A Thesis for the Degree of Master

# A Study on Localized Multi-user Broadcast Authentication Protocol for Wireless Sensor Networks

Sungjune Yoon

School of Engineering

Information and Communications University

2008

# A Study on Localized Multi-user Broadcast Authentication Protocol for Wireless Sensor Networks

# A Study on Localized Multi-user Broadcast Authentication Protocol for Wireless Sensor Networks

Advisor : Professor Kwangjo Kim

by

Sungjune Yoon

School of Engineering
Information and Communications University

A thesis submitted to the faculty of Information and Communications University in partial fulfillment of the requirements for the degree of Master of Science in the School of Engineering

Daejeon, Korea
December 18. 2007
Approved by

Professor Kwangjo Kim
Major Advisor

# A Study on Localized Multi-user Broadcast Authentication Protocol for Wireless Sensor Networks

Sungjune Yoon

We certify that this work has passed the scholastic standards required by the Information and Communications University as a thesis for the degree of Master

December 18. 2007

Approved:

Chairman of the Committee
Kwangjo Kim, Professor
School of Engineering

Committee Member
Young-Hee Lee, Professor
School of Engineering

Committee Member
Byoungcheon Lee, Assistant Professor
Dept. of Information Security, Joongbu University

M.S. Sungjune Yoon
20062075

**A Study on Localized Multi-user Broadcast Authentication Protocol for Wireless Sensor Networks**

School of Engineering, 2008, 46p.
Major Advisor : Professor Kwangjo Kim.
Text in English

# Abstract

Wireless Sensor Networks (WSNs) are one of the core techniques for the upcoming ubiquitous environment. A WSN is an *ad-hoc* network which consists of hundreds to thousands of tiny resource-constrained sensor nodes and several powerful Base Stations (BSs). WSNs will be widely used for autonomous remote data acquisitions in many different applications from battle field surveillance to building maintenance [7].

Even though, WSNs have many similarities with existing *ad-hoc* networks, there are many differences; more constrained resources, large-scale deployment, and node compromise. These differences make existing security protocols for *ad-hoc* networks impractical in WSNs [9].

In WSNs, Broadcast authentication allows sensor nodes to verify the authenticity of broadcasted messages including commands and queries. Due to above mentioned differences, existing broadcast authentication protocols for *ad-hoc* networks or Internet are impractical as they require computation intensive public key operations. To conserve the energy of sensor nodes, most protocols for WSNs utilized delayed key disclosure[1].

[1]Send a message first with a message authentication code then release the associated key.

The problem of these protocols is not only legitimate senders could broadcast messages but also adversaries easily flood malicious messages into a network by which they could easily destruct the operations of the network by overflowing the buffer of sensor nodes.

Recently, some researchers have proposed several multi-user broadcast authentication protocols which allow legitimate mobile users to directly disseminate their messages into the network in an authenticated manner. However, most of them are based on public key cryptography (PKC) which consumes a lot of energy of sensor nodes to verify the messages. Although, PKC can provide immediate authentication of messages (thus, it can prevent the network from flooding attacks), it could not prevent the network from careless and excessive message dissemination by legitimate users. Therefore, we strongly believe that network users are allowed to broadcast their messages only into their surroundings not into the entire network.

In this thesis, we propose two security protocols; one for secure clustering with enabling *in-cluster* broadcast authentication and another for user authentication. Our clustering protocol provides an efficient localized broadcast authentication to Cluster Head (CH) which manages a cluster as well as secure Node-to-CH communication, and our user authentication protocol provides an efficient secure channel establishment between a node and a user under the assumption that the user can directly communicate with BS.

With our protocols, we finally explain how to combine these protocols to allow legitimate users to locally disseminate their messages into several clusters which are in the communication range of the users. Since our protocols utilize only secret key cryptography including hash, message authentication code, and pseudo random number generator, we believe that our combination outperforms the other PKC-based multi-user broadcast authentication schemes.

# Contents

# List of Figures

# List of Tables

# List of Abbreviations

**BA** Broadcast Authentication

**BS** Base Station

**CH** Cluster Head

**KP** Key Pool

**LEACH** Low Energy Adaptive Clustering Hierarchy

**MAC** Message Authentication Code

**MBA** Multi-user Broadcast Authentication

**OHC** One-way Hash Chain

**PKC** Public Key Cryptography

**SKC** Secret Key Cryptography

**WSN** Wireless Sensor Network

**PRNG** Pseudo-Random Number Generator

**UA** User Agent

# List of Notations

$A_x$ Communication range of $x$

$AL_x$ Access control list of $x$

$CH_x$ Cluster head node of $G_x$

$D_{SET}$ Cluster set-up duration

$D_{STD}$ Cluster steady-state duration

$E_{K_{x,y}}(*)$ Secret key-based encryption with a key, $K_{x,y}$

$G_i$ $i^{th}$ group (or cluster) in the network

$H(msg)$ Hash value of $msg$

$ID_x$ Identity of $x$

$K_{x,y}$ Shared secret key between $x$ and $y$

$K_x$ $x^th$ key in a key pool

$KP_x$ Key pool of $G_x$

$MAC_x(msg)$ Message authentication code of $msg$ using $x$ as a key

$MAX_B$ Maximum broadcasting number in a round

$MAX_D$ Maximum broadcasting delay in a cluster

$N_x$ $x^{th}$ sensor node in WSN

$R$ Random number

$T_{Exp}$ Expiration time

$T_x$ Current time of $x$ (local time stamp)

$TK_{x,y}$ Ticket for $x$ and $y$ generated by BS

$\eta_x^y$ $x^{th}$ sensor node in $G_y$

$\varpi$ Total number of sensors in $G_x$

$\Rightarrow$ Secure channel

$\rightarrow$ Insecure channel

# I. Introduction

## 1.1 Overview

Wireless Sensor Networks (WSNs) are one of the core techniques for the upcoming ubiquitous environment. A WSN is an *ad-hoc* network which consists of hundreds to thousands of tiny resource-constrained sensor nodes and several powerful Base Stations (BSs). Sensor nodes sense environmental changes (e.g., temperature, seismic wave, and so forth.) from their immediate surroundings and perform very simple computation on the sensed data and transmit the data to BS via their intermediate nodes. BS is a powerful and well secured device which performs delicate computation on the received data from sensor nodes and provides it to external users via Internet. WSNs will be widely used for autonomous remote data acquisitions in main different applications from battle field surveillance to building maintenance. Figure 1.1 shows a simple WSN and its applications.

When sensor nodes are deployed in an unattained and hostile environment, security becomes important issues since they are subject to different types of attacks. For example, an adversary can easily eavesdrop the traffic, impersonate the sensor nodes, or intentionally inject misleading information into WSNs [11]. Therefore, the communication should be encrypted and authenticated.

WSNs have many similarities with *ad-hoc* networks but also have following differences; more constrained resources, large-scale deployment, and node[1] compromise. Those differences make existing security proto-

[1]We will use the terms senor, node, and sensor node interchangeably throughout this thesis.

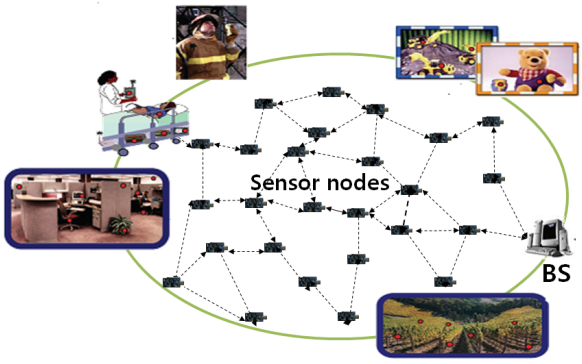


Figure 1.1: WSN and its Applications

cols for *ad-hoc* networks impractical in WSNs [9]. Table 1.1 shows the specification of sensor nodes widely used in the WSN research field.

In WSNs, Broadcast Authentication (BA) which is one of the most important security mechanisms for secure software updates, network-wide commands, and queries dissemination allows sensor nodes to verify the authenticity of broadcasted messages. As mentioned before, due to the scarce resources of sensor nodes, the large-scale deployment, and node compromise, existing BA protocols for *ad-hoc* networks or Internet are impractical as they require computation intensive public key operations.

To conserve the scarce energy of sensor nodes, most BA protocols for WSNs utilized delayed key disclosure[2]. The problem of these protocols is not only legitimate senders could broadcast messages but adversaries also easily flood malicious messages into the network by which they could easily destruct the operations of the network by overflowing the buffer of sensor nodes. To defense a WSN from such flooding attacks, we have to limit the range of broadcasting.

Recently, Ren *et al.* have proposed several multi-user broadcast au-

[2]A sender broadcasts a message first with a message authentication code then releases the associated key.

Table 1.1: Sensor Node Specification [27]

| Sensor Node | WeC | Rene | Dot | Mica | Mica2 | MicaZ | iMote2 |
|---|---|---|---|---|---|---|---|
| Release | 1998 | 1999 | 2001 | 2002 | 2003 | 2004 | 2006 |
| Clock (MHz) | 4 | | | | 7.37 | | 13-416 |
| CPU | Atmel 90LS8535 | | Atmel Atmega 163 | Atmel Atmega 103L | Atmel Atmega 128L | | PXA271 XScale |
| RAM | 0.5KB | | 1KB | 4KB | | | 32MB |
| Program Memory | 8KB | | 16KB | 128KB | | | 32MB |
| Wireless Module | RFM TR1000 | | | | Chipcon cc1000 | Chipcon cc2420 | |
| Data Rate(KB) | 10KB | | | 40KB | 38.4KB | 250 | |

thentication protocols which allow legitimate users to directly broadcast their messages into WSNs under the assumption that Public Key Cryptography (PKC) will become viable even in tiny sensor nodes. Allowing a number of legal users to disseminate any messages into the entire WSNs, however, could easily disrupt the operation of the networks, so we still have to limit the broadcasting range of the users.

## 1.2 Our Goals

Our main goal is to allow only legitimate users to efficiently disseminate their queries or data to the sensor nodes resided at their surroundings in an authenticated manner.

To achieve this goal, we provide two security protocols; 1) a secure clustering protocol with enabling *in-cluster* broadcast authentication by which we can localize the broadcasting range of users in a cluster and 2) a user authentication protocol which is used for authenticating users

before allowing them to broadcast their messages into a cluster.

Combining these two protocols, we can finally achieve our goal; i.e., whenever a user wants to broadcast his/her query or command to the surroundings, the user first has to authenticate him/herself to a cluster head[3](CH), and then the CH disseminates the query on behalf of the user to the other nodes in its cluster.

## 1.3 Outline

The remaining parts are organized as follows: In Chapter II, we explain previous BA protocols, clustering schemes, and user authentication mechanisms. In Chapter III, we provide some background information adopted in our protocol. In Chapter IV and Chapter V, we describe our clustering protocol and user authentication scheme with their security analysis respectively. At the end of Chapter V, we also describe how to combine our protocols to support localized multi-user broadcast authentication. In Chapter VI, we make a conclusion with our future work.

[3]A cluster head manages the other nodes (or member nodes) in a cluster. It usually aggregates data received from its member nodes and sends the aggregated data to BS

# II. Previous Works

## 2.1 Broadcast Authentication Protocols

Earlier studies mainly focused themselves on secret key-based broadcast authentication. $\mu$TESLA [28] is the first broadcast authentication protocol designed for WSNs. $\mu$TESLA provides source authentication and message integrity by utilizing a one-way hash chain (OHC) [19] and loosely-coupled time synchronization between sensor nodes and BS. The $\mu$TESLA is a very efficient broadcast authentication mechanism, but it has very limited scalability because of its unicast-based parameter distribution needed for adding new sensors and is subject to malicious message flooding attacks. To cope with this problem, multi-level $\mu$TESLA [21] were proposed to enhance $\mu$TESLA using hierarchical approaches to support large-scale networks and to prolong the lifetime of broadcast authentication (the lifetime of $\mu$TESLA is determined by the length of an OHC). L-TESLA [10] also provides an alternative solution for this problem by localizing the authentications of broadcasted messages. It divides a large network into small number of clusters (or groups) where few trusted and better secured nodes are deployed. A trusted node coordinates broadcast authentication in a cluster. BABRA [36] removes the requirement of time synchronization and hash key chain needed in $\mu$TESLA, and provides infinite lifetime of broadcast authentication, but it requires one additional hash value to remove the hash key chain.

However, above mentioned schemes do not support a large number of broadcast users since each user's broadcast parameters must be stored into all sensor nodes which usually have a small storage. T-TESLA [22]

provides a solution using Merkle hash tree [23], but it is only suitable when all the users could be pre-determined before the deployment of WSNs since the Merkle hash tree should be changed and re-distributed whenever a user is added or deleted.

Due to the advances in sensor nodes, public key cryptography (PKC) has become a good solution for providing security services even in a tiny sensor node [29]. The main advantages are to construct simple protocol and to authenticate a message immediately. Based on these advantages, Ren et al. [30, 31] proposed several MBA schemes which focused on reducing the number of PKC operations and increasing the number of users by utilizing Merkle hash tree and Bloom filter [24]. In order to authenticate a message in their approaches, however, all sensor nodes have to verify the signature of the message; thus, it could more easily exhaust the scarce energy of sensor nodes than secret key-based approaches. In the other words, we should restrict the use of PKC to minimum. Sluice [18] combines a digital signature scheme and an OHC to efficiently broadcast a bulk of messages usually used in program updates or reprogramming, but it is not suitable for broadcasting a large number of *small-size* messages. Benenson [5] proposed a PKC-based user authentication scheme which could be used to establish a bunch of secure channels between a user and his neighbor sensor nodes, but did not provide any specific method to broadcast user's messages into WSNs. In addition, AQF-pass [4] utilized multi-MAC approach to broadcast messages but has false positive nature.

## 2.2 Secure Clustering Protocols

Sensors are usually battery-powered devices so they should be recharged manually. However, it is very difficult to manually manage each sensor node due to the large-scale deployment. Thus, the energy efficiency is

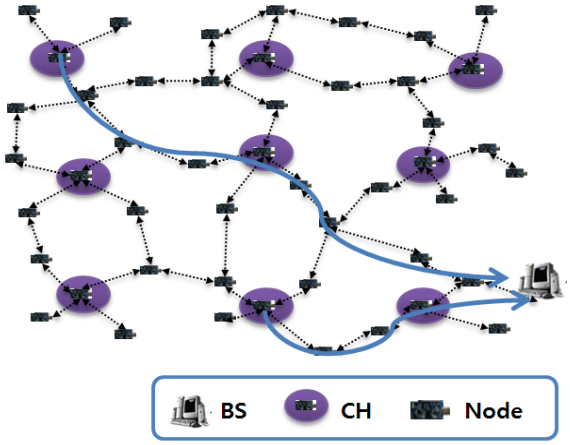


Figure 2.1: Cluster-based WSN

the most important factor in WSNs. Data transmission, above all, is the most energy consuming operation.

If sensors individually transmit their data to BS, intermediate nodes between the sensors and BS consume a lot of their scarce energy to route others' data. To reduce routing overhead, a WSN should be organize into clusters. In each cluster, there is a cluster head node ($CH$) which is responsible for managing its cluster, aggregating data received from its member nodes, the other nodes in the cluster, and transmitting the fused data to BS. Figure 2.1 shows a simple cluster-based WSN architecture.

There are a lot of clustering protocols, such as LEACH [25], PEGASIS [20], TEEN [15], *etc.* Among them, LEACH is the most frequently used clustering protocol. It uses a distributed clustering technique and self-reorganization of the sensor nodes to prolong the life time of a WSN. However, LEACH is vulnerable to a number of attacks including eavesdropping, jamming, spoofing, and so forth. To cope with these problems, SLEACH [14], SecLEACH [26], and GS-LEACH [3] were proposed to defense LEACH from such a malicious attacks. SLEACH provides CH authentication which means only legitimate nodes become CHs us-

ing $\mu$TESLA. However, it delays the set-up phase and is difficult to support large-scale WSNs since original $\mu$TESLA also has the scalability problem. SecLEACH provides secure Node-to-CH communication by utilizing a random key pre-distribution scheme (we will explain it in the next chapter), but there are many orphan nodes which means that they do not participate any clusters and the constructed clusters are not efficient. GS-LEACH was proposed to reduce the above mentioned problems of SecLEACH by exploiting deployment knowledge. We will provide the detailed description of GS-LEACH in the following chapter.

Above mentioned secure variants of LEACH do not provide an *in-cluster* (or localized) broadcast authentication mechanism which is necessary to control the operation of member nodes. In the above protocols, when a CH wants to broadcast a command or query to its member, the CH has three choices; 1) broadcasting the command or query without authentication, 2) individually sending the command or query to each member node which is protected by a shared key between the CH and each node, 3) requesting BS to broadcast its command or query.

First one is vulnerable to forgery attacks, any adversaries can broadcast their malicious commands our queries. Second one requires a lot of energy, and third one is vulnerable to DoS attacks since all the secure variants of LEACH uses original $\mu$TESLA which is used in the entire network, so any attackers can flood malicious messages into the network which results in overflowing the memory of sensor nodes.

## 2.3 User Authentication Protocols

In [4, 5], Benenson *et al.* used a public key-based certificate for verifying the source of a query, under the assumption that public key operation is viable even in the resource-constrained sensor nodes. However, public key operation is much slower than secret key operation; while launching

a *DOS* attack, an attacker can easily exhaust the limited energy of sensor node. Wong *et al.* [34] and Wang and Li [33] proposed user authentication schemes, which exhibit the same weakness as mentioned above, because they also used public key operation in their schemes. Banerjee and Mukhopadhyay [2] applied a random polynomial key predistribution scheme [1] to verify the legitimacy of a user, but did not consider the user compromise *i.e.*, by stealing a legitimate user's device, an attacker can get all the information from any sensor node at anytime.

Although Zhang *et al.* [35] have proposed a user authentication scheme that is resilient against the user compromise, the trajectory of a user has to be predetermined in their scheme. It is suitable for network management (in this case, BS can predict the trajectory of the user), but inadequate for normal users whose trajectory are difficult to predict. Moreover, all the above mentioned schemes reveal the information of the user where the ID of the user is broadcasted as a plaintext. Since an attacker can easily eavesdrop the broadcasted ID, the attacker can track the whereabouts of the user. It could violate the privacy of the user.

# III. Preliminaries

In this chapter, we explain two basic protocols widely adopted in many security mechanisms in WSNs and one new secure clustering protocol which will be modified to provide localized broadcast authentication.

## 3.1 One-way Hash Chain (OHC)

An OHC [19] is a chain of hash values generated by repeatedly applying a one-way hash function, $H()$, on a random number $R_m, R_i = H(R_{i+1})$ for $i = m-1, .., 0$. It is computationally impossible to find $R_i$ from $H(R_x)$ where $0 \leq x \leq i$. This OHC is used to generate a chain of keys which will be used in Message Authentication Code (MAC) computations[1]. To bootstrap broadcast authentication, a sender should distribute the initial key commitment, $R_0$, to all receivers in a secure manner. Before broadcasting a message, the sender uses a key, $R_i$ where $i > 0$, in the chain which is not disclosed yet to generate a MAC and broadcasts the message with the MAC. Some time later, the sender releases the key which is authenticated by $R_0$. Most OHC-based BA schemes have small communication overhead (one or two hash (or MAC) values per a message), but the authentication of messages is delayed until the related keys are disclosed.

## 3.2 ID-based Key Pre-distribution

In 2002, Eschenauer and Gligor proposed a random key pre-distribution scheme [13]. This scheme has three phases.

[1]MAC is a short piece of information used to provide the integrity and authenticity of a message.

1. **Key Pre-distribution Phase** Prior to deployment, BS generates a large key pool, $KP$ which contains an amount of keys with their associated ID. Each node randomly picks $w$ distinct keys with their ID from $KP$, and stores them into its memory. The number of keys in $KP$ and $w$ are chosen such that any two nodes share at least one key with some probability $p$. Let $s$ denote the number of keys in $KP$, then $p$ is calculated by the following equation:

$$p = 1 - \frac{\binom{s}{w}\binom{s-w}{w}}{\binom{s}{w}\binom{s}{w}} \tag{III.1}$$

2. **Shared Key Discovery Phase** After deployment, each node tries to find a common key with its neighbor nodes by sending all the IDs of its keys to the neighbors. If two neighboring nodes have a common key (by checking all the received IDs), the key will be used to secure the communication channel between these nodes.

3. **Path Key Establishment Phase** After the key discovery phase, nodes can construct path keys with their neighbor nodes which they do not share keys with via their other neighbor nodes they already shared a key with.

$s$ and $w$ are important factors which determine the connectivity and the resilience of the scheme. Connectivity is defined as the probability *i.e.*, $p$ that any two nodes share at least one key. Resilience is defined as the fraction of the compromised communication links after a certain number of nodes are captured by attackers. For example, if $s$ is one, the scheme provides a high connectivity, but it is not resilient against node capture attack as an attacker can compromise the entire network by capturing only one node. On the other hands, if $s$ is very large, resilience becomes much better, but connectivity of the network becomes low [12].

For instance, as indicated in [13], if $s$ is 100,000, $p$ is only 0.33 even if each node selects 200 keys from this key pool.

In the shared key discovery phase, each node has to broadcast all the IDs of its keys which in turn produces a lot of communication overhead. To reduce this communication overhead, Zhu *et al.* [37] utilized a Pseudo-Random Number Generator (PRNG).

At the key pre-distribution phase, instead of the random selection of $w$ number keys, each sensor node first generates $w$ distinct index values using PRNG with its ID as a seed value, then stores $w$ keys associated with the generated index values into its memory.

At the shared key discovery phase, now, each sensor only needs to broadcast its ID instead of all the IDs of keys it stores. By repeatedly generating index values (upto $w$) using PRNG with a node ID as a seed value, each node can find a shared key if it exists.

## 3.3 Grid-based Secure-LEACH

Grid-based Secure LEACH [3] (GS-LEACH) is a secure variant of LEACH. It utilizes deployment knowledge to efficiently support secure Node-to-CH communication. All the LEACH-*like* clustering protocols work in rounds and have two pre-determined phases in each round; a short set-up phase and a long steady-state phase. At each set-up phase, nodes are self-configured into clusters. After the set-up phase is finished, there is a long steady-state phase. Additionally, GS-LEACH has a key pre-distribution scheme before the deployment of sensor nodes.

Prior to deploying $l$ sensor nodes in the field, BS divides sensor nodes into $g$ group, $G_i$ where $1 \leq i \leq g$. Each group contains $l/g$ sensor nodes. Then, BS generates $g$ distinct key pools $(KP_1, ..., KP_k)$ of the same size from a large size key pool $KP$. Each node in $G_i$, $\eta_z^i$, is assigned to a unique ID and then stores $w$ distinct keys form $KP_i$ according to the

above mentioned ID-based key pre-distribution scheme. In addition, each node is also given one unique key for communication with the BS. This key is used when a node becomes a CH and sends data to the BS.

After that, the field of interest is also divided into $g$ grids. At each grid, one group of nodes is deployed randomly. The nodes deployed in each grid communicate within the group, and are the only legitimate nodes to form a cluster at the grid. Thus, all the nodes in a group would not communicate with the other nodes in the other groups. This ensures that even if several nodes in a group are captured by an attacker, security of the rest of the network will not be compromised.

At the set-up phase, each node in a group individually decides whether or not to become a CH with some probability, such that there is only one CH per a grid, and then the self-elected CH in a grid broadcasts a short message and its ID to the other nodes in the group. The nodes in the group which have a common key with the CH respond by sending a join-request message and the index value of the common key to the CH. The nodes which do not share a key with the CH in the group have themselves to sleep for this round and will participate in the next round.

After receiving all the join-requests, the CH make a TDMA schedule and unicasts a confirmation message to each node who has joined as a member in the group with its time information. This allows the members to turn off their radio module at all times except their scheduled time intervals, so they can save their scarce energy. The set-up phase is followed by a long steady-state, where the member nodes sense environmental data from their surroundings and send the data to the CH with a MAC generated by their shared key with the CH. Then, the CH first verifies the MACs, aggregates the data, and sends the result of the aggregation to BS using its shared key. Figure 3.1 shows an example of GS-LEACH where $g = 9$ and $l = 45$.

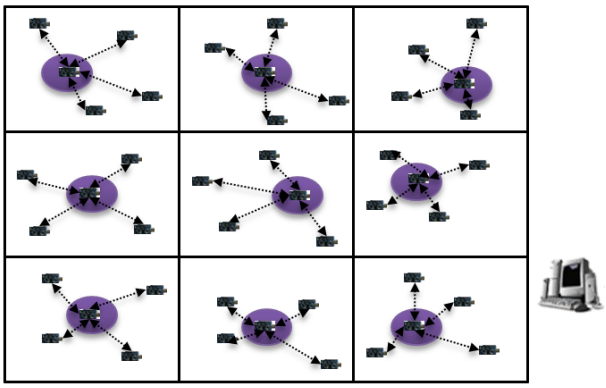

Figure 3.1: GS-LEACH

### 3.3.1 Connectivity

The connectivity in a cluster depends on the number of groups ($g$), the number of keys in the $KP$ ($s$), and the number of keys stored in a node ($w$). The connectivity of a cluster is calculated by the following equation:

$$p_{conn} = 1 - \frac{\binom{s/g}{w}\binom{s/g-w}{w}}{\binom{s/g}{w}\binom{s/g}{w}} \tag{III.2}$$

Figure 3.2 shows the connectivity of a cluster where $s = 100,000$ and $l = 1,000$

### 3.3.2 Resilience

Let $n$ denote the number of compromised nodes, then the resilience in a cluster of GS-LEACH could be computed by the following equation:

$$p_{compromise} = 1 - (1 - \frac{w}{s/g})^n \tag{III.3}$$

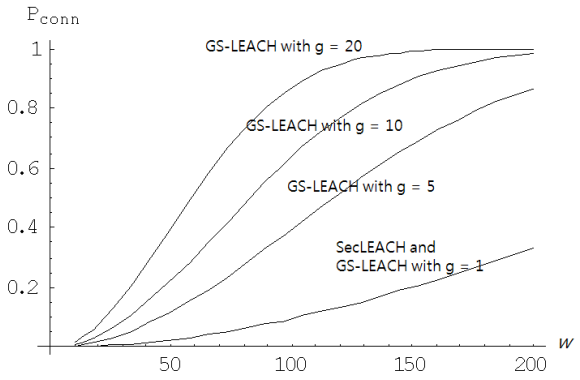


Figure 3.2: The Connectivity of GS-LEACH [3]

Since any two nodes resided in different groups does not share any keys, compromising several nodes in a cluster does not affect the other groups.

# IV. Broadcast Enabled GS-LEACH

## 4.1 Motivation

We have noticed that there is no local broadcast authentication mechanism in the secure variants of LEACH. In each cluster, a cluster head should both aggregate data as well as disseminate queries and commands to its members to control their operation [8]. Some protocols [6, 8] use a cluster-key approach which means that all the members in a cluster share a secret key, but these are vulnerable to node compromise attacks. If an attacker captures a node in a cluster, the attacker could broadcast any forged messages into the cluster in an authenticated manner. In this chapter, we slight modify GS-LEACH to bootstrap *in-cluster* broadcast authentication.

## 4.2 Assumption

We only consider a homogenous WSN which consists of a large number of sensor nodes with the same computing power and energy except BS. All the variants of LEACH assume that all the nodes can directly send data to the BS with their highest transmission power, but to conserve the scarce energy of nodes, they hardly communicate with the BS. Only CHs send data to the BS directly. For the sake of simplicity, we also do it too. In addition, we assume that all the nodes including BS are loosely time synchronized and the number of sensors in a group is relatively small so each sensor node can store the broadcast parameters of the other nodes which belong to the same group, more specifically the last hash value of the others' OHC. Finally, we assume that there is no

global broadcast authentication protocol. As mentioned before, if we allows the BS to directly broadcast messages into the entire network using a $\mu$TESLA-*like* protocol, it could be exploited by an adversary to destruct the network.

## 4.3 Overview of Our Protocol

Our protocol also consists of two phases with one additional key pre-distribution phase before deployment. At a set-up phase, we also elect a CH per a group and construct secure channels between the CH and its member nodes in an authenticated manner. After the set-up is complete, there is a much longer steady-state, the duration is pre-determined by the BS, where the member nodes sense data from their surroundings and then send the data to the CH with a MAC generated by its shared key, then the CH verifies these data, performs data aggregation, and sends it to the BS using its shared key. In addition, we add a localized (*in-cluster*) broadcast authentication mechanism by which CHs can broadcast any queries or commands to their member nodes in an authenticated manner.

Figure 4.1 and 4.2 show the overall time line and one round of our protocol.

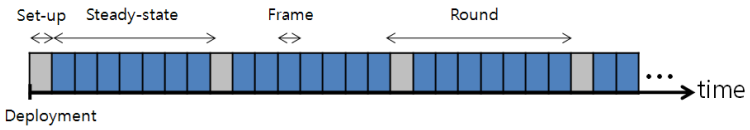


Figure 4.1: Time line of Our Protocol

Different from any variants of LEACH, our protocol requires the authentication of an advertisement at the end of every set-up phase which may consume a little more energy of sensor nodes than others.

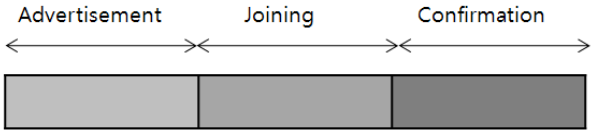


Figure 4.2: Cluster Set-up Phase

Table 4.1: Notation used in Our Clustering Protocol

| Notation | Description |
|---|---|
| $CH_i$ | the cluster head of i$th$ cluster |
| $g$ | the number of clusters (or group) in the deployment area |
| $G_i$ | i$th$ cluster (or group) |
| $H()$ | a hash function |
| $HK_j^{CH_i}$ | j$th$ hash value of $CH_i$'s OHC |
| $ID_{\eta_z^i}$ | ID of $\eta_z^i$ |
| $K_c$ | a key derived from a key pool with $c$ as its ID |
| $KP_i$ | a key pool used by all the sensor nodes in $G_i$ |
| $l$ | the total number of sensor nodes to be deployed |
| $MAC_x(msg)$ | a message authentication code of $msg$ with $x$ as a key |
| $MAX_B$ | the maximum broadcasting number in a round |
| $MAX_D$ | the maximum propagation delay to send a message from $CH_i$ to all nodes in a cluster |
| $nonce$ | a nonce |
| $\varpi$ | the number of sensor nodes in a cluster |
| $\eta_z^i$ | z$th$ sensor node in $G_i$ |

Table 4.1 describes the notation used in the protocol details.

## 4.4 Protocol in Details

### 4.4.1 Pre-distribution Phase

Prior to deploying $l$ sensor nodes in the field of interest, BS divides sensor nodes into $g$ groups $G_i$ where $1 \leq i \leq g$. Each group contains $\varpi$ ($\varpi = l/g$) sensor nodes. Then, the BS generates $g$ key pools

$(KP_0, ..., KP_g)$. Let $\eta_z^i$ where $1 \leq z \leq \varpi$ denote a sensor node in $G_i$. Each $\eta_z^i$ is given its unique id $ID_{\eta_z^i}$. The BS stores $w$ keys from $KP_i$ into $\eta_z^{i,j}$ according to the above mentioned ID-based random key pre-distribution scheme (see 3.2). In addition, the BS decides the maximum broadcasting number in a round $MAX_B$.

For each group $G_i$, the BS generates $\varpi$ random numbers $R_1$, ..., $R_\varpi$ and $\varpi$ OHCs by repeatedly hashing $R_j$, $HK_q^j = H(HK_{q+1}^j)$ for $j = 0$ to $\varpi$ and $q = MAX_B - 1$ to $-1$ where $HK_{MAX_B}^j = R_j$. Then the BS stores all $HK_{-1}^j$s into $\eta_z^{i,j}$ for $z = 1$ to $\varpi$ and $j = 1$ to $\varpi$. Finally, the BS stores $R_j$ and $MAX_B$ to $\eta_z^{i,z}$ for $z = 1$ to $\varpi$.

### 4.4.2 Deployment phase

The BS fixes the grid locations (there are exactly $g$ grid points) over the field of interest. The $i$th group members are deployed around the $i$th grid point randomly.

### 4.4.3 Cluster Set-up Phase

#### Advertisement

At the start of a round $r$, a node in $G_i$ is self-elected as $CH_i$ using a cluster head election algorithm as described in [3]. The $CH_i$ then broadcasts an `advertisement` message to the nodes in the group (every message contains a nonce to prevent replay attacks). Figure 4.3 shows an illustrative example of our advertisement where $\varpi = 5$ and $w$= 3. In this figure $\eta_3^i$ is self-elected as $CH_i$ and broadcast its `advertisement` message.

`[advertisement]` $CH_i \rightarrow G_i$:

$$ID_{CH_i}, nonce, MAC_{HK_0^{CH_i}}(ID_{CH_i}, nonce)$$

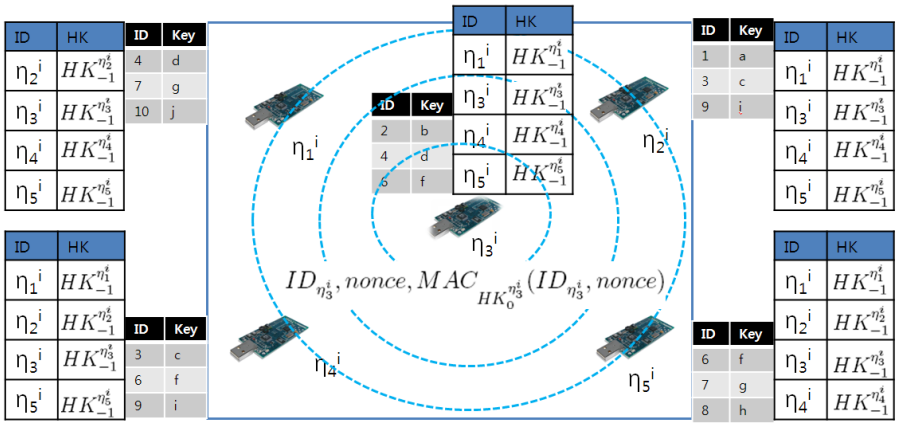


Figure 4.3: `Advertisement` Example

### Joining

On receiving the `advertisement` message from the $CH_i$, each node $\eta_k^i$ in $G_i$ who has a common key[1] with the $CH_i$, then responds by unicasting a `join-request` message which contains the ID (or index value) of the common key, $c$, to the $CH_i$. Those who do not share a key removes $HK_{-1}^{CH_i}$ from their memory and have themselves sleep for this round. These sensor nodes will take part in the next round with a higher probability of becoming a CH. Figure 4.5 shows an example of this joining step.

`[join-request]`$\eta_z^i \rightarrow CH_i$:

$$ID_{\eta_z^i}, ID_{CH_i}, c, nonce, MAC_{K_c}(ID_{\eta_z^i}, ID_{CH_i}, c, nonce)$$

### Confirmation

After receiving all the `join-requests`, the $CH_i$ determines the maximum broadcast delay, $MAX_D$, sets up a TDMA schedule and then

[1]By repeatedly generating an index value (upto $w$) using PRNG with $ID_{CH_i}$ as a seed value, each node can find a common key

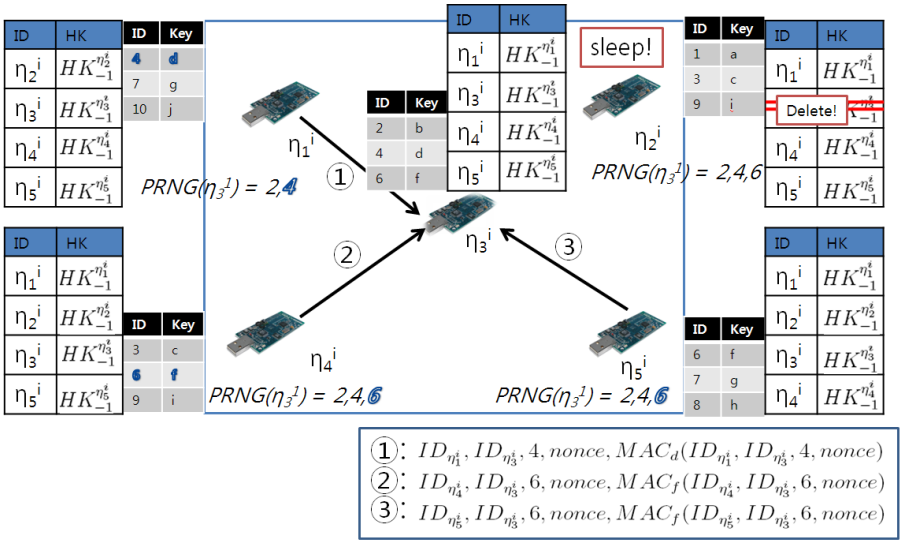


Figure 4.4: `Join-request` Example

broadcasts a `pre-confirmation` message to all the members in the group.

`[pre-confirmation]` $CH_i \rightarrow G_i$:

$$ID_{CH_i}, nonce, Schedule, MAC_{HK_0^{CH_i}}(ID_{CH_i}, nonce, Schedule)$$

After $MAX_D$ time elapsed, the $CH_i$ broadcasts a `confirmation` message to all the members.

`[confirmation]` $CH_i \rightarrow G_i$:

$$HK_0^{CH_i}$$

By verifying whether $H(HK_0^{CH_i})$ is equal to the stored $HK_{-1}^{CH_i}$ or not, the member nodes in $G_i$ can authenticate both the `advertisement` and `pre-confirmation` messages, and shut-off their transmitters at all times except the their scheduled time and the $CH_i$'s broadcasting time; thus, save their scarce energy.

Figure 4.6 shows a frame of our protocol. To the shake of simplicity, we assume $MAX_B$ is two times bigger than the number of frames in a

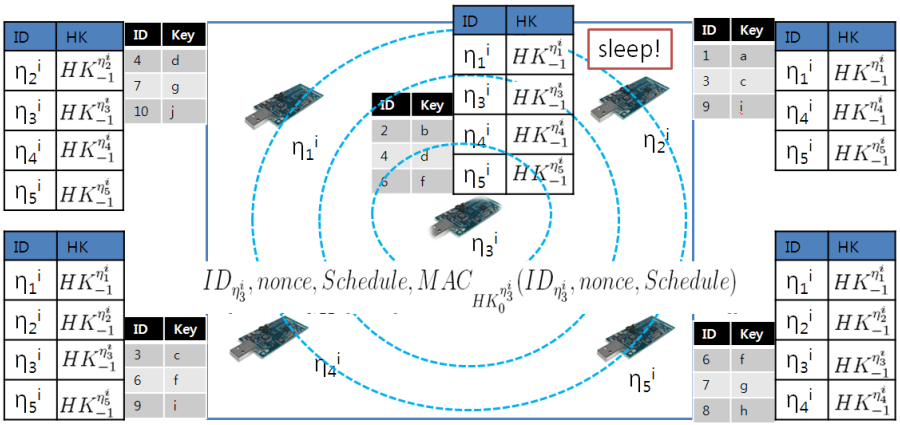


Figure 4.5: `Pre-confirmation` Example

round. In the figure, each slot can be assigned to any member nodes.

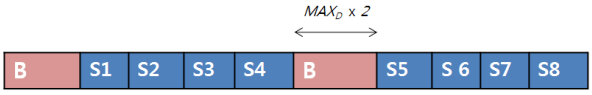


Figure 4.6: Frame Structure of Our Protocol

### 4.4.4 Steady-state Phase

The set-up stage is followed by a much longer steady-state, where the members sense data from the environment and send the data with a MAC generated by its key to the CH, which then verifies these data, performs data aggregation, and sends it to the BS using its shared key. In this phase, member nodes can securely transmit their data to their $CH_i$

`[Node-to-CH]` $\eta_z^i \rightarrow CH_i$:

$$ID_{\eta_z^i}, ID_{CH_i}, nonce, Data, MAC_{K_c}(ID_{\eta_z^i}, ID_{CH_i}, nonce, Data, )$$

Additionally, the $CH_i$ can broadcast at maximum $MAX_B - 1$ queries (or commands) to its member which is protected by $HK_z^{i,CH_i}$ where $0 < z < MAX_B$.

`[authenticated-broadcast]` $CH_i \rightarrow G_i$:

$$ID_{CH_i}, command, nonce, MAC_{HK_z^{CH_i}}(ID_{CH_i}, command, nonce)$$

After $MAX_D$ time elapsed, the $CH_i$ broadcasts a `broadcast-confirmation` message to its members.

`[broadcast-confirmation]` $CH_i \rightarrow G_i$:

$$HK_z^{CH_i}$$

On receiving the `authenticated-broadcast` message, $\eta_z^i$ first stores the message into its memory until $MAX_D$ time elapsed from its arrival time. If $\eta_z^i$ receives the `broadcast-confirmation` message on time, it first verifies $HK_z^{CH_i}$ by comparing $H^{x-z}(HK_z^{CH_i})$ with $HK_x^{CH_i}$ which is the latest authentic value received from the $CH_i$. If the verification succeeds, $\eta_z^i$ replaces $HK_x^{CH_i}$ with $HK_z^{CH_i}$ and then verifies the `[broadcast-confirmation]` message. If any of verifications is failed, $\eta_z^i$ just drops the messages.

The last broadcast slot is dedicated to re-initiate the OHC of the $CH_i$ for the next time which the $CH_i$ will become a CH again. The $CH_i$ generates a random number, $R$ and an OHC by repeatedly hashing $R$, $HK_q = H(HK_{q+1})$ for $q = MAX_B - 1$ to $-1$ where $HK_{MAX_B} = R$. Then the $CH_i$ stores $R$ into its memory and broadcasts a `reinitiation` message to its members.

`[reinitiation]` $CH_i \rightarrow G_i$:

$$ID_{CH_i}, nonce, HK_{-1}, MAC_{HK_{MAX_B}^{CH_i}}(ID_{CH_i}, nonce, HK_{-1})$$

After $MAX_D$ time elapsed, the $CH_i$ broadcasts a `confirm-reinitiation` message to its members (additionally, the $CH_i$ transmits this message to BS to support node addition which is protected by a shared key between the $CH_i$ and the BS).

`[confirm-reinitiation]` $CH_i \rightarrow G_i$:

$$HK_{MAX_B}^{CH_i}$$

If the verification of the message succeeds, member nodes store $HK_{-1}$ into its memory as $HK_{-1}^{CH_i}$ which will be used later round when the $CH_i$ becomes a CH again.

## 4.5 Analysis

### 4.5.1 Connectivity and Resiliency

Since our scheme utilizes the same key pre-distribution protocol of GS-LEACH, it provides the same connectivity and resiliency of GS-LEACH.

### 4.5.2 Local Broadcast Authentication

In our scheme, each CH has its own local OHC. Using it, a CH can broadcast any queries and commands to its members in an authenticated manner. If an adversary tries to launch DoS attack by flooding malicious messages in a specific cluster, it could not affect the other clusters as we only granted localized broadcasting.

### 4.5.3 Comparison

Table 4.2 shows the comparison of our protocol with GS-LEASH. We have not simulated our scheme yet, but we strongly believe that our scheme outperforms GS-LEACH as broadcasting a message to the members is more efficient than unicasting a message to all the members. Our scheme requires $\varpi$ hash values to be stored in each node but it occupies a small amount of the memory of sensor nodes, since the number of sensor nodes, $\varpi$, in a cluster is relatively small (usually 10 to 100). To

Table 4.2: Comparison with GS-LEACH

| Overhead | Ours | GS-LEACH |
|---|---|---|
| Communication (Bytes) | $(72 + Sch)BRO + 28\varpi P_{conn}UNI$ | $4BRO + \varpi P_{conn}(52 + Sch)UNI$ |
| Computation | $\varpi P_{conn}(4M + H + wP) + 2M$ | $\varpi 4P_{conn}M + wP$ |
| Memory | $w + \varpi$ | $w$ |

$Sch$: the size of $Schedule$, $BRO$: Broadcasting, $UNI$: Unicasting
$M$: MAC computation, $H$: Hash computation, $P$: PRNG computation

evaluate our protocol, we assume that broadcasting a byte, unicasting a byte, and receiving a byte consume 0.0167 $mWs$, 0.003 $mWs$, and 0.005 $mWs$ respectively. In addition, we use SHA1 which consumes 0.0059 $mWs$ per a byte and HMAC-SHA1 which consumes 0.0168 $mWs$ per a byte as a hash (or PRNG) function and a MAC function respectively[2]. Figure 4.7 and figure 4.8 show the comparison with GS-LEACH with respect to the total energy consumption in each round and the memory overhead in a sensor node when our protocol is only used to set-up clusters.

[2]We utilize the experimental result of the energy consumptions from [29].

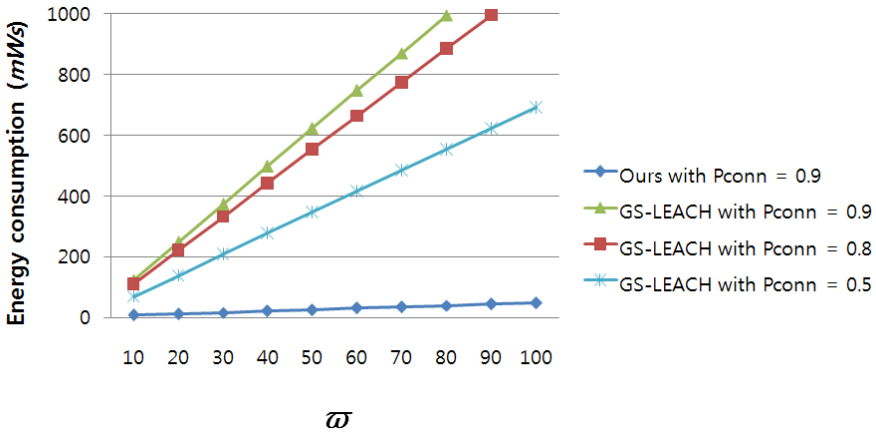


Figure 4.7: Comparison of Energy Overhead in Each Round

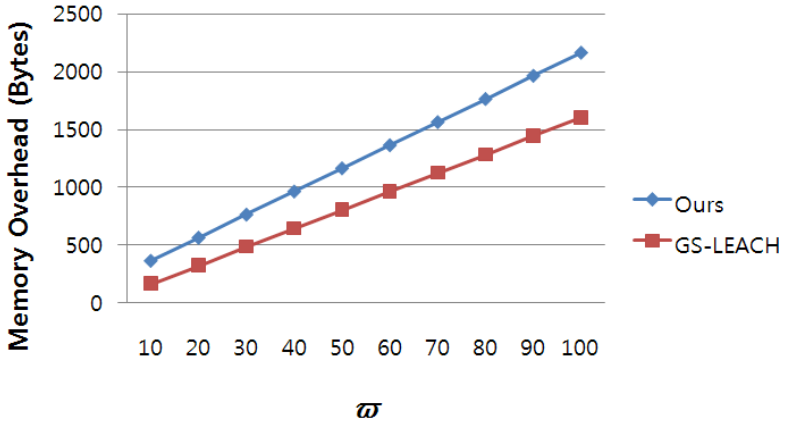


Figure 4.8: Comparison of Memory Overhead in a Sensor Node

# V. User Authentication with Privacy Protection

## 5.1 Motivation

In WSNs, user authentication schemes are classified into two categories: public key-based user authentication [4, 5, 33, 34] and secret key-based user authentication [2, 35]. Public key-based user authentication schemes assume that public key operation is feasible for even a tiny sensor node [16, 32]. All of the public key-based schemes utilize a certificate which is generated by BS. In general, however, public key operation is much more slower and consume much more energy than secret key operation. Thus, if an attacker launches *DOS* attack, the attacker can easily exhaust the limited energy of sensor nodes. Many secret key-based user authentication schemes adopt Blundo's key pre-distribution scheme [1]. Although these schemes are more efficient than the public key-based schemes, they also have some problems: once authenticated, always authenticated [2]; the trajectory of a user must be predetermined [35].

In addition, all the above mentioned schemes could violate user privacy, because they reveal the ID of a user [2, 4, 5, 34] and only consider the characteristics of sensor node and the computing power of user, but do not consider the other abilities of user. To acquire environmental information gathered by sensor nodes, a user will carry a mobile device, such as a mobile phone, a PDA, or a laptop computer. Before collecting the information from sensor nodes, this device should verify the user via password or biometric information. After authenticating the user, the device should proceed to an authentication process with its local sensor nodes so that the device can collect the environmental information.

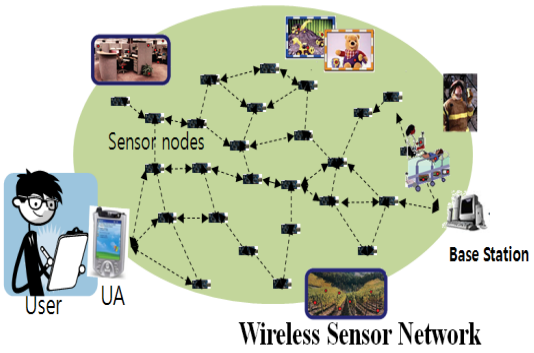


Figure 5.1: User Agent and Wireless Sensor Network

From now on, we name this device User Agent (UA). Fig. 5.1 shows the relationship between UA and a WSN.

In general, UA can directly communicate with BS through existing mobile communication system. For example, a mobile phone is able to access the Internet almost everywhere. In this case, the mobile phone can send its current position to BS and then the BS directly provides the environmental information to the mobile phone, but it has the following problems:

- Firstly, if the data produced by sensor nodes is intermittently collected by the BS, then this information provided by the BS may not be up-to-date.

- Secondly, in order to provide fresh data to user, the BS has to broadcast queries which are targeted to the user's local sensor nodes. This expends the energy of the other nodes which relay these queries.

To resolve these problems, user needs a means to directly communicate with sensor nodes and before beginning this communication, sensor nodes must authenticate the user and observe activities of the user to

protect themselves from malicious attacks. In addition, a user authentication scheme must conceal the information of the user during the authentication process. If the information is revealed in the authentication step, it could violate user privacy. An attacker, for example, can easily monitor the whereabouts of the users by eavesdropping authentication messages.

Moreover, most schemes assumed that WSN has a time synchronization mechanism by which they create a certificate [4, 5, 33, 34] or a pair-wise key [35]. Since time synchronization consistently consumes the limited energy of sensor nodes, a scheme is more efficient and reliable than others if the scheme does not depend on any time synchronization protocol.

In this chapter, we propose an efficient and secure user authentication scheme which protects the privacy of the user and utilizes the local time clock of sensor nodes and additional advantages of the user, *i.e.* the communication ability as well as the computation power of UA. The merits of our scheme are as follows:

1. It reduces the energy consumption of sensor nodes as it does not need any public key operation.
2. It does not require any time synchronization mechanism.
3. It protects the privacy of the user.

## 5.2 Assumption

### WSN

It is considered to be an *ad-hoc* network which consists of a large number of sensor nodes and few base stations. Each sensor node can be either automatically configured into a network or not, since some of sensor

nodes could be intermittently disconnected with the network due to their environmental condition. We do not consider time synchronization because it constantly consumes the limited energy of all the sensor nodes in the network.

- **Sensor node**: Each sensor node continuously collects environmental data such as temperature, humidity, seismicity, *etc* and provides the data only to legitimate user or other sensors. It means that sensor node must verify the source of the request. We assume that every sensor node has limited resources and limited energy source and has an internal clock. Some examples are Telosb and MicaZ [27]. Symmetric key operation is much more efficient and faster than any of public key operation and does not affect the life time of a sensor node.

- **Base Station** (BS): It is a device which collects the information provided by sensor node and is assumed to manages WSN. It is always trusted by all the sensor nodes and users. BS must be secured against any type of attacks. It helps user and sensor node to authenticate each other by generating a ticket which includes a pair-wise key between a user and a sensor node, its expiration time, an access control list of the user, *etc*.

### User

A user is a person who wants to utilize the information of his or her local sensor nodes in order to make his or her everyday life much more comfortable than before. The user has a mobile device which is able to communicate with WSN. We call this mobile device as user agent. Before the user agent proceeds to a user authentication process with its local sensor nodes, it must authenticate the user via password or biometric information.

**User Agent (UA)**

It is a mobile device, such as a mobile phone or a PDA with a radio module able to communicate with sensor nodes. It can communicate with sensor nodes only after authenticating its owner, first. We assume that mobile phone can communicate with BS directly through its mobile network and PDA can do it via its *WLAN*. This assumption is acceptable because these kinds of networks are now widely used in the world. Therefore, We assume that a secure *out-of-band* channel is established between UA and BS before starting user authentication processes between the UA and its local sensor nodes.

## 5.3 Protocol in Details

We adopt Kerberos [17] which provides both entity authentication and key establishment using secret key-based encryption techniques with a trusted third party and remodel it to be suitable for our assumption since Kerberos reveals the ID of user and heavily depends on time synchronization. Table 5.1 describes the notation used in the details of our proposed user authentication protocol.

**Step 1.** The UA generates a random number, $R_{UA}$, and hashes its ID concatenated with the random number. Then, the UA broadcasts the hashed value.

$$UA \rightarrow N_i : H(ID_{UA}, R_{UA})$$

**Step 2.** On receiving the above message, each sensor node encrypts its local time and the received value using $K_{BS,N_i}$ and then sends the encrypted value with its $ID_{N_i}$ to the UA.

$$N_i \rightarrow UA : ID_{N_i}, E_{K_{BS,N_i}}(T_{N_i}, H(ID_{UA}, R_{UA}))$$

Table 5.1: Notation used in Our User Authentication Protocol

| Notation | Description |
|---|---|
| $A_x$ | Communication range of $x$ |
| $AL_x$ | Access control list of $x$ |
| $E_{K_{x,y}}(*)$ | Secret key-based encryption with a key, $K_{x,y}$ |
| $H(msg)$ | Hash value of $msg$ |
| $ID_x$ | Identity of $x$ |
| $K_{x,y}$ | Shared secret key between $x$ and $y$ |
| $MAC_x(msg)$ | Message authentication code of $msg$ using $x$ as a key |
| $N_x$ | $x^{th}$ sensor node in WSN |
| $R$ | Random number |
| $T_{Exp}$ | Expiration time |
| $T_x$ | Current time of $x$ (local time stamp) |
| $TK_{x,y}$ | Ticket for $x$ and $y$ generated by BS |
| $\Rightarrow$ | Secure channel |
| $\rightarrow$ | Insecure channel |

**Step 3.** The UA sends $R_{UA}$ and the received message to the BS through the secure channel.

$$UA \Rightarrow BS : R_{UA}, ID_{N_i}, E_{K_{BS,N_i}}(T_{N_i}, H(ID_{UA}, R_{UA}))$$

**Step 4.** The BS decrypts $E_{K_{BS,N_i}}(T_{N_i}, H(ID_{UA}, R_{UA}))$ and hashes $ID_{UA}$, which the BS have already known in our assumption, concatenated with $R_{UA}$ and then compares it with the decrypted message. If the values are equal, the BS generates a ticket based on $T_{N_i}$, the right of the user, and then sends the ticket to the UA.

$$BS \Rightarrow UA : ID_{N_i}, T_{N_i}, T_{Exp}, AL_{UA}, K_{UA,N_i}, TK_{UA,N_i}$$
$$TK_{UA,N_i} = E_{K_{BS,N_i}}(H(ID_{UA}, R_{UA}), T_{N_i}, T_{Exp}, AL_{UA}, K_{UA,N_i})$$

**Step 5.** The UA generates a random number, $R'_{UA}$, and encrypts it using $K_{UA,N_i}$. Then, the UA sends it with the ticket to the sensor node, $N_i$.

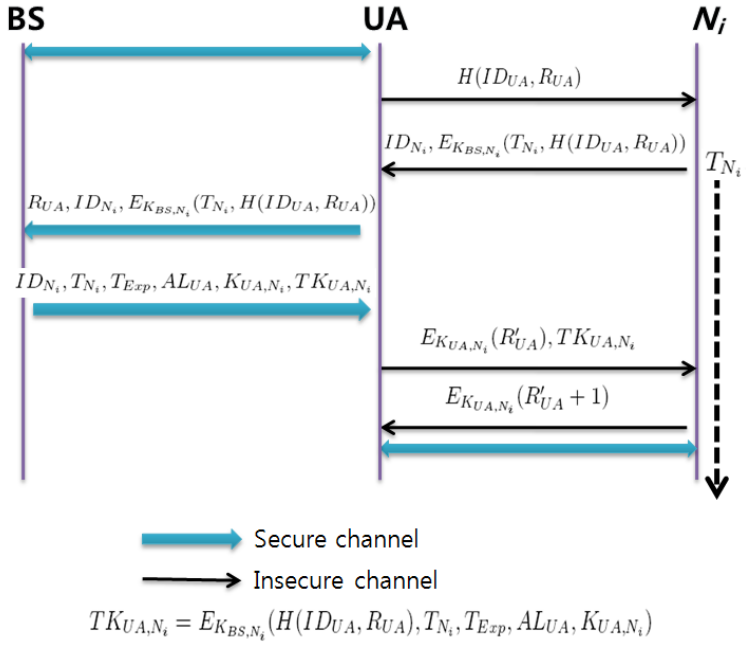


Figure 5.2: Our User Authentication Protocol

$$UA \rightarrow N_i : E_{K_{UA,N_i}}(R'_{UA}), TK_{UA,N_i}$$

**Step 6.** The $N_i$ authenticates the UA after verifying the received ticket and decrypts $E_{K_{UA,N_i}}(R'_{UA})$ using $K_{UA,N_i}$. After that, the $N_i$ encrypts $R'_{UA}+1$ using $K_{UA,N_i}$ and sends it to the UA.

$$N_i \rightarrow UA : E_{K_{UA,N_i}}(R'_{UA}+1)$$

On receiving above message, the UA verifies whether the $N_i$ knows the shared secret, $K_{UA,N_i}$, or not. If the verification is successfully finished, the UA can request information from $N_i$ in the period between $T_n$ and $T_n + T_{Exp}$ using the key, $K_{UA,N_i}$. Fig. 5.2 shows the overall protocol.

Table 5.2: Comparison with other User Authentication Protocols

| Scheme | Benenson *et al.*[5] | Wang and Li [33] | Zhang *et al.* [35] | Our scheme |
|---|---|---|---|---|
| Sensor node | $2PK$ | $2H+1SK+3PK$ | $1H$ | $4SK$ |
| Privacy protection | No | No | No | Yes |
| User trajectory | Random | Random | Predetermined | Random |
| Time synchronization | Needed | Needed | Needed | Not Needed |
| Connection with BS | Not Needed | Not Needed | Not Needed | Needed |

$PK$: Public key computation, $SK$: Secret key computation, $H$: Hash computation

## 5.4 Security Analysis

Our protocol is similar to the Kerberos protocol. For its proper operation, however, Kerberos heavily depends on the *network-wide* time synchronization which is acceptable in the typical distributed computing environment, but not in WSNs as it consists of a large number of resource-constrained sensor nodes. Our scheme does not depend on the time synchronization because it uses the local time stamp of sensor node to which the UA wants to authenticate itself. Table 5.2 shows the comparison with other schemes.

### 5.4.1 Mutual Authentication

The BS is a third party trusted by both *UA* and sensor node. For user authentication, it issues a ticket according to the ID of the user, $ID_{UA}$, and the sensor node, $N_i$. Since only legitimate user can request a ticket and legitimate sensor node can share a secret with the BS, both the user and the sensor node authenticate each other according to the ticket in the authentication Step 5 and Step 6. Even if an attacker compromise a few number of sensor nodes, it does not damage any

others authentication processes.

### 5.4.2 Privacy Protection

Most previous works do not consider the privacy of the user, but it must be deliberated. Since all messages are broadcasted on the air in WSNs, an attacker can easily eavesdrop the messages. It can violate user privacy such as monitoring the whereabouts of the user. To protect the user privacy in our scheme, the ID of user is always hashed with a random number, $H(ID, R)$, for hiding the ID before broadcasting. Receiving the broadcasted hashed value, the sensor node, $N_i$, starts to verify the user. Even after finishing the user authentication, the sensor node does not know who the user is because it identifies the user with the hash value, *i.e.*, none of the sensor nodes know the real ID of the user. This prevents user privacy violation.

### 5.4.3 Efficiency

Our proposed scheme only uses four secret key operations in a sensor node. Since secret key operation is generally much faster and more efficient than any public key operation, it reduces the energy consumption of sensor nodes. Although Zhang *et al.*'s scheme only require one hash operation per a sensor node, as mentioned before this restricts the trajectory of users.

### 5.4.4 Access Control

A sensor node processes a request of a legitimate user only if the request is allowed to the user based on the access control list of the user, $AL_{UA}$. It protects the sensor node from careless queries of the legitimate user and conserves the energy of the sensor node.

### 5.4.5 No Time Synchronization

Kerberos and other authentication schemes that we mentioned before heavily depend on time synchronization. In WSNs, the time synchronization continuously consumes the limited energy of all the sensor nodes. Even more, if an attacker destroys some parts of WSN, the time synchronization will be not provided for a while. In this case, Kerberos and other schemes are not operated properly, but our scheme is not affected. Our scheme does not need any time synchronization protocol at all because it creates tickets based on the local time stamp, $T_{N_i}$, of the sensor node, $N_i$. Thus, our scheme can conserve the limited energy of sensor node and continue to operate even when the network configuration is disintegrated.

## 5.5 Enabling Localized Multi-user Broadcast Authentication

In this section, we briefly describe how to combine our clustering protocol and user authentication protocol to enable localized multi-user broadcast authentication (MBA). Since we have constructed a secure clustering protocol with *in-cluster* broadcast authentication, we can utilize it as the underlying network protocol of our user authentication protocol. Figure 5.3 describes the sequence of our localized MBA protocol. The difference is that now a user has to authenticate himself to a cluster head as well as a normal node. Then, the user sends a broadcasting request to the cluster head and the head broadcasts the request on behalf of the user. We assume that the communication range of the user is limited, so the user could only broadcast his/her queries and commands in his/her surroundings to which the signal of the user' device is reached ($A_U$ denotes the communication range of the user).

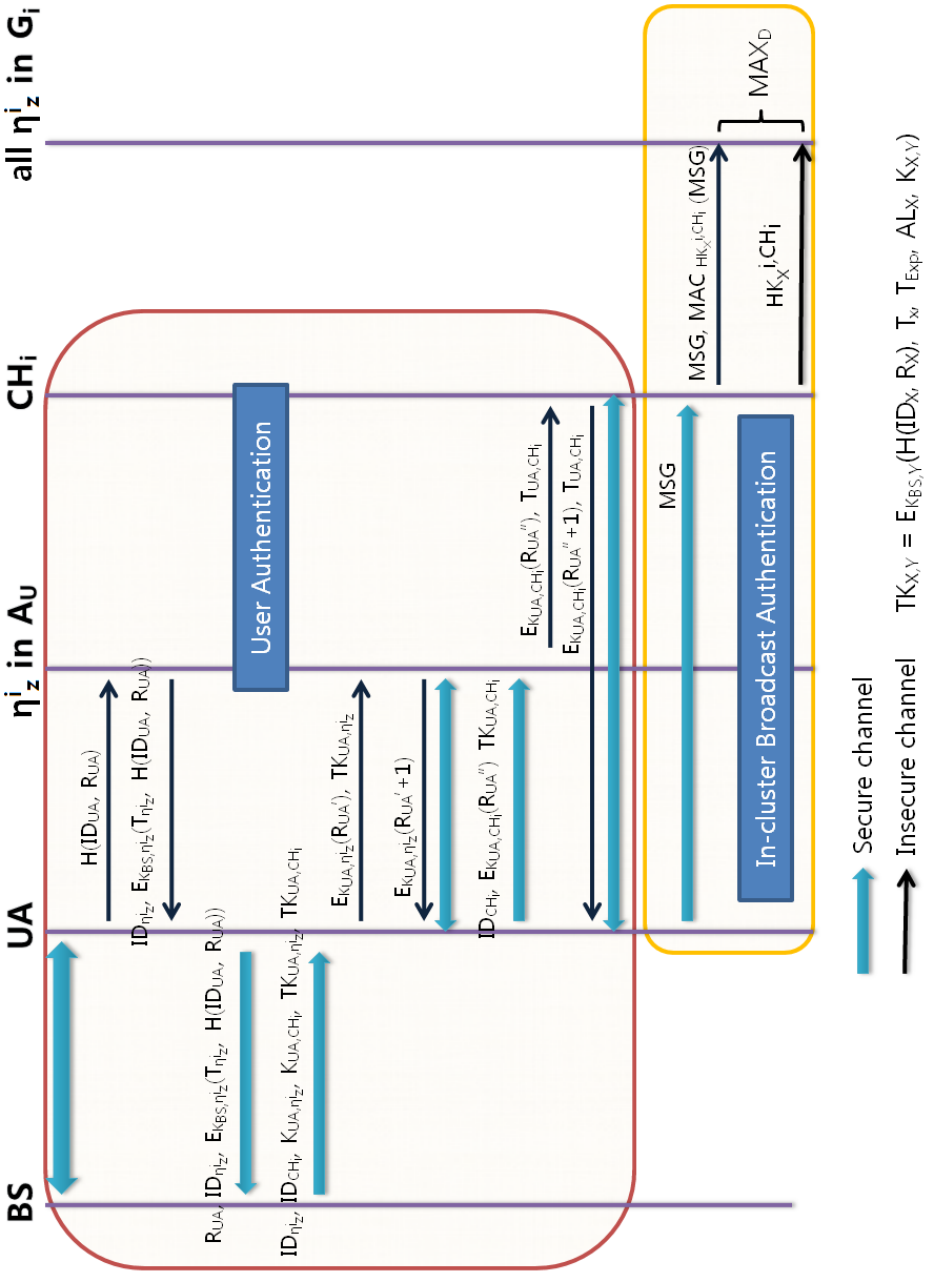


Figure 5.3: Localized Multi-user Broadcast Authentication

# VI. Summary and Conclusion

In this work, we have provided a modified clustering protocol to enable localized broadcast authentication in a cluster and a user authentication protocol with privacy protection exploiting existing mobile networks. Our proposed secure clustering protocol provides secure Node-to-$CH$ communication and localized broadcast authentication in a cluster more efficiently comparing with GS-LEACH and our user authentication protocol provides an efficient secure Node-to-User communication with the anonymity of users.

By combining these two protocols, we can provide an efficient localized multi-user broadcast authentication mechanism.[1] We strongly believe that localized broadcasting is enough to the users since their excessive and careless message dissemination into the entire network could easily destruct the operation of the networks.

We have a lot of remaining work to be done. At first, we have to carefully simulate our clustering protocol. We should consider an efficient node addition mechanism, and design and efficient node compromise detection/recovery mechanism.

[1]Our clustering protocol can be combined with any other user authentication protocols to bootstrap in-cluster broadcasting of users' messages.

# 무선 센서 네트워크에서의 지역적인 다중 사용자 방송형 인증 기법에 관한 연구

윤성준

무선 센서 네트워크 (Wireless Sensor Network, WSN)는 유비쿼터스 시대를 앞당길 가장 핵심적인 기술 중 하나로 많은 연구가 진행되고 있다. WSN은 수백에서 수천 개의 자원 제약적인 소형의 센서 노드들과 소수의 안전하고 자원이 풍부한 기지국으로 구성된 무선 *Ad-hoc* 네트워크로써 각각의 센서 노드들은 자신의 주변 환경 데이터를 획득하고 정제하여 이를 기지국에 제공한다. 기지국은 센서 노드들로부터 획득한 데이터를 보다 유용한 정보로 가공하여 이를 필요로 하는 사용자들에게 제공한다. 기존의 유/무선 네트워크 (인터넷, 모바일 *Ad-hoc* 네트워크)와는 다르게, WSN은 자동화된 원격 데이터 획득을 위해, 전장 감시에서 빌딩 관리 응용에 이르기까지 다양한 분야에서 연구가 수행되어오고 있다 [7].

WSN은 기존의 *Ad-hoc* 네트워크와 많은 유사점을 가지고 있지만 센서 노드들이 Ad-hoc 네트워크에서 고려되는 무선 단말에 비해 보다 자원 제약적이며, 네트워크의 규모가 훨씬 크다는 차이점을 가지고 있다. 따라서 무선 Ad-hoc 네트워크를 위해 고안된 보안 기법들을 WSN에 직접 적용되기에는 많은 문제점이 존재한다.

WSN에서 방송형 인증은 네트워크 전체적으로 확산되는 소프트웨어의 갱신, 질의 등의 메시지를 센서 노드가 효율적으로 인증하기 위한 기법이다. 초기에 제안된 방송형 인증 기법들은 주로 비밀키 암호화 시스템 (Secret Key Cryptosystem, SKC)에 기반을 두어 설계되었으며, 소수의 송신자 (주로 기지국)만을 지원한다. 최근에는 다수의 송신자 (모바일 유저)를 지원하기 위해 공개키 암호화 시스템 (Public

Key Cryptosystem, PKC)을 이용한 기법들이 제안되고 있다.

본 논문에서 다루는 주제는 무선 센서 네트워크를 위한 안전한 클러스터링 기법과 사용자 인증 기법이다. 제안하는 클러스터링 기법은 클러스터 내부적으로 방송되는 메시지를 센서 노드가 효율적으로 인증할 수 있는 방법을 제시하며, 사용자 인증 기법은 올바른 네트워크 사용자와 센서 노드 사이에 안전한 통신 채널을 효율적으로 구축하는 방법을 제시한다.

제안한 두 기법을 조합하므로써, 사용자가 자신의 인접 클러스터에 메시지를 방송하고, 이를 센서 노드가 효율적으로 인증하는 방법을 제시한다. 이러한 조합은 비밀키 암호화 시스템 (해쉬, 메시지 인증 코드 포함)만을 사용하기 때문에 기존의 공개키 암호화 시스템을 이용해 설계된 다중 사용자 방송형 인증 기법에 비해 효율적이다.

# References


1. C. Blundo, A. D. Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, "Perfectly-Secure Key Distribution for Dynamic Conferences," *Advances in Cryptology, Proceedings of CRYPTO'92*, LNCS 740, pp. 471-486, 1993.

2. S. Banerjee and D. Mukhopadhyay, "Symmetric Key Based Authenticated Querying in Wireless Sensor Networks," in *Proceedings of the 5st ACM International Conference on Integrated Internet Ad hoc and Sensor Networks*, May 2006.

3. P. Banerjee, D. Jacobson, and S. N. Lahiri, "Security and Performance Analysis of a Secure Clustering Protocol for Sensor Networks," in *Proceedings of the 6th IEEE International Symposium on Network Computing and Applications*, July 2007.

4. Z. Benenson, "Authenticated Queries in Sensor Networks," in *Proceedings of the 2nd European Workshop on Security and Privacy in Ad hoc and Sensor Networks*, July 2005.

5. Z. Benenson, N. Gedicke, and O. Raivio, "Realizing Robust User Authentication in Sensor Networks," in *Proceedings of the 1st Workshop on Real-World Wireless Sensor Networks*, June 2005.

6. H. Chan and A. Perrig, "PIKE: Peer Intermediaries for Key Establishment in Sensor Networks," in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, Mar. 2005.

7. C. Chong and S. Kumar, "Sensor networks: evolution, opportunities, and challenges," *Proceedings of the IEEE*, Vol.91 No.8 pp. 1247-1256, 2003.

8. J. Deng, R. Han, S. Mishra, "Security Support for In-Network Processing in Wireless Sensor Networks," in *Proceedings of the 1st ACM Workshop Security of Ad Hoc and Sensor Networks"*, Oct. 2003.

9. T. Dimitriou and I. Krontiris, "Autonomic Communication Security in Sensor Networks," in *Proceedings of the 2nd International Workshop on Autonomic Communication*, Oct. 2005.

10. J. Drissi and Q. Gu, "Localized Broadcast Authentication in Large Sensor Networks," in *Proceedings of International Conference on Networking and Services*, July, 2006.

11. W. Du, J. Deng, Y. S. Han, and P. K. Varshney, "A Pairwise Key Pre-distribution Scheme for Wireless Sensor Networks," in *Proceedings of the 10th ACM Conference on Computer and Communications Security*, Oct. 2003.

12. W. Du, J. Deng, Y. S. Han, S. Chen, and P. K. Varshney, "A Key Management Scheme for Wireless Sensor Networks Using Deployment Knowledge," in *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies*, Mar. 2004.

13. L. Eschenauer and V. D. Gligor, "A Key Management Scheme for Distributed Sensor Networks," in *Proceedings of the 9th ACM conference on Computer and Communication Security*, Nov. 2002.

14. A. C. Ferreira, M. A. Vilaca, L. B. Oliveira, E. Habib, H. C. Wong, and A. A. F. Loureiro, "On the Security of Cluster-based Commu-

nication Protocols for Wireless Sensor Networks," in *Proceedings of the 4th IEEE International Conference on Networking*, Apr. 2005.

15. A. Manjeshwar and D. P. Agrawal, "TEEN: A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks," in *Proceedings of the 15th International Parallel and Distributed Processing Symposium Workshops*, Apr. 2001.

16. V. Gupta, M. Millard, S. Fung, Y. Zhu, N. Gura, H. Eberle, and S. C. Shantz, "Sizzle: A Standards-based End-to-End Security Architecture for the Embedded Internet," in *Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communication*, Mar. 2005.

17. J. T. Kohl and B. C. Neuman, "The Kerberos Network Authentication Service (Version 5)," RFC 1510, Sep. 1993.

18. P. E. Lanigan, R. Gandhi, P. Narasimhan, "Sluice: Secure Dissemination of Code Updates in Sensor Networks," in *Proceedings of the 26th International Conference on Distributed Computing Systems*, July, 2006.

19. L. Lamport, "Password Authentication with Insecure Communication," *Communications of the ACM* , Vol.24, No.11, pp. 770-772 1981.

20. S. Lindsey and C. S. Raghavendra. "PEGASIS: Power-Efficient Gathering in Sensor Information Systems," in Aerospace Conference Proceedings, 2002. IEEE, 2002.

21. D. Liu and P. Ning, "Multi-level $\mu$TESLA: Broadcast Authentication for Distributed Sensor Networks," *ACM Transactions in Embedded Computing Systems*, Vol.3, No.4, pp. 800-836, Feb. 2004.

22. D. Liu, P. Ning, S. Zhu, and S. Jajodia, "Practical Broadcast Authentication in Sensor Networks," in *Proceedings of the 2nd Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services*, Nov. 2005.

23. R. Merkle, "Protocols for Public Key Cryptosystems," in *Proceedings of the IEEE Symposium on Research in Security and Privacy*, Apr. 1980.

24. M. Mitzenmacher, "Compressed Bloom Filters," *IEEE/ACM Transactions on Networks*, Vol.10, No.5, pp. 613-620, Oct. 2002.

25. W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," *IEEE Transactions on Wireless Communications*, Vol.1 No.4 pp. 660-670, Oct. 2002.

26. L. B. Oliveira, H. C. Wong, M. Bern, R. Dahab, and A. A. F. Loureiro, "SecLEACH-A Random Key Distribution Solution for Securing Clustered Sensor Networks," in *Proceedings of the 5th IEEE International Symposium on Network Computing and Applications*, July 2006.

27. J. Polastre, R. Szewczyk, and D. Culler, "Telos: Enabling Ultra-Low Power Wireless Research," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks*, Apr. 2005.

28. A. Perrig, R. Szewczyk, V. Wen, D. Culler, and D. Tygar, "SPINS: Security Protocols for Sensor Networks," in *Proceedings of 7th Annual International Conference on Mobile Computing and Networks*, July 2001.

29. K. Piotrowski, P. Langendoerfer, and S. Peter, "How Public Key Cryptography Influences Wireless Sensor Node Lifetime," in *Proceedings of the 4th ACM Workshop on Security of Ad hoc and Sensor Networks*, Oct. 2006.

30. K. Ren, K. Zeng, W. Lou, and P. Moran, "On Broadcast Authentication in Wireless Sensor Networks," in *Proceedings of the First Annual International Conference on Wireless Algorithms, Systems, and Applications*, Aug. 2006.

31. K. Ren, W. Lou, and Y. Zhang, "Multi-user Broadcast Authentication in Wireless Sensor Networks," in *Proceedings of the 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, June 2007.

32. S. Wander, N. Gura, H. Eberle, V. Gupta, and S. C. Shantz, "Energy Analysis of Public-Key Cryptography for Wireless Sensor Networks," in *Proceedings of the 3rd International Conference on Pervasive Computing and Communication*, Mar. 2005.

33. H. Wang and Q. Li, "Distributed User Access Control in Sensor Networks," in *Proceedings of the 2nd IEEE International Conference on Distributed Computing in Sensor Systems*, June 2006.

34. K. H. M. Wong, Y. Zheng, J. Cao, and S. Wang, "Dynamic User Authentication Scheme for Wireless Sensor Networks," in *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, June 2006.

35. W. Zhang, H. Song, S. Zhu, and G. Cao, "Least Privilege and Privilege Deprivation: Towards Tolerating Mobile Sink Compromises in Wireless Sensor Networks," in *Proceedings of the 6th ACM Interna-*

*tional Symposium on Mobile Ad Hoc Networking and Computing*, May 2005.

36. Y. Zhou and Y. Fang, "BABRA: Batch-based Broadcast Authentication in Wireless Sensor Networks," in *Proceedings of the 49th Annual IEEE Global Telecommunications Conference*, Nov. 2006.

37. S. Zhu, S. Xu, S. Setia, and S. Jajodia, "Establishing Pairwise Keys For Secure Communication in Ad Hoc Networks: A Probabilistic Approach," in *Proceedings of the 11th IEEE International Conference on Network Protocol*, Nov. 2003.

# Acknowledgement

I would like to express my sincere gratitude to Prof. Kwangjo Kim, my academic advisor, for his constant direction and support. Without his guidance, I could never have carried out my research in ICU. Special thanks are also due to Prof. Young-Hee Lee and Prof. Byungcheon Lee for their generosity and agreeing to serve as advisory committee members. I would also like to thank all members of our Cryptology and Information Security Laboratory: Hyunrok Lee, Zeen Kim, Kyusuk Han, Vo Duc Liem, Konidala Munirathnam Divyan, Dang Nguyen Duc, SungChul Heo, Sungbae Ji, Jangsung Kim, Minhea Kwak, Hanyoung Noh, Hyewon Park, and Hyeran Mun for giving me a lot of interests and good advices during the course of my study. I also thank Hyunkyoung Park for helpful support as a staff member.

# Curriculum Vitae

Name : Sungjune Yoon

Date of Birth : April. 10. 1979

Sex : Male

Nationality : Korean

## Education

| | |
|---|---|
| 1999.03–2006.02 | Computer and Multimedia Engineering<br>Pukyong National University (B.S.) |
| 2006.03–2008.02 | Cryptography and Information Security, Engineering<br>Information and Communications University (M.S.) |

## Career

| | |
|---|---|
| 2006.03–2006.12 | Graduate Research Assistant<br>Research on Link Layer Security Electronics and Telecommunications Research Institute (ETRI) |
| 2006.03–2006.12 | Graduate Research Assistant<br>Research on Security Standardization in RFID/USN<br>Electronics and Telecommunications Research Institute (ETRI) |

| | |
|---|---|
| 2006.03–2007.12 | Graduate Research Assistant<br>Development of Sensor Tag and Sensor Node Technologies for RFID/USN<br>Ministry of Information and Communication (MIC)<br>Institute for Information Technology Advancement (IITA) |
| 2006.08–2006.12 | Graduate Research Assistant<br>Research on Security Technology in Ubiquitous-Web Platform<br>KT Future Technology Lab |
| 2006.07–2007.06 | Graduate Research Assistant<br>A Study on the ID-Based Encryption (IBE) and Its Application in 4G. Network<br>Samsung-ICU Research Center |
| 2006.12–2007.12 | Graduate Research Assistant<br>Research and Development of Next Generation DRM<br>SK Telecom |
| 2007.02–2007.10 | Graduate Research Assistant<br>Research on the Anti-cloning Methods of Sensor Node for Wireless Sensor Networks<br>National Security Research Institute (NSRI) |
| 2007.03–2007.12 | Graduate Research Assistant<br>Development of Cyber Security Policy Standard for Digital I&C System<br>Korea Institute of Nuclear Safety (KINS) |
| 2007.06–2007.08 | Apprentice Researcher<br>Information Technologies Laboratories, Sony, Japan |

2007.03–2007.12 Teaching Assistant
Institute for IT-gifted Youth

## Publications

(1) 2006.09 한규석, 서영준, 윤성준, 김광조, "Enhancing Security for Vertical Handoff in SARAH under the Heterogeneous Networks," 2006년도 정보보호학술발표회논문집, pp. 159-166, 2006.09.29-30, 목원대학교, 대전.

(2) 2007.02 윤성준, 이현록, 김광조, "센서 노드에서의 효율적인 NTRUEncrypt," 2007년도 정보보호학회 영남지부 학술발표회논문집, pp. 26-31, 2007.02.23, 대구한의대 바이오센터, 대구.

(3) 2007.08 Sungjune Yoon, Hyunrok Lee, Sungbae Ji and Kwangjo Kim, "A User Authentication Scheme with Privacy Protection for Wireless Sensor Networks," *The 2nd Joint Workshop on Information Security*, pp.233-244, Aug. 6-7, 2007, Tokyo, Japan.

(4) 2007.10 윤성준, 이현록, 김광조, "무선 센서 네트워크에 적용되는 방송형 인증 기법의 조사 분석," 2007년도 정보보호학술발표회논문집, pp.211-219, 2007. 10.12, 한국기술교육대학교, 천안.

(5) 2007.10 지성배, 이현록, 윤성준, 김광조, "전자태그 시스템을 위한 인증 프레임워크의 요구 사항," 2007년도

정보보호학술발표회논문집, pp.63-70, 2007. 10.12, 한국기술교육대학교, 천안.

(6) 2007.12 윤성준, 이현록, 김광조, "이종망을 고려한 무선 센서 네트워크에서의 다중 사용자 방송형 인증 기법," 2007년도 한국정보보호학회 동계학술대회, 2007.12.01, 상명대학교, 서울. Best Paper Award.

(7) 2008.01 Sungjune Yoon, Hyunrok Lee, and Kwangjo Kim, "Hybrid Multi-user Broadcast Authentication for Wireless Sensor Networks," *Symposium on Cryptography and Information Security -SCIS'08*, To be appeared, Miyajaki, Japan.