석 사 학 위 논 문

Master's Thesis

무선 센서네트워크에서의 안전한 에너지 효율적인 통신 방법

A Secure and Energy Efficient Communication Scheme for Wireless Sensor Networks

마데 하르타 드위작사라 (Made Harta Dwijaksara)

전산학과

Department of Computer Science

한 국 과 학 기 술 원

Korea Advanced Institute of Science and Technology

2011

무선 센서네트워크에서의 안전한 에너지 효율적인 통신 방법

A Secure and Energy Efficient Communication Scheme for Wireless Sensor Networks

A Secure and Energy Efficient Communication Scheme for Wireless Sensor Networks

Advisor : Professor Kwangjo Kim

by

Made Harta Dwijaksara
Department of Computer Science
Korea Advanced Institute of Science and Technology

A thesis submitted to the faculty of the Korea Advanced Institute of Science and Technology in partial fulfillment of the requirements for the degree of Master of Science in the Department of Computer Science

Daejeon, Korea
2011. 05. 25.
Approved by

Professor Kwangjo Kim
Advisor

A Secure and Energy Efficient Communication Scheme for Wireless Sensor Networks

Made Harta Dwijaksara

위 논문은 한국과학기술원 석사학위논문으로 학위논문심사 위원회에서 심사 통과하였음.

2011년 05월 25일

심사위원장 Kwangjo Kim (인)

심사위원 Hyunsoo Yoon (인)

심사위원 Myungchul Kim (인)

MCS
20094300 마데 하르타 드위작사라. Made Harta Dwijaksara. A Secure and Energy Efficient Communication Scheme for Wireless Sensor Networks. 무선 센서네트워크에서의 안전한 에너지 효율적인 통신 방법. Department of Computer Science . 2011. 48p. Advisor Prof. Kwangjo Kim. Text in English.

Abstract

The need for ubiquitous computing has lead to new invention of technologies which are able to leverage those computing powers in our daily life. One of the well known technologies which comes up to serve this pervasive computing is Wireless Sensor Network (WSN). WSN consists of spatially distributed autonomous sensor to quickly and efficiently monitor physical or evironmental conditions, such as temperature, fire, sound, pressure, vibration, motion, pollutant, border surveillances, *etc.* and to cooperatively pass their data through the network to a dedicated location (*i.e.,* base station (BS)). For that reasons WSN is compound by hundreds or even thounsands sensor nodes in order to cover wider monitoring area. These low cost sensor nodes have inherent characteristic such as limited energy resource, limited computation power and small memory storage. As a result to deliver an application over these resource constrained devices one should consider the trade off between these limitations and the performance requirement.

The mission-critical application over wireless sensor network (WSN) such as fire alarm, radiation leakage, surveillance reconnaissance, *etc.* should be fast, reliable, fault tolerant on its routing protocol. Otherwise, the application cannot support its own functionality and bring an unexpected failure. However, the existing routing protocols are found not to consider security issues and to deal with system reliability together. In this paper, we propose a secure, fast rebuilding and energy efficient cluster-based routing protocol for mission-critical application. Compared to previous protocols like LEACH and HPEQ, our approach provides better reliability while reducing processing time and energy dissipation through cluster-based authentication mechanism and delayed propagation of management messages. According to the NS2 simulation, our protocol consumes a reasonable amount of energy and reduces almost 30% of cluster rebuilding time compared to HPEQ.

Contents

List of Tables

List of Figures

1. Introduction

This section discusses the role of Wireless Sensor Network (WSN) in supporting the Ubiquitous Computing Environment (UCE). It provides brief overview about WSN it self and how it can be beneficial when used in supporting UCE. Meanwhile, the contribution of the paper is also discussed to complete the introduction section.

1.1 Overview

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 severe damage from economic or environmental point of view. Here, we call these applications as "mission-critical applications" over WSN. The typical examples of these applications are fire alarm, monitoring of toxic area, 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), WSN 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. In general, security mechanism is believed to require 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].

Similar with those conventional network technologies, wireless sensor networks also consist of several network layers. In order to provide security services over this network, we should consider the security on every single layer of sensor network. Several researches have been done [2, 3, 6] in this field to show the security vulnerabilities of sensor networks. In this work, we mostly will address the security on Network layer of the wireless sensor network. This is because this layer is very typically to be a target of attack due to its critical role in preserving the existence of the network services. Beside that, we will

also discuss the security feature on MAC (Medium Access Control) layer altogether with its vulnerabilities. Because this layer defines the most basic protocols in order the sensor network can be functioned as it should be.

IEEE 802.15.4 is the predominant standard commonly used to speficy the physical and MAC layer for Low Rate Wireless Personal Area Networks (*i.e.* sensor network) [4]. Sensors rely on MAC layer to coordinate their transmission to share the wireless media fairly and efficiently. Furthermore, MAC layer is also resposible for the security features as specified by IEEE 802.15.4 standard. MAC layer should provide a set of security suites to ensure confidentiality, privacy and integrity, but the application developers can also specify their security suite in the upper layer (network or application layer) in addition to security features provided by the standard. In general, there are three main categories of security services provided when the communication is done [5, 6].

1. Message Integrity Code (MIC), the term message integrity code is used instead of message authentication code (MAC) to avoid confusion with the medium access control (MAC). Applying only MIC will ensure the message is not changed illegitimately during the transmission without noticed by the legitimate user. In this scheme only integrity can be ensure not with the confidentiality.

2. Message Encryption/Decryption, with this mechanism the message sent during communication can only be read by the authentic user, it means the confidentiality can be achieved but not the integrity of the message.

3. Combination of both, the encryption/decryption method is done to the message as well the message integrity code is embedded to the message being sent. By combining these two schemes it can ensure that we will get the confidentiality and integrity.

Beside those three security features, the message received should be also protected from being replayed (replay attack). A device should be able to distinguish other devices that are willing to be authenticated to communicate. This can be done by keeping track the freshness of the message sent by those devices. But, due to the incomplete design of the standard [5], there are some vulnerabilities regarding of the IEEE 802.15.4's security.

In other hand, the network layer responsible for the routing protocol used for communicating between one node with other nodes. Since the intensive coomunication can occur within the network, an efficient routing protocol is required to extend the network lifetime. This is a reason 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 [8]. A CBRP for WSN should consider either regular

CH (Cluster Head) selection process or new CH (Cluster Head) selection process when the unexpected failure occur on the current CH. The regular new CH selection process is required for load balancing so the energy usage is well distributed over the cluster member. Such a new selection process of CH is called "cluster rebuilding process". Although several CBRPs have been proposed in the literature, most of them [9, 10] did not consider security issues. Even though some protocols consider security issues, but they are still inefficient and vulnerable to the insider attacker who has compromised the legitimate node by capturing the node physically and use it to attack the network. The attacker can extract the key material stored on the node within a few minutes [11] and exploit this information to execute another attacks such as: bogus routing information, hello flooding, sinkhole, black hole, selective forwarding and denial of service attacks. Therefore, previous protocols [12, 13, 14, 15, 16, 18, 19] cannot be applied for mission-critical applications.

1.2 Our Contribution

In this thesis, we firstly survey the security vulnerabilities of wireless sensor network both in MAC and Network layers. From this result, we suggest several security improvements for the IEEE 802.5.4 standard which specifies the MAC layer for wireless sensor network. In Network layer side, we point out some weaknesses on the current well known CBRP (*i.e.* HPEQ - Hierarcichal Periodic, Event-Driven and Query-Based WSN) using simulation tools. Later, we propose a secure, fast rebuilding and energy efficient CBRP for mission-critical application over WSN. Through CH Candidate (CHC) selection and delayed propagation mechanism, we can reduce energy consumption and time required for cluster rebuilding process. Only two-hop distance nodes from new CH are required to join cluster rebuilding process and the other remaining nodes can use the previous cluster information. Then, these nodes will be informed with new cluster information using delayed propagation mechanism. Furthermore, the cluster-based authentication and misbehavior detection provide the security feature to defend against various attackers. To illustrate the efficiency of our protocol, we simulate then compare our protocol with the HPEQ [10]. Simulation result shows that our protocol consumes a reasonable energy and reduces 30% of cluster rebuilding time.

1.3 Organization

The rest of this thesis 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 protocol in Section 3. The simulation result used to evaluate our protocol is showed in Section 4 together with the security analysis. Finally, our conclusion and future research are discussed in Section 5.

2. Related Work and Background

This section explains the drawbacks of current WSN design which can affect the reliability of the network service. It is also shown that the previously introduced standard and related works are still inadequate in supporting the security requirement for mission critical applicaiton over WSN. Therefore, we claim that several improvement need to be done in the standard and a new approach should be introduced to adequately serve the security requirement of WSN in mission critical application. Furthermore, the system model is explained following those security vulnerabilties discussions. Finally, several assumption are described to support our approach.

2.1 Related Work

WSN consist of hundreds or even thousand small resource constrained sensor nodes. These sensor nodes are self organized to form the ad hoc wireless networks. In these days the WSN applications are widely used to support Ubiquitous Comptuing Environment (UCE). This trend attracks the malicious user to exploit the security vulnerabilities of WSN. In order to protect the WSN applications we have to apply security protocol in each layer of WSN so forth there is no any hole that can be used by malicious user to exploit the networks. Below we discuss the security of MAC layer which is specified in IEEE 802.15.4 standard and Newtork layer of WSN, but before that we want to show the characteristic of sensor node.

2.1.1 Sensor Nodes

As previously mentioned, sensor node is a resource constrained device which has very limited computation capability, memory storage and source of power. In addition, sensor node is also not tamper resistant therefore it is very easy to be compromised. Figure 2.1 present the example of sensor node (Mica2Dot manufactured by Crosbow Inc.)

As we can see, the main components of a typical sensor node include an antenna and a radio frequency (RF) transceiver to allow communication with other nodes, a memory unit, a CPU, the sensor unit (i.e. thermostat) and the power source which is usually provided by batteries. The operating system running on sensor nodes is called TinyOS and

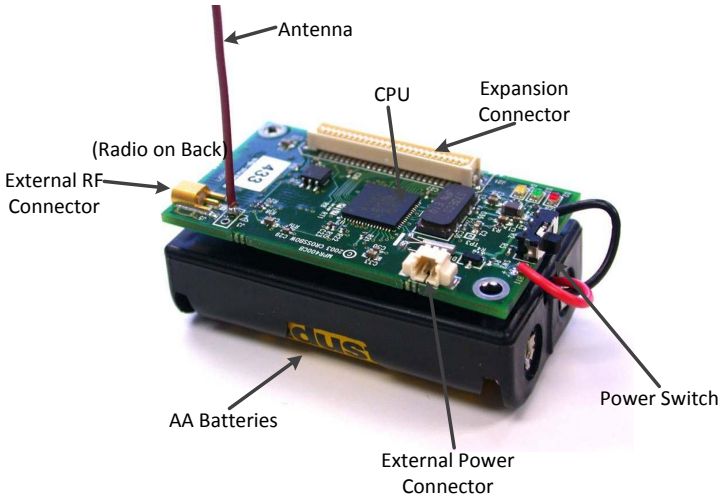


Figure 2.1: Mica2Dot sensor node

was initially developed at the University of California, Berkeley. TinyOS is designed to run on platforms with limited computational power and memory space.The programming language of TinyOS is stylized C and uses a custom compiler called NesC. Though it may work on other platforms, the supported platforms are Linux RedHat 9.0, Windows 2000, and Windows XP. Figure 2.2 lists some of the capabilities of the current sensor network platforms organized by device class.

2.1.2 IEEE 802.15.4 Security

Access Control List (ACL)

IEEE 802.15.4 standard packs the security suites into ACL which contains list of the security suites available and the associate address of devices to which particular security method will be used. It controls the access to which device the communication can be established by associating its supported security services. When the device knows which security service will be used that information is stored as ACL entry. The ACL then becomes a look up table for finding the correct security suite of other devices. The format of the ACL can be seen in Figure 2.3.

Node	CPU	Power	Memory	I/O and Sensors	Radio	Remarks
Special-purpose Sensor Nodes						
Spec 2003	4-8MHz Custom 8-bit	3mW peak 3uW idle	3K RAM	I/O Pads on chip, ADC	50-100Kbps	Full custom silicon, traded RF range and accuracy for low-power operation.
Generic Sensor Nodes						
Rene 1999	ATMEL 8535	.036mW sleep 60mW active	512B RAM 8K Flash	Large Expansion	10Kbps	Primary TinyOS development platform.
Mica-2 2001	ATMEGA 128	.036mW sleep 60mW active	4K RAM 128K Flash	Large Expansion connector	76Kbps	Primary TinyOS development platform.
Telos 2004	Motorola HCS08	.001mW sleep 32mW active	4K RAM	USB and Ethernet	250Kbps	Support IEEE 802.15.4 standard. Allows higher-layer Zigbee standard. 1.8V operation
Mica-Z 2004	ATMEGA		4K RAM 128K Flash	Large Expansion connector	250Kbps	Support IEEE 802.15.4 standard. Allows higher layer Zigbee standard.
High-bandwidth Sensor Nodes						
BT Node 2001	ATMEL Mega 128L 7.328Mhz	50MW idle 285MW active	128KB Flash 4KB EEPROM 4KB SRAM	8-channel 10-bit A/D, 2 UARTS Expandable connectors	Bluetooth	Easy connectivity with cell phones. Support TinyOS. Multi-hop using multiple radios/nodes
Imote 1.0 2003	ARM 7TDMI 12-48MHz	1mW idle 120mW active	64KB SRAM 512KB Flash	UART, USB, GPIO, I^2C, SPI	Bluetooth 1.1	Multi-hop using scatternets, easy connections to PDAs, phones, TinyOS 1.0, 1.1
Gateway Nodes						
Stargate	Intel PXA255		64KNSRM	2 PCMICA/CF, com ports, Ethernet, USB	Serial connection to sensor network	Flexible I/O and small form factor power management.
Inrysnc Cerfcube 2003	Intel PXA255		32KB Flash 64KB SRAM	Single CF card, general-purpose I/O		Small form factor, robust industrial support, Linux and Windows CE support
PCI 04 nodes	X86 processor		32KB Flash 64KB SRAM	PCI Bus		Embedded Linux or Windows support.

Figure 2.2: Sensor network platforms organized by device class [20]

Figure 2.3: ACL format

Table 2.1: Security Suites

ID	Security Suite	MIC	Encryption
0x00	No Security	-	-
0x01	AES-CTR	-	AES
0x02	AES-CCM-32	32-bit	-
0x03	AES-CCM-64	64-bit	-
0x04	AES-CCM-128	128-bit	-
0x05	AES-CBC-MIC-32	32-bit	AES
0x06	AES-CBC-MIC-64	64-bit	AES
0x07	AES-CBC-MIC-128	128-bit	AES

Security Services

As briefly introduced in the previous section, the IEEE 802.15.4 standard supports three main security services packed into ACL. Table 2.1 shows the complete security services provided by the standard.

MIC is computed as follows:

$$\begin{aligned} &X_{i+1} = E(Key, X_i \oplus B_i) \text{for all i} \in 0, 1, ...t \\ &where\ X_0 = 0^{128} \\ &E(k, m) = \text{Encryption with key} = k, \text{data} = m \end{aligned} \tag{2.1}$$

The MIC code is selected from n most significant bit of X_{i+1} (where n is the length of MIC code). The value of B_i is basically calculated from certain value called CCM* nonce (simply refer as nonce). CCM nonce is additional keying material for AES CCM. CCM is a new mode of operation of a block cipher that combines the existing Counter (CTR) and CBC-MIC mode. Figure 2.4 depicts the format of the nonce (Note that frame counter is a replay counter from ACL).

Combining data with nonce is supposed to produce a different value whenever a frame is sent. Similarly with the MIC generation method, the encryption process also requires

Figure 2.4: Format of CCM* nonce

nonce as input. Nonce will ensure that every time message is encrypted by certain device it will always produce different ciphertext even if the plaintext and the key are the same. The encryption mechanism is done as follows:

$$\begin{aligned} &C_i = E(Key, A_i) \oplus M_i \\ &A_i : \text{nonce for block } i^{th} \\ &M_i : \textit{block message } i^{th} \end{aligned} \tag{2.2}$$

2.1.3 Security Vulnerabilties of IEEE 802.15.4

Here we focus of the security vulnerabilities that related to the shortcoming design of the nonce, discussed in [6, 7] because it gives very severe impact if it is not well treaten. Beside the vulnerabilities caused by shortcoming design of nonce, there are also other security vulnerabilities in IEEE 802.15.4 standard [6, 7] which are out of our scope.

Same Key on Multiple ACL Entries

The nonce value is derived from the ACL entry. The value of replay counter will sequentially increase every time there is a packet necessary to be sent. Therefore, the value of replay counter on one entry happens to be the same with other counter, as shown in Figure 2.5. In addition, with the same key is used on multiple ACL entries, it may breaks the confidentiality property of symmetric cryptography.

Lemma 1. *[6] If on multiple ACL entries, the same key and the same value of nonce are used, when encrypted messages sent to different receiver it may break the confidentiality property of symmetric cryptography (i.e., the exclusive or of the two plaintexts is equal to that of the two ciphertexts).*

Proof. The cipher text is defined as $C = E(k, nonce) \oplus m$. So when there is two messages, $C_1 = E(k_1, nonce_1) \oplus m_1$ and $C_2 = E(k_2, nonce_2) \oplus m_2$. By result $C_1 \oplus C_2 = [E(k_1, nonce_1) \oplus m_1] \oplus [E(k_2, nonce_2) \oplus m_2]$. Since the value for key and nonce are same, so $C_1 \oplus C_2 = m_1 \oplus m_2$. □

Address	Security Suite (ID)	Key	Last IV	Replay Counter
04:6F:···.	0x07	XXX		YYY
1C:B2:···.	0x07	XXX		YYY
…….	…….	…….	…….	…….
AA:43:···.	0x07	XXX		YYY

Figure 2.5: Same key in multiple ACL entries

No support for Group Keying Model

The other implication of shortcoming design of nonce is that IEEE 802.15.4 standard does not support group keying model, because the nonce itself does not ensure uniqueness when used together with other nonce value on multiple ACL entries. It means the key should be different for each entry to ensure the confidentiality property will not be broken (group key is not allowed).

Incompatible with Network Shared Keying Model

The network shared keying model also cannot be supported well. As described before, the uniqueness of nonce is merely determined by the value of frame counter which serves as replay protection. This design is incompatible with replay protection when used to support network shared keying. In the network shared keying model only one key is used for all nodes then the key is loaded to default ACL (in this case ACL contains only one entry).

Lemma 2. *If the shared network key is used for all nodes, using the current standard, it will end to incorrectness of the network services (message from legitimate node will be rejected).*

Proof. Assume that senders S want to send message to single receiver R which will create only one record on its ACL entry. Now suppose sender S_j sends message with counter C_{j_k} ($k \in$ 1,..,n, $C_{j_1} = 0$) to R, R will increment its counter (C_r). Later, S_l (j $\neq$ l) sends the message to R with counter C_{l_m} ($m =$ 1,..,n, $C_{l_1} = 0$). If $C_{l_m} < C_{j_k}$ then $C_{l_m} <$ Cr, by result R will reject message from S_l because it will trigger replay attack alarm. In order that all messages from S_l can be accepted by R, C_{l_m} should be always greater than C_{j_k} where this is almost impossible to be fulfilled because $C_{j_1} = 0$ and $C_{l_1} = 0$. □

Lost ACL State Due To Power Failure

When the power of node is suddenly down, all the information about the ACL will be lost, because the information stored on volatile memory will disappear. At this point the node will have no information about the previous ACL's state when the node tries to recover later. Node will reset the information and use the default value for the ACL. Later, when node tries to communicate with other nodes within the same network it may reuse the same value for nonce.

2.1.4 Cluster Based Routing Protocol

After discussing security on MAC layer, now we turn into Network layer which defines routing protocol to be used for communication. LEACH [9] and HPEQ [10] are two well known CBRPs for WSN. Both methods use random CH selection mechanism. During CH selection, each node generates random number between 0 and 1 then compares it with the threshold (t) calculated using:

$$t = \frac{p}{1 - p * (r \ mod \ \frac{1}{p})} \tag{2.3}$$

Where p is the probability of nodes becoming CH and r is the current round. If the generated random number is less than threshold t, the node becomes CH for current round. The major different between two methods is: in LEACH all communication is done directly without intermediate node (1 hop communication only). On the other hand, HPEQ implements multi-hop mechanism so it increases scalability and distribute energy dissipation evenly to all members of the network. Therefore, HPEQ provides better performance compared to LEACH in term of energy usage and scalability.

In order to limit the size of a cluster, CH notification message in HPEQ carries a time-to-live (*ttl*) field which is the number of hop to the CH. One node may receive several CH notification messages which come from difference CH. In this case, the node compares the *ttl* value and joins the closest CH. The node will forward this notification message to its neighbors as long as the *ttl* value is not expired. The problems of this scheme are: 1) if the value of *ttl* is set too small, some nodes may not listen to the CH notification message. This is because the *ttl* is expired very soon. 2) If the value of *ttl* is set too big, the scheme requires a lot of time to finish the cluster formation. Since we have to wait longer for *ttl* to be expired and the nodes to reply to the notification message. We verified these problems using NS2 simulation and the results are shown in Figures 2.6 and 2.7. Note that both LEACH and HPEQ do not support security mechanism hence they are very vulnerable to

any attacks.

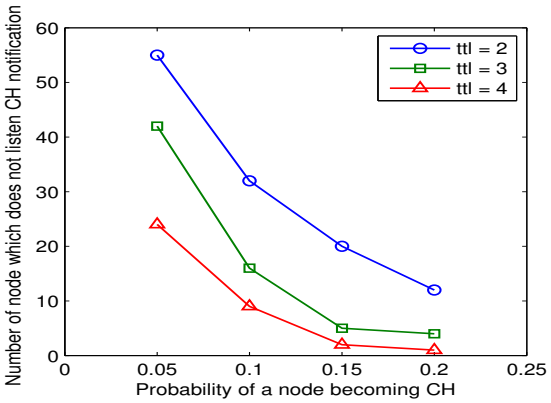


Figure 2.6: Average number of nodes which does not listen CH notification

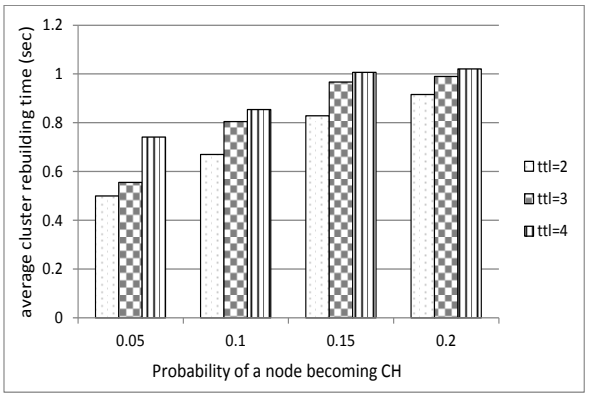


Figure 2.7: Rebuilding time

2.1.5 Secure Cluster-Based Routing Protocol

In term of secure CBRP, there are several protocols were published. Based on the security method applied, those protocols can be grouped into several categories.

Random key pre-distribution based approach [12, 14, 16]

Here, CBRP which employs LEACH is combined with random key pre-distribution for securing node-to-CH and CH-to-BS communication. In random key pre-distribution mechanism, a set of key is selected from a pools of keys and loaded to each node in the network prior to deployment. Two nodes can communicate each other if they share a same key. When this scheme is applied over LEACH which only uses one hop communication only, the nodes can join a cluster if they share the same key with the corresponding CH. The problem with this scheme is the probability of a node shares the same key with particular CH may be very low since the CH in LEACH is selected using random mechanism. In addition, the number of CH is also very limited for efficiency that make the chance of a node shares the same key with CH become lower. Therefore when a node (*e.g., A*) only shares the same key with CH which is located far away from *A*, A should consume much more energy than it should be to send data to CH. Even though there are nearby CHs, but since *A* does not share the same key with these CHs *A* can not communicate with them. In the worst case the nodes may not be able to join any cluster because they do not share same key with any CH.

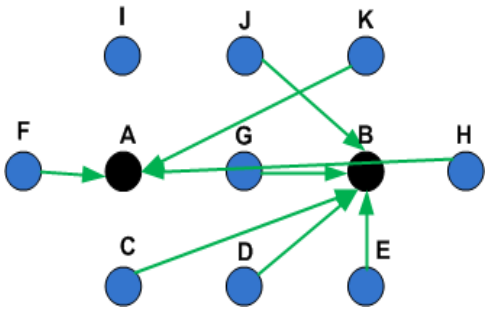


Figure 2.8: CBRP with random key pre-distribution

Figure 2.8 shows the network condition when a CBRP is used combined with the random key pre-distribution scheme. Node A and B are the CH in the network, the rest of the nodes are ordinary sensor node which should join one of the CH. In this case, node *C*, *H*, and *K* are failed to join the closest CH since they do not share any same key with the corresponding cluster. These nodes will exhaust more energy if they want to communicate with the CH. If such a system is not used these node can join the closest CH in order to minimize energy consumtion for communcation with CH. The worst case is node *I*. Since this node does not share any same key with current selected CH, node *I* can

not join any CH. Therefore, either node I directly communcate with the BS which will require much more energy or stay idle by doing nothing (but in this case the network service is not completely working since some nodes will not delivered any information about surrounding condition).

Micro-tesla based approach [15, 13]

Here, micro-tesla(µTESLA) key chain distribution [17] is used for delivering secret key from BS to CH and sensor nodes as well. µTESLA introduced asymmetry by delaying the disclosure of asymmetry keys. A sender broadcasts a message with a MIC generated with a secret key K, which will be disclosed after a certain period of time. When a receiver receives this message, if it can ensure that the packet was sent before the key was disclosed, the receiver can buffer this packet and authenticate it when it receives the corresponding disclosed key. To continuously authenticate the broadcast packets, µTESLA divides the time period for broadcasting into multiple time intervals, assigning different keys to different time intervals. The key for interval i (also known as key-chain) is generated follows:

$$\begin{aligned} &K_i = H(K_{i+1}) \ for \ 0 \leq i \leq n-1 \\ &where \ H : \text{any hash function} \\ &K_0 : \text{predefined initial key} \end{aligned} \tag{2.4}$$

The generation of these key-chain is done by the BS. The BS discloses the key per-each interval to all the nodes using broadcast mechanism. Then, all packets broadcasted in a particular time interval are authenticated with the same key assigned to that time interval by BS. Therefore, when the aggregated data sent by CH is received by BS, BS can notify CH back about which data is the legitimate one. The use of micro-tesla key chain may introduce high overhead since it requires time synchronization. Time synchronization is required to adjust the time when the BS should disclose the key-chain for particular interval. As a result, the nodes and BS can use the same key at the same time interval. But due to the nature of time synchronization, the nodes are required to send periodic time synchronization packet to ensure their local time is same with the global time controlled by BS. This intensive beacon packet requires the nodes to transmit message frequently which consume alot of energy.

Multi-path authentication based approach [18]

A message is sent through multi-path link. The message sent by certain node contains unique message ID which identify the source of node. In this case, the destination node will receive multiple duplicate message. Upon receiving these duplicates messages, the destination node compares the messages. If there is no attack these message should be same and vice versa.

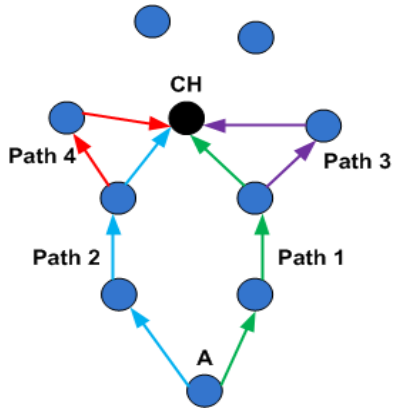


Figure 2.9: An example of multi-path authentication based approach

For example see Figure 2.9. The node A wants to send packet to CH. Multi-path links are used to delivered the packet to the CH. The CH finally receives 4 duplicate packets. Then, CH will compares all these packets. If there is no different is shown by the all packets the CH can decide that the packet is authenticated packet. In other hand, if CH finds any different in the packet, CH can suspect that some packets are not authentic. Then, CH can trace back to find the authentic packet and malicious node. This scheme is inefficient since every node needs to send data using multi-path link. Therefore, it wastes too many resources for delivering the authentic message.

Public key based approach [19, 21]

Public key cryptosystem is used 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). Table 2.2 [22] shows the time elapsed for publoic key cryptography algorithm in sensor node (Mica2Dot). The RSA refers to Riverst Aldeir

Table 2.2: Time Consumption for Public Key Crytpgrapghy in Sensor Node

Algorithm	**Key Size (bit)**	**Key Exchange**		**Signature**	
		Client	**Server**	**Sign (s)**	**Verify (s)**
RSA	1024	1.12	22.03	22.03	0.86
	2048	4.14	166.85	166.85	3.89
ECC	160	1.62	1.62	1.65	3.27
	224	4.38	4.46	4.46	8.84

Table 2.3: The Estimated Energy Consumption (mWs)

Algorithm	**Key Size (bit)**	**Key Exchange**		**Signature**	
		Client	**Server**	**Sign (s)**	**Verify (s)**
RSA	1024	39.96	726.99	726.99	28.38
	2048	136.62	5506.05	5506.05	128.37
ECC	160	53.46	53.46	54.46	107.91
	224	144.54	144.54	147.18	291.72

Shamir scheme and ECC refers to Eliptic Curve Crytography. Even when we use the ECC, we still require quite long time to finish the operation in sensor node compare to when we use symetric key algorithm (AES - 128bit). Using AES-128bit, the process only takes less than 15ms to finish [23].

In term of the energy consumption of the public key mechanism Table 2.3 [22] depicts the energy consumpt when RSA or ECC is used in sensor node (MICA2DOT).

Based on those facts, the public key cryptography is not suitable to be applied on resource constrained devices such as sensor node. That is why the symetric key cryptography is still the best choice to be used to provide security feature for WSN.

Also note that: all those previous works are 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. We found that by adjusting the cluster rebuilding process, we can reduce total energy usage of the network and at the same time provide a secure cluster-based routing protocol. Furthermore, we also consider the insider attack which is usually neglected in previous works.

2.2 Background

2.2.1 System Model

We consider cluster based WSN which comprises four entities *i.e.,* base station (BS), gateway (GW), CH and sensor nodes. BS has to collect all the sensed data from the sensor nodes. Figure 2.10 shows the typical system model of cluster-based WSN.

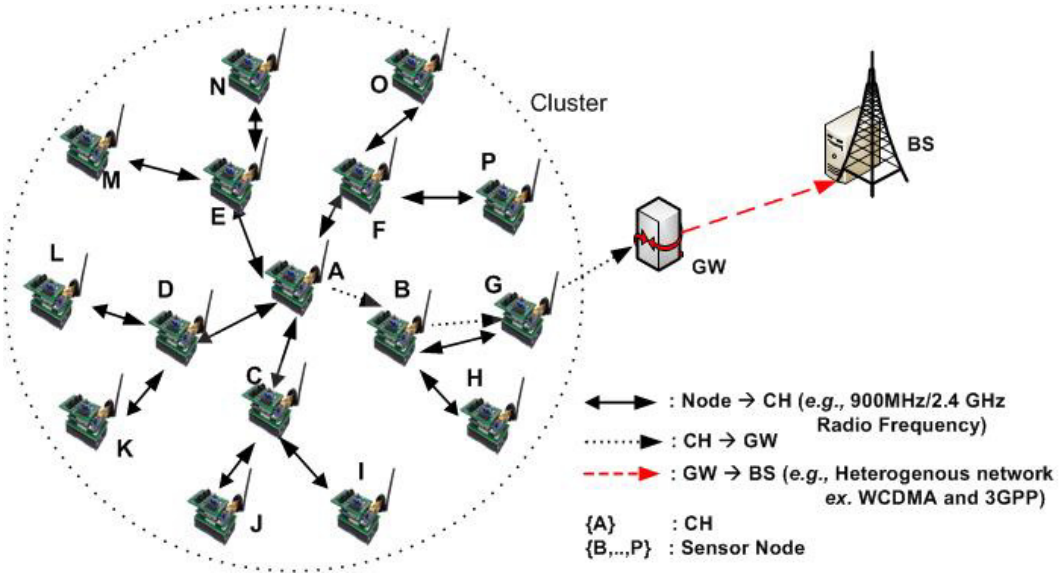


Figure 2.10: Cluster-based WSN

Base Station

Base station is the center of the network. It has the role to populate all the reading data about the environment from the sensor node. It may be located nearby the deployment place of the WSN or far away from the deployment place. The base station also has to be able to identify all the sensor node deployed within the network through their ID. Therefore, the BS should store the ID of sensor node in its local storage. The base station can communicate with the sensor node using various wireless communication means. The BS is not a sensor node instead it is special designed hardware which has powerful capability to support the network services. Upon receiving the reading data from the sensor nodes, the BS should be able to make appropriate decision regarding of those reading data. Predefined role may be already assigned to BS for making decision or it can be done manu-

ally by the operator who is controlling the BS. For the security sake, the BS will act as thrusted authority for the network since it can verify all members of the network.

Gateway

The GW is special designed sensor node which has powerful capability in term of data transmission and energy. It is not similar with ordinary sensor node which has limited capability. The GW can communicate within very far distance because it is equiped with special communication means such as: 2.4GHz radio, WCDMA and 3GPP similar with the BS. Therefore, it can communicate directly with sensor node and BS for data delivery. The GW only forwards message from CH-to-BS or vice versa (no computation is performed by GW). The GW will be distributed over the deployment area of WSN so the nearby sensor node can contact at least one of the GW directly or through multi-hop communication. By introducing the GW, we can reduce energy consumption of the intermediate nodes. Since the data from CH is only required to be delivered to nearby GW, then GW forwards it to BS. In addition, we can reduce transmission delay and packet loss due to congestion close to the base station. Thus, many prototype systems were designed to use GW [24, 26]

Sensor Node

As metioned on the previous section that sensor node is resouce constrained device which has very limited resources. This is because the sensor node should be as cheap as possible in order we can deploy as much as possible to cover wider area with less expense. The sensor placed inside the device will sensed if there is any change in the environment. This sensed (reading) data will be delivered to the CH which is the center of the cluster. For this purpose, to be able to join the network each sensor node should join one of the CH within the network. The sensor node uses multi-hop communication to contact the CH since it communication range is limited within certain meters. In our system model, the sensor node is not required to communicate with other sensor node. The sensor node only needs to verify then forward the message from other sensor nodes. Since multi-hop communication is used to contact the CH, routing protocol becomes one of the most important thing that should be accomplish by sensor node. In this case, our system only uses the CBRP for data delivery.

Cluster Head

CH is selected from ordinary sensor node. The CH has a role to collect the reading data from all of the cluster member. After receiving those data the CH aggregates the data and send it to the nearby GW. All the communication is done using multi-hop communication. For this purpose CH will exhaust more energy compare to the other sensor nodes. Since the CH is an ordinary sensor node it also has limited capability. Therefore for load balancing requirement the role of CH should be distributed to other nodes. When certain node is selected as CH, it has to notify the other nodes using advertisment message. By examaining this advertisement message the sensor node can determine which CH should be joined during that particular interval of time. For load balancing, the role of CH should be equally distributed to all cluster members. Therefore, the cluster needs to be rebuilt every particular interval of time (known as cluster round time) to select a new CH.

2.2.2 Assumption

To be able to conduct our research we take several assumptions related to the condition of WSN. For that reason, we consider the following assumptions in our model.

1. The network is static and the sensor nodes are deployed in appropriate way so it can cover the whole area of monitoring. Moreover, the sensor nodes are distributed evenly over the area of monitoring so the network density is quite similar for each are of monitoring.

2. BS has unlimited resource and it is assumed that the BS can not be compromised by malicious user. Every communication done by the BS is also assumed secure.

3. The GW is assumed has its own energy resource so any energy consumption by GW is not considered. Since GW also has powerful capability it can provides novel security mechanism therefore the communication between GW and BS is also assumed secure against any attacks.

4. The communication range of sensor node is limited withing certain meter. The link of this communication is bidirectional. It means when sensor node A can send data to sensor node B, then sensor node A can overhear every message sent by sensor node B.

5. The adversay can eavesdrop on all the traffic, inject packets, reply old message previously delivered and even capture the devices physically. If the adversary captures the devices, all the keying information it holds can also be compromised.

6. All the sensor nodes deployed within the monitoring area are pre-loaded with global key used for initial authentication. This key is also known by the BS an will be also used for future authentication when another initial cluster formation is required or when a node wants to join different cluster.

3. Proposed Scheme: Secure and Energy Efficient CBRP

This chapter focus on our proposed scheme for securing communication on WSN while minimizing the energy consumption. It mostly covers the security for network layer (routing protocol) which extends the capability of CBRP. But due to important role of MAC layer, we also discuss our security suggestion for MAC layer to be implemented in IEEE 802.15.4 standard. Before going further to our proposed scheme, at first we describe the security requirement that should be fulfilled by our protocol.

3.1 Security Requirements

Our protocol, secure communication scheme for wireless sensor networks, should achieve the following requirements.

3.1.1 Authentication

The sender of the messages should be authenticated to guard against impersonation attack. And, message authentication should also be provided to protect against forgery and related attacks. Even though an attacker compromises some sensor nodes, the attacker cannot execute forgery attack since the BS store all identification of sensor nodes so any forgery attack can be detected when the message is received by BS.

3.1.2 Integrity

The message integrity deals with methods that ensure that the contents of a message have not been tampered with and altered. It means that the transmitted message should be valid and thereis no illegitimate modification occur on the message during transmission. The sender should also be able to indentify whether the message comes from the legitimate sender or not.

3.1.3 Confidentiality

Confidentiality is required to ensure that the malicious user can not determine the content of the message being sent. The confidentiality feature is optional feature since not all communication message need to be hidden from the adversary. This feature is important when we want to protect the WSN from the adversary who wants to analyze the traffic to find out the network topology.

3.1.4 Freshness

Message freshness ensure that the scheme is secured against replay attacks. A replay attack is an attack where an authentication session is replayed by an attacker to fool other party into granting access. It may be any form or retransmission of a network data transmission but is usually used to gain authentication in a fraudulent manner.

3.1.5 Availability

We also have to ensure that the network service is always available. Previous security feature is still vulnerable against the malicious user who compromised the sensor node physically. Using compromised sensor node malicious user can execute further attack known as insider attack. It this case the adversaries can make the network service unavailable since they may use all of the network resources by themselves. Therefore, we implement miss-behavior detection to strengthen our scheme against such attack.

3.2 MAC and Network Layer Security

In this section, we describe our approach for securing communication of WSN while minimizing the energy usage. We firstly give a security suggestion for IEEE 802.15.4-standard which specifies the MAC layer for WSN. Then, we propose our secure CBRP over WSN which employs cluster fast rebuilding process, cluster based authentication mechanism and miss-behavior detection.

3.2.1 Security Suggestion for IEEE 802.15.4-Standard

After examining the security features of IEEE 802.15.4 especially related with nonce, we introduce the guidelines for designing nonce, i.e.

1. To make the value of nonce unique per each source node, nonce should be able to identify the source address.

2. The value of nonce should sequentially increase even for multiple source nodes.

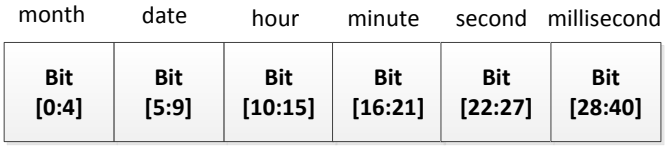
3. The value of nonce should be unique even in multiple ACL entries.

Based on the guidelines, the timestamp combined with source address can be used as a component of nonce. The nature of timestamp expresses that its value will increase each time. This characteristic directly expresses its uniqueness. Futhermore, the source address will give uniqueness per the source node.

Proposed Format for CCM* Nonce

Nonce contains 13 byte data as shown in Figure 3.1. Source address identifies the node origin of packet, so this value should not be changed. The field that we will modify is the last two fields with 5 byte long. We need to fit the timestamp to this short slot. Figure 6 shows the new nonce format using timestamp.

(a) Nonce



(b) Timestamp field

Figure 3.1: Proposed format for CCM* nonce

where:
month (4 bit long) : $1 \sim 12$
date (5 bit long) : $1 \sim 31$

hour (5 bit long) : $0 \sim 24$
minute (6 bit long) : $0 \sim 60$
second (6 bit long) : $0 \sim 60$
millisecond (13 bit long) : $0 \sim 1000$

Theorem 1. *If timestamp with precision until millisecond is used as a component of nonce, it will always give uniqueness to the value of nonce even it is used in multiple ACL entries. In addition, that value will aslo sequentially increase.*

Proof. If a sender S wants to send data to receiver R_1 and R_2 at the same time t (Assume that S uses same key k for both communication.). For this reason S will compute encrypted message C_1 and C_2 sequentially ($C = E(k, nonce) \oplus m$) at $t_1 = t$. Based on [23] using AES to finish computation of 30 byte message, it is required at least δt $\simeq$ 8ms. So encrypted message C_1 will be ready at $t_2 = t_1 + \Delta$t. Then at t_2, S will compute the next encrypted message (C_2) where $t_2 > t_1$. This implies that value $t_2 \simeq t_1$ by result $C_1 \oplus C_2 \neq M_1 \oplus M_2$ (contradictory with Lemma 1). Since t_i is always greater than t_{i-1}, the freshness will be always ensured as well (contradictory with Lemma 2). □

3.2.2 Secure and Energy Efficient CBRP

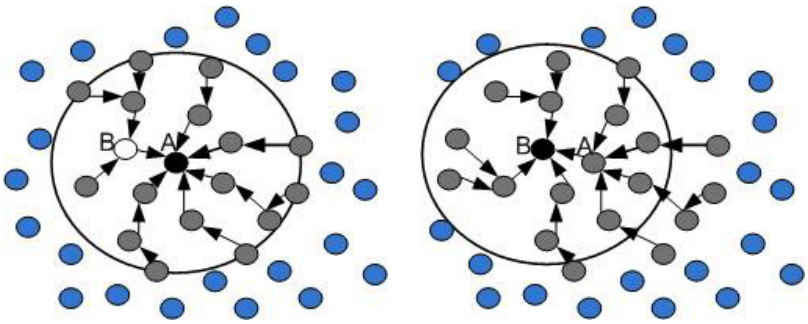
As we mentioned before in network layer we would like to deploy our secure and energy efficient cluster-based routing protocol. We refer to HPEQ protocol as the base idea which uses the multihop communication scheme. Our protocol employs efficient rebuilding process to minimize the time and energy usage for cluster rebuilding process, cluster-based authentication scheme for ensuring that the joining node is legitimate node and miss-behavior detection for detecting insider attacker which tries to compromise the CH.

Efficient rebuilding process

Although HPEQ has advantages regarding with energy efficiency and scalability, it suffers from disadvantages due to the use of *ttl* on the CH notification message. Moreover, it also does not support fast rebuilding process. When a CH is not working properly, the time required to rebuild the cluster is same as the time required for initial cluster formation. This is unsuitable for mission-critical application because it requires fast data delivery even though such case happens.

To support efficient rebuilding process, we introduce the notion of CHC and delayed propagation. The current CH selects CHC among its neighbor nodes having the most en-

ergy left. On the next cluster round time or when CH is not working properly, the selected CHC will become a new CH. The new selected CH only sends the notification message to the nodes within 2-hop distance from its self. Since the notification message does not flood the network, we can save more energy and the time required for cluster rebuilding process. We depict this idea in Figure 4, Figure 3.2(a) shows that A (CH) selects B as a CHC. At cluster round time, B will be a CH and A just turns itself as normal node (Figure 3.2(b)). B notifies all nodes within 2-hop distance. The 3-hop distance nodes or more do not received the notification message from B. Later, the nodes which do not receive CH's notification message during cluster formation are informed with new cluster information using delayed propagation mechanism. But, before these nodes receive delayed propagation message they may use the previous route.

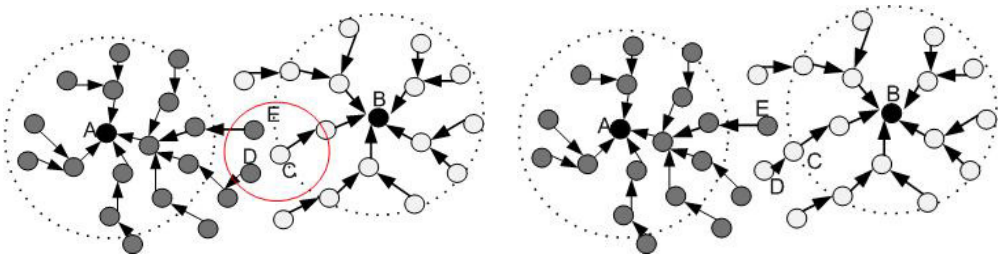


(a) Case 1 (b) Case 2

Figure 3.2: CHC is turned to be CH

The delayed propagation mechanism works as follow: 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 delayed propagation mechanism.

In Figure 3.3(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 hop), nodes D and E overhear this message and compare their current recorded hop count to A (D = 4 hop, E = 3 hop). 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 path through C is greater or same as the current recorded hop count (*i.e.* $3 + 1 \geq 4$), like Figure 3.3(b).



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

Figure 3.3: Delay propagation mechanism

Cluster-Based Authentication

In security-sensitive environment we have to consider the existence of malicious users. Malicious user may join the network and execute an attack either to interrupt the system or make system unavailable. Therefore, we equip our protocol with the authentication protocol and misbehavior detection to defend against the malicious users. Table 3.1 shows the notations used in the paper.

Cluster Formation

During initial cluster formation, the CH is selected using LEACH method. The selected CH floods a notification message which contains ID of CH, new *Nonce* to guarantee freshness and hop count. We refer to our system model (Figure 2.10) for our security protocol. In Figure 2.10, there are only one and two hops distance cluster members from CH. In real case there might be three or more hops distance cluster member within a cluster as well. The 1-hop cluster members are nodes {B, C, D, E, F} and the 2-hop cluster members are nodes {G, H, I, J, K, L, M, N, O, P}.

As we mentioned before, we distinguish the cluster formation between initial cluster formation and the next ones. During initial cluster formation, all nodes receive and forward CH notification messages. The nodes compare the hop count in each message and selects the closest CH. But, during the next cluster formation only node within 2-hop distance from CH will receive and forward the notification message. After selecting the closest CH, the node answers the notification message 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 used to track how

Table 3.1: Notations

Notation	**Note**
ID_X	ID of node X
CN	All CH immediate neighbors
N_X	Nodes within X-hop away from CH
K_G	A global key preloaded to node
K_X	A key is owned by entity X this key is also known by BS
K_C	A cluster key
$E(M, K_A)$	A message M is encrypted by K_A
$MIC(M, K_X)$	MIC (Message Integrity Code) or MAC (Message Authentication Code) operation with message M and key K
$AUTH\_REQ_X$	Authentication token of node X
$Credentials_X$	Pseudonym of node X, $E(ID_X \|\| Nonce, K_{BS-X})$
TR	The number of transmission and receiving packet
$\Rightarrow$ / $\rightarrow$	Broadcast / unicast transmission

many packets are transmitted by the node so far. Complete protocol for cluster formation is depicted in Figure 3.4.

Parent nodes receive reply message from their children and then attach their authentication token. The nodes forward this message recursively to the CH. Finally, the CH gathers all authentication tokens from the cluster members. The CH computes MIC for the message. This MIC is added to the message along with its own *Credential* and REP_EN, then send it to BS.

BS authenticates the message upon receiving it. If the received message is valid, the BS generates the K_C and new *Credentials* which contain each node's ID and new *Nonce*. Then, these values are encrypted with nodes' unique key before transmitting them. The generated cluster key will be used to secure communication on the corresponding cluster.

Cluster Head Candidate Selection

The selected CH broadcasts an energy request message to its immediate neighbors (1-hop nodes). The encrypted message with cluster key enables only valid nodes within the

CH Advertisement

1. $CH \Rightarrow N_1 : E\left(ID_{CH} \parallel Nonce \parallel HopCount\ , K_G\right)$
2. $N_1 \Rightarrow N_2 : E\left(ID_{CH} \parallel Nonce \parallel HopCount++, K_G\right)$

Join Message

3. $N_2\ computes \quad AUTH\_REQ_{N_2} =$
   $Credentials_{N_2} \parallel E\left(ID_{N_2} \parallel TR \parallel Nonce, K_{N_2}\right)$
   $N_2 \rightarrow N_1 : AUTH\_REQ_{N_2}$
4. $N_1\ computes \quad AUTH\_REQ_{N_1} =$
   $Credentials_{N_1} \parallel E\left(ID_{N_1} \parallel TR \parallel Nonce, K_{N_1}\right)$
   $N_1 \rightarrow CH : M = AUTH\_REQ_{N_2} \parallel AUTH\_REQ_{N_1}$

Authentication Process by *BS*

5. $CH\ computes \quad MIC = MIC\left(M, K_{CH}\right)$
   $CH \rightarrow BS : M \parallel Credentials_{CH} \parallel MIC$
6. *BS* generates *CK* and new *Credentials* for each node
   $BS \rightarrow N_X : Credentials_{N_X} \parallel E\left(K_C \parallel Credentials_{N_X}, K_{N_X}\right)$

Figure 3.4: Cluster-based authentication

cluster to decrypt it. After receiving energy request message, all the 1-hop neighbor nodes reply to the message with their current amount of energy.

Nonce is incremented to ensure the freshness of the sent message. Then, CH chooses the neighbor node which has the most energy left as CHC. This information is sent back to all 1-hop neighbor nodes. This process is showed in Figure 3.5.

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.

Energy Request

1. $CH \Rightarrow CN : E(ID_{CH} \,||\, Nonce, K_C)$
2. $CN \rightarrow CH : E(ID_{CN} \,||\, Nonce{+}1 \,||\, Energy, K_C)$

CHC Notification

3. $CH \Rightarrow CN : E(ID_{CH} \,||\, ID_{CHC} \,||\, Nonce{+}2, K_C)$

Figure 3.5: CHC selection

New Node Join the Network

Secure scheme for new joining node into the network is discussed here. The term of new joining node here refers to: 1) the process when there is a completely new node needs to be added to the network, 2) the process when a certain node from particular cluster has to join another different cluster regarding its distance to the CH. The first is needed because there is a possibility that we wants to cover wider monitoring area therefore some new sensor nodes have to be deployed and integrated to the existing network. In this case, the underlying network should be able to differentiate which one is a new legitimate sensor node and which one is not. Since the adversaries may also deploy their own malicious nodes at the same time. The second procedure is needed when a sertain cluster member receives delay propagation message from another CH and it is found the its current distance to current CH is longer than the new informed distance on the delay propagation message. Therefore, the node needs to change the cluster. But since the node does not hold any CK of the corresponding cluster, the node needs to be authenticated again to ensure that it is a legitimate node.

Thanks to the K_G and K_{node} which is loaded to the all nodes prior to deployment. For such authentication we can use these keys. The process is depicted in Figure 3.6. Using K_G the new node (NN) generates $AUTH\_REQ$ which is sent to the $inNode$ (intermediate node which will forward the message to the CH). Upon receiving this message, the CH verifies the message using K_G then compute the MIC for the message before forwarding it to the BS. BS verifies the message then assign CK and new credentials to the *NN*. This information is encrypted using the K_{NN} and sent directly to the NN by BS.

Join Message

1. $NN computes\ AUTH\_REQ_{NN} =$

   $Credentials_{NN} \parallel E(ID_{NN} \parallel Nonce, K_G)$

   $NN \rightarrow inNode : AUTH\_REQ_{NN}$

2. *inNode* verifies and forward the message to *CH*

   $CH\ computes\ \ MIC = MIC(AUTH\_REQ_{NN}, K_{CH})$

   $CH \rightarrow BS : AUTH\_REQ_{NN} \parallel MIC$

3. After successfully verifies the message, *BS* assigns

   CK and new *Credentials* for *NN*

   $BS \rightarrow NN : Credentials_{NN} \parallel E(K_C \parallel Credentials_{NN}, K_{NN})$

Figure 3.6: New node joins the cluster (network)

3.2.3 Misbehavior Observation

Even we have prevented the illegitimate node to join the network by employing cryptographic tools, the network is still vulnerable to the attacker who compromises the nodes. Such kind of attack is known as insider attacker. From this point, the intrusion detection mechanism is one of mandatory tools to enhance the security feature of our protocol. Since compromising CH will result more severe side effects than compromising sensor nodes therefore in this paper we mainly observe the misbehavior of the CH. The misbehavior of CH can be categorized into three cases:

1. **Misbehavior during cluster formation**: The adversary can compromise a CH during the cluster formation stage. In our protocol, when a sensor node is selected as the CH, the node should pick one of its neighbor nodes which have the most energy left as CHC. The compromised CH may misbehave by not selecting any CHC or select only other compromised node as CHC (collaboration attack).

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. If the BS receive fake aggregate data from CH, a wrong decision may be made by the 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.

The misbehavior detection is intended to detect the compromised CH, which is physically captured by the adversary. Because compromising CH give more severe effect than compromising sensor nodes.

Detecting misbehavior during cluster formation

To detect collaboration attack, the immediate neighbor nodes of CH observe the CHC notification message. Under the assumption that when a certain node become a CH, it will exhaust more energy than ordinary node due to the intensive communications for data aggregation. The energy consumptions on mica2dot Berkeley motes [25] are: energy for computation is E_C=8mA, energy for receiving and transmitting packet are E_R=12mA and E_T=24mA, respectively. If the number of node in a cluster is k (*e.g.*, k=20) and when being a CH the node should aggregate n times data (*e.g.*, n=5), then CH will dissipate E_{dis} energy, where:

$$\begin{aligned} E_{dis} &= n \times [k \times (E_C + E_R) + E_T] \\ &= 5 \times [20 \times (8 + 12) + 24] \\ &= 5 \times (424) = 2,120mA \end{aligned} \tag{3.1}$$

The value of k and n used above are just an example. In the real situation their values may vary based on the network condition. If the energy dissipated by a sensor node in reporting the sensed data to CH is E_{node}. Let us assume that each node has to receive and forward $p = 5$ packets.

$$\begin{aligned} E_{node} &= n \times [E_C + E_T + p \times (E_R + E_T)] \\ &= 5 \times [8 + 12 + 5 \times (12 + 24)] \\ &= 5 \times (20 + 180) = 1,000mA \end{aligned} \tag{3.2}$$

This result implies that the CH dissipates twice much more energy compared to the sensor node. As a result, certain node cannot become CH for multiple times when its immediate neighbors have never been selected as CH. On the other hand, 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. 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.

Algorithm 1 Detecting misbehavior during cluster formation

Require: List of previously two selected CHs

Ensure: Cluster rebuilding or a misbehavior report

```
currentCH ← listen_to_CH_notification()
if listPrevCH.length() = 2 then
    if listPrevCH.isExist(currentCH) then
        report_to_BS()
    else
        continue_rebuilding_process()
        listPrevCH.removeFirst()
        listPrevCH.add(currentCH)
    end if
else
    continue_rebuilding_process()
    listPrevCH.add(currentCH)
end if
```

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. To detect this type of misbehavior, the nodes estimate the distribution parameters of the sensed data which they have received. Assume the sensed data sent by cluster member i to CH is x_i. The CH should aggregate all these data, if the number of cluster member is n and the aggregate function is $f(x_i)$, $1 \leq i \leq n$. The immediate neighbor nodes of CH can compute estimation value for the aggregation result using $f(x_j)$, $1 \leq i \leq p$ where $p < n$. This is because the immediate neighbor nodes know some sensed data from the other cluster members since they have forwarded these data to CH. Finally, the immediate neighbor nodes define the confidence interval (ci) for the aggregation result of CH using $f(x_i) \in (f(x_j) \pm a)$ for some constant a. Any data outside ci will be threaten as outlier.

For example, let see the case if we use "mean" as our aggregation function f. If the

unbiased estimation of the aggregation result = $\bar{x}$ and the unbiased estimation of standard deviation of the aggregation result = s [27] then,

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{3.3}$$

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{3.4}$$

$$ci = (\bar{x} - k \times s, \bar{x} + k \times s) \tag{3.5}$$

k is constant value.

When the immediate neighbor nodes overhear the aggregation message sent by CH to BS, it checks whether the aggregation value resides within *ci*. Any detection of outlier will be reported to BS. If BS receives two or more outlier reports from certain cluster, BS can make decision that the corresponding CH has been compromised.

Detecting misbehavior during data reporting

The compromised CH may want to make the service of the cluster is not available to the BS. The compromised 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 [28] to monitor the activities of CH. The immediate neighbor nodes of CH define a rule that when they transmit sensed data to CH, the CH should send aggregate value to BS within several amount of time (timeout). The immediate neighbor nodes of CH can overhear this report message since the links are bidirectional. Using watchdog, when the certain timeout is reached and the immediate neighbor nodes do not overhear any report message from CH to BS, the immediate neighbor nodes may decide that the CH has been compromised. Detection result of compromised CH will be reported to other node and BS as well. Therefore, the compromised CH can be isolated from the network.

Algorithm 2 Detecting misbehavior during data aggregation

Require: maximum waiting time Δt

Ensure: Aggretation report to BS or a misbehavior report

1: **while** there is still data to be forwarded **do**
2: $newData \leftarrow data\_is\_received()$
3: $receivedData.addData(newData)$
4: $forward\_data\_to\_CH()$
5: $t_0 \leftarrow getCurrentTime()$
6: **end while**{*all data has been forwarded to CH*}
7: $dataCH \leftarrow listen\_CH\_report()$
8: $t_1 \leftarrow getCurrentTime()$
9: **if** $t_1 \leq t_0 + \Delta t$ **then**
10: $min \leftarrow receivedData.getMin()$
11: $max \leftarrow receivedData.getMax()$
12: **if** $min \leq receivedData.getEstimation() \leq max$ **then**
13: $continue\_process()$
14: **else**
15: $report\_to\_BS()$
16: **end if**
17: **else**
18: $report\_error\_on\_CH()$
19: **end if**

Algorithm 3 Detecting misbehavior during data reporting

Require: trusted value threshold δ

Ensure: Process is continued or trusted value is modified

```
1: sData ← sentData()
2: if data_is_sent then
3:    listOfSignature ← sData.getSig()
4: end if
5: rData ← receivedData()
6: if data_is_received then
7:    if listOfSignature.isExist(rData.getSig()) then
8:       listOfSignature.remove(rData.getSig())
9:    end if
10: end if
11: for sig in listOfSignature do
12:    if sig.isTimeOut() then
13:       //decrease trusted value of corresponding node
14:       tValue.decrease()
15:       if tValue.getValue() < δ then
16:          report_as_compromised_node()
17:       end if
18:    end if
19: end for
```

4. Security and Performance Analysis

After explaining our security suggestion for the standard and our new approach in providing secure and energy efficient CBRP for WSN, this section will show you the security and performance analysis either for MAC and Networks layer. The security analysis is explained based on the security requirements described on the previous chapter. In other hand, the performance analysis is described numerically using through simulation.

4.1 Security Analysis

Our scheme, secure and energy efficient communication scheme for Wireless Sensor Network has the following security feature.

4.1.1 MAC Layer Security

As we have claimed before, using timestamp together with source address as nonce can tackle all the problems related to MAC layer security because of the incomplete design of the nonce.

Same Key in Multiple ACL Entries

When the same key in multiple ACL entries is used, the same nonce attack can be happened because previous nonce can not guarantee unique value when it is used for multiple ACL entries list. As described before on this case the value of nonce may be reuse since source address, frame counter and key counter may be same for different packet. By using timestamp as part of nonce will guarantee that each time the value of nonce is computed it will result a different value. The value of timestamp will always change every time and it will impact to the changing of nonce's value. Therefore refer to Lemma 1, since the value of nonce is always unique even if used in multiple ACL entries, so $C = E(k, nonce) \oplus m$ will always give different value whenever it is computed. By result the confidentiality property of symmetric cryptography can be maintained.

Group Keying Model

Since now the timestamp will guarantee to give the different value each time it is needed, we do not need to give a strict rule for key in different ACL entry list. With the previous nonce design we can not have same key for different ACL entry because the nonce may be reused. After applying timestamp we do not care about the value of nonce since it is guaranteed to be always unique. Allowing to have the same key on different ACL entries implies that we can support group keying model. In this condition, set of nodes which belongs to certain group will have the same record on the key field of the ACL entry identified by theirs address. This implies that the nodes which have the same key record use group keying model.

Network Shared Keying Model

The problem appears because using network shared keying model with previous nonce design the replays protection can not well supported. Each node within network will record different state of replays counter and there is no way to synchronize this value. So when one node sends packet to certain destination after another node its replays counter may less than the current replays counter value recorded. When we use timestamp it will directly serve as replays counter. In network shared keying model time stamp will ensure that the packets sent after the other packet will always have greater value of timestamp (replay counter). So the value of replays counter on the packet sent later will be always greater than the value of replays counter of the previous packet. As a result the network shared keying model can be supported without any problem.

Loss ACL State

Using timestamp we do not need to store any ACL state when there is power failure. After recovery process we just need to synchronize again with the BS to get global clock and use the current timestamp to check the freshness. Once we get our local clock set we are sure that the current value will different and greater than the previous value before the power failure. Then the node's software can just repopulate the ACL table with the appropriate keys and the nonce state can be simply assigned with the current time after synchronization. So the same nonce (timestamp) will be never reused.

4.1.2 Network Layer Security

From network layer point of view, our protocol provides authentication, confidentiality, message integrity and freshness for uni-cast (CH-to-GW/BS), multi-cast (CH-to-nodes) and broadcast (initial advertisement message from CH) communication. 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 execute forgery attack to the network. The forgery attack is done by compromising one node and use the secret information to create a fake node. Since the authentication token uniquely determines 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.

The proposed scheme is employed with the CH misbehavior detection mechanism. In CBRP, CH is likely to be a target of attack 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 misbehavior detection mechanism is done by observing the activities of CH. The WatchDog helps in determining whether CH honestly follows the protocol in reporting the aggregation message to BS. At the same time, it also defends against black hole attack. We believe through those misbehavior observation, our protocol is robust enough against CH compromised attack. Our protocol is also secure against jamming attack. The nodes record the number of transmission of their neighbor (TR). By defining the threshold for the number of transmission allowed in certain interval of time the jamming attack can be avoided.

Secure multi-hop cluster-based routing protocol may suffer from ultimate shortcoming when the nodes within one or two hops from the CH are compromised [1]. When significant of these nodes have been compromised, all is lost. In our protocol, as long as there is still a node connected to the CH the delayed propagation mechanism can help other nodes to find alternative path to the CH. The PathRater avoids the message to be routed to the compromised nodes. In the worst case when all those nodes have been compromised, one or more legitimate nodes may introduce them selves as a new CH. And then, the process can be sequentially followed by cluster rebuilding process.

4.2 Performance Analysis

To verify our approach we conduct numerical experimentation using NS2. We simulated our protocol in two conditions *i.e.,* without and with authentication protocol. We refer to it as FREE and SFREE respectively. This is because we wanted to compare the simulation

result with HPEQ which does not support any security feature.

4.2.1 Simulation Setting

The complete simulation parameters are listed on Table 4.1. For simulation metric, we address the total energy usage, cluster rebuilding time and average packet delivery ratio. We compare our result with the HPEQ when the ttl=4 and p=0.2. We choose this value because on that condition most of the nodes can join the network, Figure 2.6. During the simulation we only consider the energy usage for transmission and reception. Since the energy usage for computation is very small so it is negligible in comparison to the energy for transmission.

Table 4.1: Simulation Parameters

Parameters	**Value**
Simulation time	500 second
Simulation area	100 m x 100 m
Topology	Fix-grid network topology
Transmission Range	10m
Number of nodes	100 node
Cluster round time	20 second
MAC type	MAC/802_15_4
Radio propagation model	Two ray ground
Antenna	Omni-antenna

For energy consumption we refer to [9] which defines the radio characteristic of sensor node and how transmit (E_T) and receive (E_R) energy are computed. The radio characteristic is listed on Table 4.2.

Table 4.2: Radio Characteristics

Operation	**Energy Dissipated**
Transmitter Electronics (E_{Tx}) Receiver Electronics (E_{Rx}) ($E_{Tx} = E_{Rx} = E_{elec}$)	50nJ/bit
Transmit Amplifier (ϵ_{amp})	100pJ/bit/m^2

The transmit energy is defined as follows:

$$E_T = E_{elec} * k + \epsilon_{amp} * k * d^2 \tag{4.1}$$

To receive the message, the node should dissipate:

$$E_R = E_{elec} * k \tag{4.2}$$

where:
k is the number of bit being sent or received (bit)
d is the distance of the node each other (m)

4.2.2 Simulation Result

As we specified before, we address three simulation metrics *i.e.,*

Total Energy Dissipation

Figure 4.1 shows the total energy dissipation by the network during 500 second simulation time. When we do not applied authentication protocol, our protocol(FREE) performs better than HPEQ. It can reduce energy usage almost 10%∼15% compared to HPEQ. In SFREE, since the size of each packet increases due to authentication protocol the total energy dissipation also increases. But total energy dissipation of SFREE is almost similar with HPEQ which does not support any security feature. Therefore we can claim that from security point of view our protocol has more advantages than HPEQ.

This can be achieved because our protocol does not need to flood CH notification message on each cluster round time. CH notification message only floods the network at the initial cluster formation.

Cluster Rebuilding Time

Figure 4.2 shows that the cluster rebuilding time using both FREE and SFREE is faster than HPEQ. SFREE requires more time than FREE, because of the additional computation required for encryption and decryption. This result implies that during the critical condition or when CH is not working properly, the time required to rebuild the cluster in our protocol is less than HPEQ.

Average Packet Delivery Ratio

In this experiment, we addressed the average of packet delivery ratio which is counted as: number of packet successfully delivered / total packet sent. We only compared the result

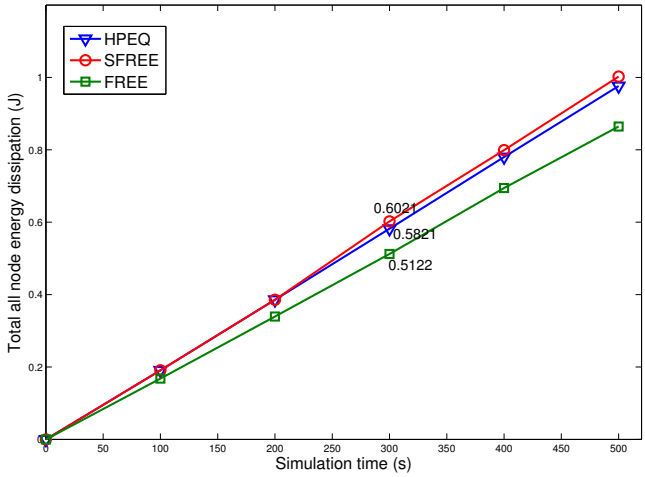


Figure 4.1: Total Energy Dissipation

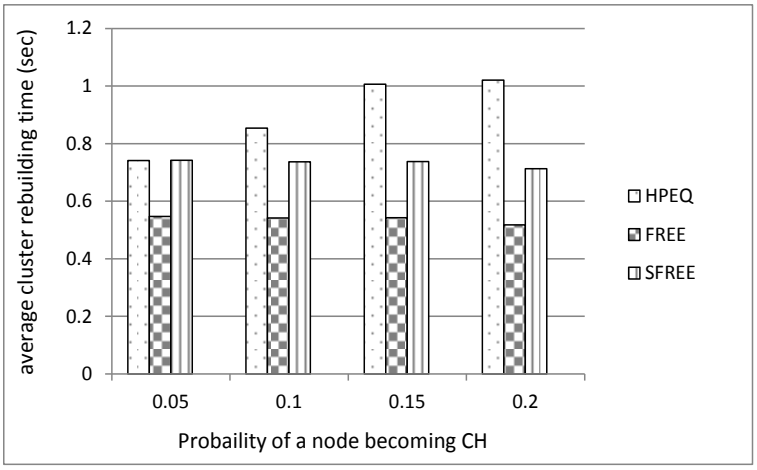


Figure 4.2: Cluster Rebuilding Time

of HPEQ and SFREE, since SFREE is employed with all security features previously explained. Figure 4.3 depicts the simulation result, in HPEQ as the number of malicious CH increases the average of packet delivery ratio drastically decreases. This is mainly because HPEQ does not employ any security mechanism. The malicious CH misbehaves by not in-

forming the aggregation result to the BS (in this case CH may just drop the packet from all cluster members). It implies that information from node is not successfully delivered to the BS. We encounter this problem in SFREE. The misbehavior detection mechanism can exclude the malicious CH from the network. Therefore, packet delivery ratio can be maintained high. In SFREE, the packet lost mostly because of the packet collision happened during cluster formation. But, this value is very small if compared to the total packet successfully sent along the simulation time.

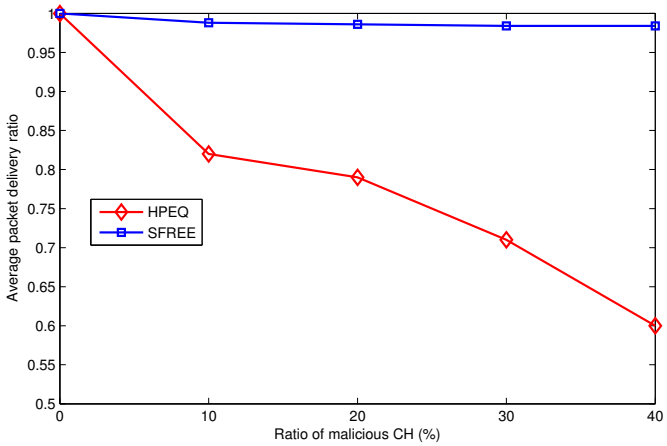


Figure 4.3: Average Packet Delivery Ratio

5. Conclusion and Future Work

WSN consists of houndred or even thousand of tiny sensor nodes which have limitation of computation power, energy resource, memory, *etc.* The growth of UCE trends has brought popularity to WSN application since it can provide services in real time manner. This popularity attracts the attackers to exploit the vulnerabilities of WSN application and get benefit for their own use. Thefore, for reliability issue the designer of WSN application should consider the security issue of WSN by applying security protocol. The challenge is the security protocol which usually requires extensive resource should be feasible to be applied in resource constraint devices (sensor node).

In order to increase the defense system against malicious user, the security protocol should be applied in each layer of WSN. In this thesis, we only consider the security within MAC layer defined in IEEE 802.15.4 standard and Network layer. In the MAC layer, we show that the nonce plays an important role for IEEE 802.15.4's security. The incomplete/immature design of nonce may lead to many serious problems currently faced by 802.15.4 compliant network. To tackle this problem, 802.15.4's nonce should be redesigned with the requirements: the value should be kept unique and sequentially increased even if used in multiple ACL entries. These requirements can be well supported when timestamp used for one of the nonce's component. From network layer point of view, as one of well known CBRP, HPEQ still has some drawbacks both in term of energy efficiency and security support. We have shown those drawbacks using our simulation. In addition, several previously proposed secure CBRPs have also suffered from several weaknesses. Therefore, they are inadequate to be used for mission critical application over WSN. We proposed a secure, fast rebuilding and energy efficient CBRP for mission-critical application over WSNs. It employs fast rebuilding mechanism to achieve the speed requirement and cluster-based authentication for reliability requirement on mission-critical application. The fast rebuilding is achieved by sending the CH's advertisement message only to certain neighbor nodes. In other hand, the cluster-based authentication is dedicated to get CK which will be used to secure all communication within the cluster. Meanwhile, the misbehavior detection strengthen the protocol so illegitimate access can be detected. According to our simulation result, our protocol consumes a reasonable amount of energy and can reduce 30% of average cluster rebuilding time compared to HPEQ.

For future work, it is good if an efficient key establishment process for WSN can be

addressed. This mechanism is required since we do not want to keep the global key forever in each node. The global key may be deleted as long as we can provide another way to establish a new key so either node and BS can recognize it. This is important when there is a new node needs to join the network. One solution that may be used is by loading the node with prior knowledge about what the next joining node will be. The information may contain partial key of joining node. When the node start to join the network the BS can supply another partial of the key so it can be used to authenticate the new node properly.

요 약 문

무선 센서네트워크에서의 안전한 에너지 효율적인 통신 방법

유비쿼터스 컴퓨팅이 필요해짐에 따라, 이는 우리 일상에서 이러한 컴퓨팅 능력을 향상시키기 위한 새로운 기술의 발전을 이끌었다. 퍼베이시브 컴퓨팅을 가능하게 해주는 기술 중 널리 알려진 것 중 한 가지는 무선 센서 네트워크이다. 무선 센서 네트워크는 온도, 불, 소리, 압력, 진동, 움직임, 오염, 국경 감시와 같은 물리적 혹은 환경적 상태를 빠르고 효율적으로 감지하는 다수의 분산된 자동화 된 센서와, 이렇게 수집한 데이터를 협동하여 보내는 특정한 대상(베이스 스테이션)으로 이루어진다. 무선 센서 네트워크는 보통 수백 혹은 수천의 센서 노드로 이루어지는데 이는 위와 같은 목적을 위해 넓은 범위의 영억을 감시 범위에 포함시키기 위해서이다. 이러한 센서 노드들은 값이 저렴하고 그에 따라 태생적으로 한정된 에너지 자원과, 컴퓨팅 능력 및 메모리 공간이라는 제약이 따른다. 이와 같이 자원이 제약 된 기기를 사용해 어플리케이션을 전송하기 위해 사용자는 이러한 자원의 제약과 성능 요구 사항 간의 균형을 맞춰야 한다.

화재 경보, 방사능 유출, 정찰 임무와 같은 치명적인 임무를 무선 센서 네트워크 상에서 수행하는 어플리케이션의 경우 빠르고, 신뢰할 수 있고, 내고장성을 갖춘 라우팅 프로토콜이 필요하다. 그렇지 않다면 해당 어플리케이션은 본연의 임무를 수행하지 못하며 예상치 못한 오류를 일으킬 수 있다. 현존하는 라우팅 프로토콜들은 보안 이슈들을 고려하고 있지 않으며 시스템의 안전성 또한 고려하고 있지 않다. 이 논문에서 우리는 치명적인 임무 수행을 위한 안전하고 에너지 효율적이며 신속한 재구축이 가능한 클러스터 기반의 라우팅 프로토콜을 제안한다. LEACH나 HPEQ와 같은 이전의 프로토콜과 비교했을 때, 우리의 프로토콜은 클러스터 기반의 인증과 관리 메시지의 전파 지연을 이용하여 프로세싱 시간을 단축시키고 에너지 소모를 감소시키는 한편, 안정성 또한 향상시켰다. NS2를 활용한 시뮬레이션에 의하면, 우리의 프로토콜은 HPEQ와 비교하여 에너지 소모를 의미있게 감소시키면서 프로토콜 재구축 시간을 30% 가량 감소시킨다.

References

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

[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] P. Baronti, P. Pillai, V. W. C. Chook, S. Chessa, A. Gotta and Y. F. Hu. ""Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards". *Journal of Computer Communications*, vol. 30, Issue 7, Elsevier, pp. 1655-1695, 2006.

[4] Wikipedia, "IEEE 802.15.4-2006", *http://en.wikipedia.org/wiki/IEEE_802.15.4-2006*, accessed January 2011.

[5] IEEE, "Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specification for Low-Rate Wireless Personal Area Networks (LR-WPANs)" IEEE Computer Society, 2006.

[6] N. Sastry and D. Wagner, "Security Consideration for IEEE 802.15.4 Networks", Proceedings of the 3rd ACM workshop on Wireless security, ACM Press, pp. 32-42, 2004.

[7] X. Yang, C. Hsiao-Hwa and S. Bo, "Security Services and Enhancement in the IEEE 802.15.4 Wireless Sensor Networks", GLOBECOM, IEEE, pp. 5-9, 2005.

[8] H. Alzaid, E. Foo and J. G. Nieto. "Secure Data Aggregation in Wireless Sensor Network: a survey". *Proc. 6th Australian Information Security Conference*, CRPIT vol. 81, 2008.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

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

[14] 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.

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

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

[17] 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.

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

[19] C. Mallanda, S. Basaravaju, A. Kulshrestha, R. Kamman, A. Durresi, and S.S. Iyengar. "Secure Cluster based Energy Aware Routing for Wireless Sensor Networks". *in Proceeding of ICWN*, pp.461-466, 2004.

[20] J. Hill, M. Horton, R. Kling, and L. Krishnamurthy. "The Platforms Enabling Wireless Sensor Networks". *Communications of the ACM - Wireless Sensor Networks*, vol. 47 no. 6, June 2004.

[21] R. Srinath, A. V. Reddy, R. Srinivasan, "AC: Cluster Based Secure Routing Protocol for WSN". *Third International Conference on Networking and Services*, IEEE, pp. 45-47, 2007.

[22] K. Piotrowski, P. Langendoerfer, S. Peter, "How Public Key Cryptography Influences Wireless Sensor Node Lifetime", Proceedings of the fourth ACM workshop on Security of ad hoc and sensor networks, USA, pp. 169–176, 2006

[23] V. Andrea, P. Gianni, "Rijndael for Sensor Networks: Is Speed the Main Issue?", Electronic Notes in Theoretical Computer Science, Elsevier, 2007

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

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

[26] 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, pp.339-352, 2006.

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

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

Acknowledgement

First of all, I would like to thank my advisor, Prof. Kim, Kwangjo for his guidance and support throughout the course of my study in KAIST. I would like to thank the other member of my master advisory committee: Prof. Yoon, Hyunsoo and Prof. Kim, Myungchul. Their insightful comments and suggestion have greatly helped me to improve my Master Thesis.

I also had privileged of working with many kind and enthusiastic people here in Cryptology and Information Security Laoratory at KAIST. I have learn so many things from them during my time here in Korea. I am very thankful for their research advice, encouragement, and support. I would like to personally thank Divyan M. Konidala, Jangseong Kim, Kyusuk Han, Dang Nguyen Duc, Hyunrok Lee, Zeen Kim, Junhyun Yim, Doyoung Chung, Yi Jae Park and Ji Yeong Hong.

I have been also very lucky to be surrounded by great community of Indonesian student (KAIST-INA) here in KAIST. We have done a lot of interesting activities here such as: futsal, cooking, hiking, etc. Those are of course will be unforgettable memories for me. I am really proud to be captain of KAIST-INA futsal team and bring the team to beat the other teams here in KAIST. Such a fun activities have helped me a lot to release my stress within my hectic time in the Lab.

Most importantly, I thank to God (Ida Sang Hyang Widhi Wasa) for His grace and for leading me through at all the times. I believe He always watches me all over time and blesses me in every step that I take.

Last but not the least, I want to express my deepest gratitude to all the members of my family; My mother: Ni Luh Suastiasih, my father: I Wayan Pageh, my brother: I Gde Harta Wijaya, my sister-in-law: Dewa Ayu Eka whom I have not met in person until now and others familly member that I can not mention one by one. Their prayers, love, support and constant encouragement are the source of my inspiration and immense happiness. I therefore dedicate this thesis to all my family members.

Curriculum Vitae

Name : Made Harta Dwijaksara

Date of Birth : September 01, 1985

Address : 305-701 대전광역시 유성구 대학로(구성동373-1)한국과학기술원 나눔관 2105호

E-mail : made.harta@kaist.ac.kr

Educations

2009. 9. – 2011. 8. KAIST, Computer Science (M.S.)

2008. 9. – 2009. 2. Ajou University (IT-ISIP Program)

2004. 8. – 2008. 7. ITB, Informatics Engineering (B.S.)

2001. 7. – 2004. 6. SMU N 1 Bangli, Bali (High School)

Career

2007. 6. – 2007. 8. PT. Mitrais - IT. Consultant (Internship)

2007. 8. – 2007. 12. Strategy Algoritmic, Undergraduate Teaching Assistant, Bandung Institute of Technology (ITB)

2010. 2. – 2010. 5. Introduction to Information Security, Undergraduate Teaching Assistant, KAIST

Publications

1. **Made Harta Dwijaksara**, Doyoung Chung, Yi Jae Park, Jangseong Kim and Kwangjo Kim, "Secure, Fast Rebuilding and Energy Efficient Routing Protocol for Mission Critical Application over Wireless Sensor Networks", 2011 Symposium on Cryptography and Information Security (SCIS 2011), Jan. 25-28, 2011, Kokura, Japan.

2. Doyoung Chung, **Made Harta Dwijaksara**, Yi Jae Park, Jangseong Kim and Kwangjo Kim, "An Efficient and Privacy Preserving Authentication Protocol for HAN", Symposium on Cryptography and Information Security 2010 (SCIS 2011), Jan. 25-28, 2011, Kokura, Japan.
3. Divyan M. Konidala, **Made Harta Dwijaksara**, Kwangjo Kim, Dongman Lee, Daeyoung Kim, Byoungcheon Lee, and Soontae Kim, "Resuscitating Privacy-Preserving Mobile Payment with Customer in Complete Control", Journal of Personal and Ubiquitous Computing (PUC). Accepted on 06/10/2010, to appear 2011.
4. Divyan M. Konidala, Kwangjo Kim, **Made Harta Dwijaksara**, and Daeyoung Kim, "Diffusion-Confusion based Light-Weight Security for Item-RFID Tag-Reader Communication", Journal of Internet Technology (JIT). Accepted on 08/09/2010, to appear 2011.
5. **Made Harta Dwijaksara**, Kwangjo Kim, "Detecting DDoS Attack in Mobile Ad Hoc Network Based on Network Density Awareness", CISC-Summer, South Korea, 2010.