBRP for WSN, as briefly introduced on the previous section. LEACH [3] and HPEQ [4] are two well known CBRPs for WSN. In LEACH, CH is selected using random mechanism, each node will generate random value. If the value generated is less than the pre-defined threshold, it will announce itself as CH. Upon receiving join request from node, CH creates schedule for the cluster members when they can send data to the CH. By this method cluster members can save energy because they can turn themselves into sleep mode. All communication here is done directly (1 hop communication only) without intermediate node. HPEQ uses LEACH method to select CH but instead of using direct communication, HPEQ implements multi hop mechanism so it increases scalability and distribute energy dissipation evenly among all members of the network. Therefore, the authors claimed that HPEQ provides better performance compare to LEACH.

In order to limit the size of a cluster, cluster advertisement packet carries a time-to-live (*ttl*) field which is the number of hop to the cluster head. The node will join the cluster which is closer to the cluster head, by checking the *ttl* of the message. The problems of this scheme are: 1) if the *ttl* is set too small, some nodes may not listen to the CH notification message. 2) If the *ttl* is set too big, the scheme requires too much time to send advertisement and to finish the cluster formation. We verified these problems using NS2 simulator and the results are shown in Figures 2 and 3. Moreover, LEACH and HPEQ also are not employed with security features hence they are very vulnerable

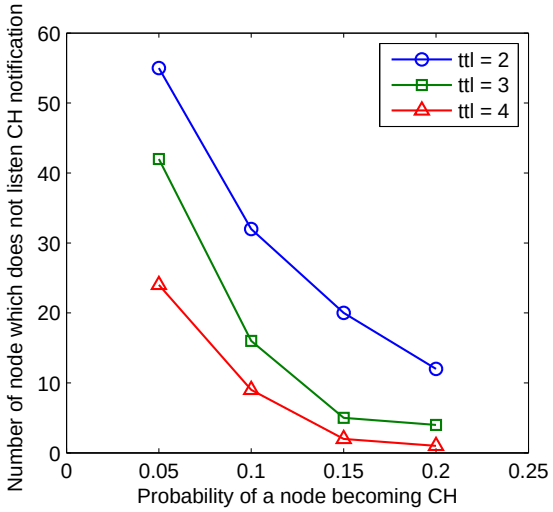


Figure 2: Average number of nodes which does not listen CH notification

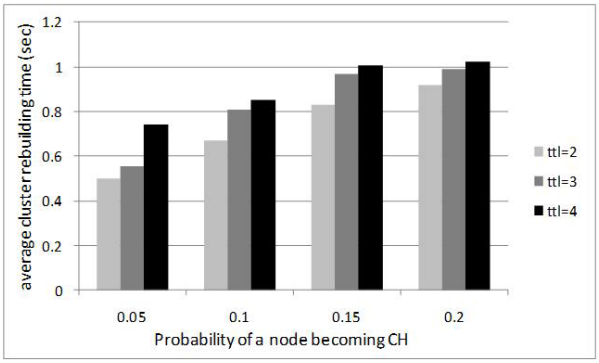


Figure 3: Rebuilding time

to any attacks.

2.2 Secure Cluster Based Routing Protocol

Leonardo *et al.* [6] introduced SecLEACH, a CRP which employs LEACH combined with random key distribution for securing node-to-CH and CH-to-BS communication. A set of key ring is selected from a pool of keys and loaded to each node in the network. During the setup phase, after a set of CH sends the notification message the other nodes select the nearest CH to which it shares the same key. The problem with this scheme is when the node shares the same key with CH which distance is very far. It will exhaust much energy of the node to send data to CH. This method later improved by Abuhelaleh *et al.* (Armor-LEACH) [7]. Mallanda *et al.* introduced SCEAR [8]. SCEAR uses public key cryptosystem to secure the network. The secret key is generated by CH and distributed to each of the sensor nodes in the cluster after encrypting it using their corresponding public keys. This scheme currently is quite impossible to be applied on resource limited device (sensor node).

Similar with thus SecLEACH and Armor-LEACH, Zhang *et al.* introduce RLEACH [9]. RLEACH also employs random key distribution. But instead of directly using shared key from a key ring for encryption, it computes shared key first using hash function which input is the shared key on key ring and group seed predefined prior to deployment of the sensor nodes. In SELDA [10], reputation value (Rv) is used. The CH will calculate Rv of its neighboring nodes. Every data sent by node to CH will be weighted using Rv. When nodes send data to CH, they use multi path link so that CH will receive duplicate message. This scheme is also inefficient since every node need to send data using multi path link. Lastly, there are several methods (SAPC [13], L. Hu *et al.* [14]) which use micro-tesla key chain distribution [15] for delivering secret key from BS to CH and node as well. Finally, when the aggregated data sent by CH is received by BS, BS will notify CH back about which data is the legitimate one. Later CH will take action based on BS's information. The usage of micro-tesla key chain may introduce high overhead since it requires time synchronization.

The approaches merely concern with the security feature of CBRP and neglect the energy efficiency. In addition, those works mostly based on LEACH which suffers from energy efficiency and scalability disadvantages compared to HPEQ. In our approach, we examine an efficient CBRP which employs multi hop communication based on HPEQ, since it has better performance compared to LEACH. We found that by adjusting the cluster rebuilding process, we can reduce energy usage of the network and at the same time provide a secure cluster-based routing protocol. Therefore we can make a good deal between energy usage and security.

3 Secure, Fast Rebuilding and Energy Efficient CBRP

We will design secure, fast rebuilding and energy efficient cluster based routing protocol which is based on HPEQ. Although HPEQ has advantages regarding with energy efficiency and scalability, it suffers from some disadvantages due to the use of *ttl* on the CH notification message. Moreover, it also does not support fast rebuilding process. When CH fails, the time required to rebuild the cluster is same as the time required for cluster formation. This is unsuitable for mission-critical application because it requires fast data delivery even though such case happens. On the other hand, in security-sensitive environment we have to consider the existence of malicious users. Malicious user may compromise the nodes especially CH which has very vital role in maintaining the availability of network services. The compromised CH may drop or modify the authentication request from cluster members and its corresponding response. It may also send fake aggregation message to BS, collaborate with other compromise nodes to continuously select themselves as CH. Therefore, we need to equip our scheme with the authentication protocol to ensure that the joining node is a legitimate node.

Even we have prevented the illegitimate node to join the network by employing cryptographic tools, the network is still vulnerable to the attacker which compromises the nodes. Such kind of attack is known as insider attacker. From this point, the intrusion detection mechanism is mandatory tools to enhance the security feature of our protocol. For intrusion detection we observe the misbehavior of the nodes (Note for this work we will only consider the misbehavior of the CH, since compromising CH will result more severe side effects than compromising sensor nodes.). We categorize the misbehavior of CH into three group:

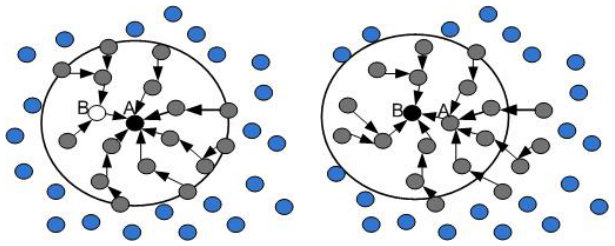
1. **Misbehavior during cluster formation**
   The first point of failure when there is a compromised CH during the cluster formation stage. In our protocol, the selected CH should pick one of its neighbor nodes which have the most energy left as cluster head candidate (CHC). The compromised CH will misbehave by not selecting any CHC or only select particular nodes as CHC which has been compromised as well. It results that the member nodes join the compromised cluster. Therefore, there is no guarantee that the sensing data received from this cluster is valid.

2. **Misbehavior during data aggregation**
   The compromised CH may make fake data aggregation and send it to the BS. This gives a very severe effect since the BS should make decision based on the information sent by the CH. The false aggregate data from CH will result in wrong action to be taken by BS.

3. **Misbehavior during data reporting**
   The last action but not the least that might be done by the compromised CH is: by not reporting the aggregate data to the BS. Therefore, the existence of all node members becomes useless due to their sensed data is not delivered to BS. On the other side, the BS does not know whether the cluster has been compromised. The BS may think that there is collision during message delivery therefore it cannot receive sensed data from that particular cluster.

3.1 Fast Cluster Rebuilding

The problem in HPEQ is caused by *ttl* field in the CH notification message. In our approach, we distinguish the initial cluster formation and next cluster formation process. The appropriate *ttl* is only used once during the initial cluster formation to limit the flooding message. The initial cluster formation process is based on HPEQ which is the first CH will be selected using LEACH mechanism. The selected CHs send the notification message to other nodes using multi hop communication. Thus, all nodes can listen the notification messages from CH and select one which has the shortest path to CH (based on the number of hop). This initial cluster formation is only done once upon the deployment of the sensor network. On the next cluster formation process, we introduce the notion of CHC. CHC is selected by the current CH from its neighbor node which has the most energy left. On the next change cluster time, the CHC will be a CH for that particular cluster where it belongs to. Then newly selected CH sends the notification only to several nodes (*e.g.,* the node within 2-hops distance from this CH). The other node can use the previous cluster information for delivering sensed data to CH. As a result, all nodes can still join the network whether they do not receive direct notification from CH. It also saves more energy, since we do not need to flood the network with the advertisement message. Furthermore, the cluster rebuilding time may be reduced since fewer nodes are involved during cluster formation process. We depict this idea in Figure 4, Figure 4(a) shows that cluster head A selects B as a cluster candidate. At time to change cluster, B will be a cluster head and A just turns itself as normal node (Figure 4(b)). B notifies all nodes within 2-hops distance. The 3-hops distance nodes or more are not notified by this new CH (but they may overhear another advertisement from another cluster head). When those nodes receive notification from other cluster head, they will compare the hop count to CH currently recorded with the hop count in the notification message. If the hop count in the message is less than their currently recorded hop count, they will join the cluster where the notification message comes from, if not they will just follow the previous route.

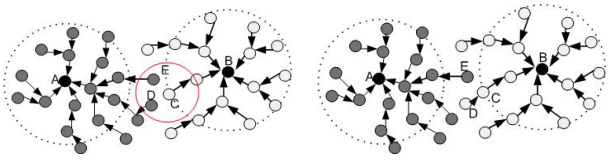


(a) Case 1 (b) Case 2

Figure 4: CHC is turned to be CH

Later, the nodes which do not receive CH notification during cluster formation are updated with new cluster information using delay propagation mechanism. When a node sends its sensed data to CH, it appends the current recorded hop count to CH on the message. The neighbor nodes which overhear this message will compare their current recorded hop count with the hop count on the message. If the hop count on the message is less than their current recorded hop count, the nodes will change the path to the node which sends the message. Figure 5 shows the delay propagation mechanism.

In Figure 5(a), node C joins cluster B, node D and E join cluster A. When C sends its sensed data to B together with hop count information (2 hops), nodes D and E overhear this message and compare their current



(a) Before delay propagation (b) After delay propagation

Figure 5: Delay propagation mechanism

Table 1: Notations

Notation	Note
CH_NTF	CH notification message
ID_X	ID of node X
CN / P	CH neighbor / parent node
N / C	Ordinary nodes / All cluster nodes
K_G	A global key preloaded to node
K_X	A key is owned by entity X
CK	A cluster key
$E(M, K_A)$	A message M is encrypted by K_A
$MAC(M, K_X)$	MAC operation with message M and key K
$AUTH\_REQ_X$	Authentication token of node X, $Credentials_X$ $\|\|E(ID_X\|\|TR\|\|Nonce, K_X)$
$Credentials_X$	Pseudonym of node X, $E(ID_X\|\|Nonce, K_{BS})$
TR	The number of transmission and receiving
REQ_EN	Request the remaining amount of energy
REP_EN	Reply message to the sender node
SET_CHC	Message to notify the selected CHC
=> / ->	Broadcast / unicast transmission

recorded hop count to A (D = 4 hops, E = 3 hops). D will change its path to C since hop count through C is less than current recorded hop count (*i.e.* 2 + 1 < 4). But E will not change its path since the current recorded hop count is same with the hop count through C (*i.e.* 3 + 1 ≤ 4), like Figure 5(b).

3.2 Cluster-Based Authentication

In Table 1, we present the notations that we will use throughout the paper.

3.2.1 Cluster Formation

The CH is selected using LEACH method. The selected CH floods a notification message which contains ID of CH, new *Nonce* (nn) to guarantee freshness and hop count hp.

(S1) [CH => N] CH_NTF$||E(ID_{CH}||nn||hp,\ K_G)$

As we mentioned before, we distinguish the cluster formation between initial and next cluster ones. During initial cluster formation, all neighbor nodes which receive this message forward this message. The node compares the hop count in each message and select the closest CH. But, during next cluster ones only node within 2-hops distance from CH will forward the message. After selecting the CH, the node answers with its authentication token (AUTH_REQ). The authentication token contains two essentials factors: *Credential* and *TR*. *Credential* is used to prevent exposure of the cluster topology from eavesdropper by encrypting node's ID and a *Nonce* with the BS's unique key. *TR* is a remaining energy metric for observation by the BS. This metric consist of just 16 bit. Half bits are for transmission and the other is for receiving. If the number of communication is over 8 bit, it will be set into 0, but the BS can calculate properly.

(S2) [N -> P] AUTH$_REQ_N$

Parent nodes receive reply from their children and then attach their authentication token. Then, they transmit this message recursively to higher parents which send notification message to them before.

(S3) [P -> CH] AUTH$_REQ_P||$AUTH$_REQ_{N1}||...$

Finally, the CH gathers authentication tokens. The CH computes MAC for the message. This MAC is added to the message along with its own *Credential* and REP_EN, then send it to BS.

A = AUTH$_REQ_1||...||AUTH_REQ_N$
(S4) [CH -> BS]
A$||REP_EN||Credentials_{CH}||MAC(A, K_{CH})$

BS authenticates the message upon receiving it. If the received message is valid, the BS generates the cluster key(CK) and new *Credentials* which contain each node's ID and new *Nonce*. Then, these values are encrypted with nodes' unique key before transmitting them.

(S5) [BS => C]
$Credentials_X||E(CK||newCredentials_X,\ K_{BS})$

The generated cluster key will be used for aggregation of sensed data from legitimate cluster members.

3.2.2 Cluster Head Candidate Selection

The selected CH broadcasts an energy request message to its immediate neighbors. The encrypted message with cluster key enables only valid nodes within the cluster to decrypt it.

(S6) [CH => CN] $REQ\_EN||E(ID_{CH}||Nonce, CK)$

All the immediate neighbor nodes reply to the message with their current amount of energy left.

(S7) [CN -> CH]
$REP_EN||E(ID_{CN}||$Nonce+1$||$Amount of Energy$, CK)$

Nonce is added by 1 from the original nonce for the freshness of the sent message. Then, CH choose the neighbor node which has the most energy left as CHC. This information is sent back to the neighbor nodes.

(S8) [CH => CN]
$SET\_CHC||E(ID_{CH}||ID_{CHC}||\text{Nonce+2}, CK)$

However, although the CH selects a node as CHC, we assume that CH does not believe the selected CHC yet, since an adversary can compromise a normal node and exaggeratedly inform its remaining amount of enery resources. Thus, CH asks the BS to authenticate the selected CHC during the next cluster formation.

3.3 Distributed Misbehavior Detection

The intrusion detection is intended to detect the compromised node which is physically captured by the malicious user. The misbehavior of CH explained before becomes the basic criteria of detection scheme. Mainly, the misbehavior detection of CH is done by immediate neighbor of CH but the other several nodes also contribute to the detection algorithm. But, the cluster members within the path from CH to the BS will also observe the packet sent by the Ch to BS.

3.3.1 Detecting misbehavior during cluster formation

To detect collaboration attack (two compromised nodes interchangeably select themselves as CH and CHC), the immediate neighbor nodes of CH will observe the CHC notification message from CH. Using the assumption that when a certain node become a CH, it will exhaust more energy than ordinary node due to intensive computation for message aggregation. If energy for processing one sensed data packet is E_C (*e.g.*, 8mA [16]), energy for receiving and transmitting packet are E_R (*e.g.*, 12mA [16]) and E_T (*e.g.*, 24mA [16]) respectively, number of node in the cluster is k (*e.g.*, $k = 20$) and when being a CH, the node should aggregate data n (*e.g.*, $n = 5$) times, then CH will dissipate E_{dis} more energy than the sensor nodes, where:

$$\begin{aligned} E_{dis} &= n \times [k \times (E_C + E_R) + E_T] \\ &= 5 \times [20 \times (8 + 12) + 24] \\ &= 5 \times (424) = 2,120mA \end{aligned} \quad (1)$$

The value of k and n used above are just example. In the real situation their value may vary based on the network condition. E_T is for CHC notification sent by CH to its immediate neighbor nodes. If the energy dissipated by a sensor node in reporting the sensed data to CH is E_{node} (Assume that the node has to receive and forward $p = 5$ packets on multi hop communication.).

$$\begin{aligned} E_{node} &= n \times [E_C + E_T + p \times (E_R + E_T)] \\ &= 5 \times [8 + 12 + 5 \times (12 + 24)] \\ &= 5 \times (20 + 180) = 1,000mA \end{aligned} \quad (2)$$

This result implies that the CH dissipates twice much more energy compare to the sensor node. As a result if at time i node A acts as CH, it is impossible that the same node will be a CH again at the time j, where $i < j < i + 3$ unless there is only 3 or less nodes in the cluster. Since, the node A must dissipate more energy than its immediate neighbor nodes at that moment. Therefor its energy left must be less than its neighbors. The detection mechanism requires the nodes to record the last 2 CHs. If the current selected CH is on the list on those last 2 CHs, it will trigger intrusion alarm. This information is reported to all node members and BS as well. And the illegitimate node can be isolated from the network.

3.3.2 Detecting misbehavior during data aggregation

During data aggregation, the compromised CH may create a fake aggregation value and send it to the BS. When the BS receives this fake aggregation data, BS may make a wrong decision based on the sensed data received. Hence to detect this type of misbehavior, the nodes estimate the distribution parameters of the sensed data which they have received. The sensed data has distribution with mean $= \bar{x}$ and standard deviation $= s$ [17]. Any data outside the acceptance region (*acc. reg.*) will be threaten as outlier.

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i \quad (3)$$

$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2} \quad (4)$$

$$acc.\ reg. = (\bar{x} - k \times s, \bar{x} + k \times s) \quad (5)$$

for some constant k. When the node overhear the aggregation message from CH to BS, it checks wheather the aggregation value resides within the acceptance region. Any detection of outlier will be reported to BS. If BS receive two or more outlier reports from the same cluster, BS can make decision that the corresponding CH has been compromised. This mechanism becomes visible since we use multi-hop communication. Hence, the node within the cluster receive and forward packet from other node to the CH.

3.3.3 Detecting misbehavior during data reporting

The compromised CH which wants to make the services of cluster may be not available to the BS. The CH does not report the aggregation data to BS. In this case, BS does not know whether CH intentionally does not report the aggregation data or the data has been lost due to collision during transmission. We use the WatchDog and PathRater technique [18] to monitor the activities of CH. Detection of compromised CH will be reported to other node and BS as well. Therefore, the compromised CH can be isolated from the network.

4 Security Analysis and Simulation

4.1 Security Analysis

S-FREE provides authentication, confidentiality, message integrity and freshness either for uni-cast (CH-to-BS), multi-cast (CH-to-nodes) and broadcast (CH advertisement) communication. It is also employed with intrusion detection mechanism to deal with the compromised CH. Hence, the security requirements for mission critical applications including reliability can be achieved. The K_G which was preloaded into each node prior to deployment is used for initial authenticate during cluster formation. To mitigate the illegitimate node joining the network, the authentication token is sent to BS. Later, BS verifies each token whether it is valid or not. This verification can also ensure that the adversary can not do forgery attack to the network. The forgery attack is done by compromising one node and use the secret information inside to create a fake node. Since the authentication token uniquely determine the node, if the forgery attack exist there will be a duplicated authentication token received by the BS. As a result, BS can recognize which node is being compromised and the fake node as well. Finally when the authentication is done, each cluster will be assigned the CK that can be used for further secure communication among the cluster.

The proposed scheme is also employed with the intrusion detection mechanism to monitor the compromised CH. In CBRP, CH is likely to be a target of attacking since it has a role as the center of the cluster. By compromising the CH, the adversary can be the owner of the cluster. Our intrusion detection mechanism can detect the misbehavior of compromised CH by observing its activities. The WatchDog and PathRater help in determining whether CH honestly follows the protocol in reporting the aggregation message to BS. We believe through those misbehavior observation, our protocol is robust enough against CH compromised attack.

4.2 Simulation

To verify our approach we conduct numerical experimentation using NS2. We compare our result with the original work of HPEQ. If our scheme performs better than the original work of HPEQ, we can claim that our work will also perform better than all those other LEACH and HPEQ based approaches.

4.2.1 Simulation Setting

The complete simulation parameters are listed on Table 2. For simulation metric, we address the total energy usage, cluster rebuilding time and energy distribution.

4.2.2 Simulation Result

As specified before, we address three simulations metrics on this section such as: total energy dissipation, cluster rebuilding time and energy usage distribution.

Total Energy Dissipation

Figure 6 shows the total energy dissipation by the network during 500 second simulation time. Roughly from the graphic, our approach can reduce energy usage almost 10%∼15% compare to the HPEQ when probability of node being CH is set to 0.2 and *ttl* is set to 4.

Table 2: Simulation Parameters

Parameters	**Value**
Simulation time	500 second
Number of nodes	100 node
Cluster round time	20 second
Transmit Energy	10^{-10} J/bit.m^2
Receive Energy	5×10^{-8} J/bit
Radio Range	10m

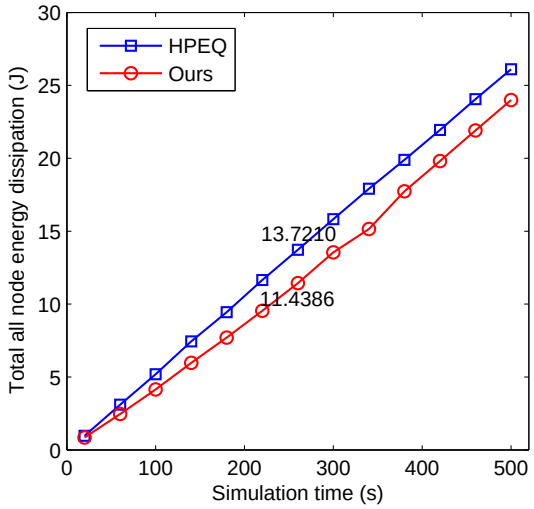


Figure 6: Total Energy Dissipation

Cluster Rebuilding Time

Figure 7 shows that the cluster rebuilding time using our approach is faster than HPEQ. Therefore we can claim during the critical condition and when CH is not working properly, the time required to rebuild the cluster is shorter. As a result, the message does not need to wait longer on the certain node before it can be delivered to BS.

Energy Usage Distribution

The last result Figure 8 shows that the energy distribution can be still maintained, though for some nodes it shows high discrepancy. It is because that a node may be selected as CH for multiple times. But overall it is still evenly distributed and the most important thing it is less than HPEQ.

5 Conclusion and Future Work

We proposed a secure, fast rebuilding and energy efficient cluster routing protocol for mission-critical application over WSNs. It employs with fast rebuilding mechanism to achieve the speed requirement and cluster based authentication for reliability requirement on

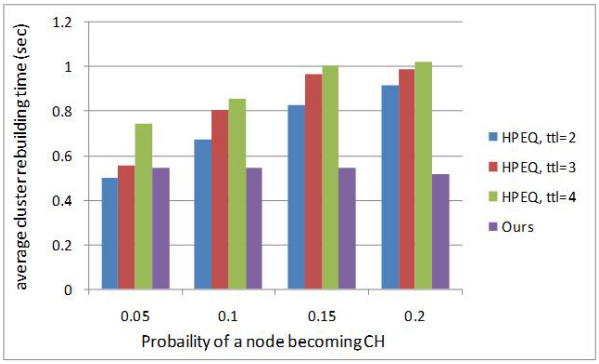


Figure 7: Cluster Rebuilding Time

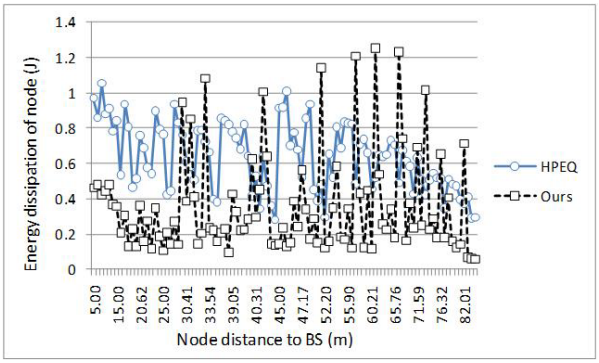


Figure 8: Energy Usage Distribution

mission-critical environment. According to simulation result, our scheme can reduce 10%∼15% of total energy dissipation and 40% of average cluster rebuilding time.

Our simulation did not consider node's movement. In addition, we also only consider grid topology of the network. For future work, we would like to do an extensive simulation for our protocol especially when the node is moving under random network topology. The simulation of multiple attackers on WSN is required as well.

References

[1] C. Karlof and D. Wagner. "Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures". *Ad Hoc Networks*, vol. 1, Elsevier, 2003, pp. 293-315.

[2] Y. Wang, G. Attebury and B. Ramamurthy. "A Survey of Security Issue in Wireless Sensor Network". *IEEE Communications*, vol. 8, No. 2, IEEE, 2006.

[3] W. R. Heizelman, A. Chandrakasan and H. Balakrishnan. "Energy-Efficient Communication Protocol for Wireless Microsensor Networks". *In the Proceeding of the Hawaii International Conference on System Sciences*, IEEE, 2000.

[4] A. Boukerche, R. W. N. Pazzi and R. B. Araujo. "HPEQ - A Hierarchical Periodic, Event-driven and Query-Based Wireless Sensor Network Protocol". *LCN'05*, IEEE, 2005.

[5] C. Hartung, J. Balasalle and R. Han. "Node Compromise in Sensor Networks: The Need for Secure Systems". *Technical Report CU-CS-990-05*, January, 2005.

[6] L. B. Oliveira, M. Bern, H. C. Wong, R. Dahab and A. A. F. Loureiro. "SecLEACH - A Random Key Distribution Solution for Securing Clustered Sensor Networks". *NCA'06*, IEEE, 2006.

[7] M. A. Abuhelaleh, T. M. Mismar and A. A. Abuzneid. "Armor-LEACH - Energy Efficient, Secure Wireless Networks Communication". *ICCCN'08*, IEEE, 2008.

[8] C. Mallanda, S. Basaravaju, A. Kulshrestha, R. Kamman, A. Durresi, and S.S. Iyengar. "Secure Cluster based Energy Aware Routing for Wireless Sensor Networks". 2008.

[9] K. Zhang, W. Cong, and W. Chuirong. "A Secure Routing Protocol for Cluster-Based Wireless Sensor Networks Using Group Key Management". *WiCOM '08*, IEEE, 2008.

[10] S. Ozdemir. "Secure and Reliable Data Aggregation for Wireless Sensor Networks". *LNCS 4836*, Springer-Verlag, 2007, pp. 102-109.

[11] R. Beckwith, D. Teibel, and P. Bowen. "Pervasive Computing and Proactive Agriculture". *in Adjunct Proceedings PERVASIVE Computing and Proactive Agriculture*, Vienna, Austria, 2004.

[12] C. Kappler and G. Riegel. "A Real-World, Simple Wireless Sensor Network for Monitoring Electrical Energy Consumption". *in Proceeding of First European Workshop on Wireless Sensor Networks*, Springer-Verlag, 2006, pp.339-352.

[13] C. Berkara, M. L. Maknavicius and K. Berkara. "SAPC: A Secure Aggregation Protocol for Cluster-Based Wireless Sensor Networks". *LNCS 4864*, Springer-Verlag, 2007, pp. 784-798.

[14] L. Hu, D. Evans. "Secure Aggregation for Wireless Networks". *SAINT-W '03*, ACM, 2003.

[15] A. Perrig, R. Szewczyk, V. Wen, D. Culler and J. Tygar. "SPIN: Security Protocols for Sensor Networks". *In Proceedings of Mobile Networking and Computing 2001*, ACM , 2001.

[16] G. Anastasi, M. Conti, A. Falchi, E. Gregori, A. Passarella. "Performance Measurements of Mote Sensor Networks". *MSWiM '04*, ACM, 2004.

[17] A. Hayter, *Probability and Statistic for Engineers and Scientist, 3rd Edition*, Thomson Brooks/Cole, 2007.

[18] S. Marti, T. Giuli, K. Lai and M. Baker. "Mitigating routing Misbehavior in mobile Ad hoc networks". *in Proceedings of MOBICOM 2000*, 2000, pp.255-265.