석 사 학 위 논 문

Master's Thesis

# 무선 센서 및 액터 네트워크에서 위치 기반의 인증된 키합의 기법에 관한 연구

A Study on Location-Based Authenticated Key Agreement Scheme in Wireless Sensor and Actor Networks


최 임 성 (崔 林 成 Choi, Imsung)

정보통신공학과

Department of Information and Communications Engineering

한 국 과 학 기 술 원

Korea Advanced Institute of Science and Technology



2010

# 무선 센서 및 액터 네트워크에서 위치 기반의 인증된 키합의 기법에 관한 연구

# A Study on Location-Based Authenticated Key Agreement Scheme in Wireless Sensor and Actor Networks

# A Study on Location-Based Authenticated Key Agreement Scheme in Wireless Sensor and Actor Networks

Advisor : Professor Kim, Kwang Jo

by

Choi, Imsung
Department of Information and Communications Engineering
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 Engineering in the Department of Information and Communications Engineering

Daejeon, Korea
2009. 12. 18.
Approved by

---

Professor Kim, Kwang Jo
Advisor

# 무선 센서 및 액터 네트워크에서 위치 기반의 인증된 키합의 기법에 관한 연구

최 임 성

위 논문은 한국과학기술원 석사학위논문으로 학위논문심사 위원회에서 심사 통과하였음.

2009년 12월 18일

심사위원장 김 광 조 (인)

심사위원 김 명 철 (인)

심사위원 이 병 천 (인)

MICE 20084264 최 임 성. Choi, Imsung. A Study on Location-Based Authenticated Key Agreement Scheme in Wireless Sensor and Actor Networks. 무선 센서 및 액터 네트워크에서 위치 기반의 인증된 키합의 기법에 관한 연구. Department of Information and Communications Engineering . 2010. 35p. Advisor Prof. Kim, Kwang Jo. Text in English.

## Abstract

Wireless Sensor and Actor Network (WSAN) refers to a group of sensors and actors linked by wireless medium to perform distributed sensing and acting tasks. In WSAN, sensors gather useful information about the physical world, while actors take decisions and then perform appropriate actions upon the environment. The coexistence of sensors and actors introduces new challenges to realize WSAN. For securing WSAN, the new challenge is to provide the security mechanisms between actors and sensors because this problem is not addressed by the previous work for Wireless Sensor Network (WSN).

In this thesis, we focus on providing security mechanisms between actors and sensors. Specifically, we propose location-based Authenticated Key Agreement (AKA) scheme, operated over ID-based cryptosystem. The proposed scheme provides authentication and key establishment between actors and sensors and further offer the countermeasure against Denial of Service (DoS) attack of routing layer by utilizing the location information of sensors to generate location-based public and private key pair of sensors. We also analyze the security and performance of the proposed scheme. The proposed scheme provides the stronger security than other schemes for WASN in terms of resilience to node compromise and DoS attack. Although we try to assign smaller overhead to sensors to make sensors compute no pairing operation, the consumption of the energy of sensors is about twice than that of Yu *et al.*'s scheme.

# Contents

# List of Tables

# List of Figures

# 1. Introduction

## 1.1 Overview

Recently, Wireless Sensor and Actor Network (WSAN), which is the integrated network of actors and sensors, has been appeared. WSAN is capable of observing the physical world, processing the data, making decision based on the observations and performing appropriate actions [1]. In WSAN, the phenomena of sensing is performed by sensors, and the phenomena of acting is performed by actors.

Due to the coexistence of sensors and actors, many schemes, which have been proposed for Wireless Sensor Network (WSN), may not be well-suited for WSAN because WSN considers only sensors. For the security schemes for WSAN, security mechanisms of sensor-sensor are covered by the security schemes of WSN [13, 8, 12, 18, 19]. However, security mechanisms of actor-sensor are covered by none of the security schemes for WSN. Even though there has been some research effort related to secure mobile sink [30, 40] or access control schemes [33, 41, 20] for WSN, none of the schemes to investigate research challenges of security mechanisms for actor-sensor.

So far, some researches [7, 37] have been proposed to provide security mechanisms for actor-sensor. These studies focus on providing Authenticated Key Agreement (AKA) between actors and sensors because other security services can be covered by the previous schemes for WSN. These schemes tried to provide security mechanisms between actors and sensors, but the schemes have some vulnerabilities. Cao *et al.*'s scheme [7] utilized a symmetric key which is shared by all nodes for AKA between actor and sensors. In this scheme, the actors cannot perform AKA scheme dynamically because all nodes remove the key after first AKA, even though actors generally require to communicate dynamically with sensors [31, 34]. Yu *et al.*'s scheme [37] utilized Merkle hash tree instead of certificates. In the scheme, sensors authenticate themselves to actors by using the hash tree. Because the scheme is not fully based on public key cryptosystem, the scheme has the weakness of node compromise. If an adversary compromises with a sufficient number of nodes, she can capture the control of an entire network.

In this thesis, we only deal with providing security mechanisms between actors and sensors like as the two schemes for WASN. The proposed scheme is fully based on ID-based cryptosystem. So, the proposed scheme provides the stronger security compared with other

schemes [7, 37]. We also try to assign light overhead to sensors. Generally, operations for ID-based cyrptosystem requires big computation overhead, especially, the pairing operation is the expensive computation among them. Other AKA schemes utilizing ID-based cyrptosystem [38, 39] requires one pairing operation for each entity respectively. Instead, only actors compute one pairing operation, and sensors do not compute it in the proposed scheme. In addition, we consider Denial of Service (DoS) attack of routing layer which is introduced in [16]. To protect the attack, we utilize the location information of sensors to generate the public and private key pair. This characteristics makes the proposed scheme being resilient to DoS attack of routing layer.

## 1.2 Organization

The rest of this thesis is organized as follows: In Chapter 2, we introduce the background and related work for the proposed scheme. In Chapter 3, we present the proposed scheme and analyze the security and the performance of the proposed scheme. Finally, we make conclusion in Chapter 4.

# 2. Related work

In the chapter, we introduce the existing security work for WSN or WSAN. Especially, we survey key pre-distribution schemes[13, 8, 12, 18, 19] for WSN, security schemes[30, 40] for the mobile sink, security schemes[7, 37] for WASN, and access control schemes[33, 41, 20] for WSN.

## 2.1 Key pre-distribution schemes

### 2.1.1 Eschenauer and Gligor's scheme

Eschenauer and Gligor proposed the first random key pre-distribution scheme (denoted by EG02) to support secure communication between two nodes with pre-loaded $m$ keys. EG02 is composed of three phases, key pre-distribution, shared-key discovery, and path-key establishment. Based on the graph theory, EG02 analyzed the node connectivity, effect of node compromise, etc for various size of $m$.

The key pre-distribution phase is performed before deployment. A Trusted Authority (TA) first generates a large pool of $P$ keys. TA, then, randomly chooses $m$ keys for each node and load the $m$ keys into the corresponding node. Note that $P$ is sufficiently larger than $m$ (*i.e.,* $P \gg m$).

In the shared-key discovery phase, sensors broadcast their key information to neighbor nodes within transmission range. The sensors check whether their neighbor nodes share more than one key or not. For neighbor nodes which share a key, the sensors make secure link and confirm possession of the key through the challenge-response protocol with the mutually shared key.

In the path-key establishment phase, sensors share a key and establish a secure link with nodes which do not share any key within their transmission range. At this time, sensors utilize the secure links which are established in the shared-key discovery phase.

EG02 guarantees the connectivity of secure links based on the graph theory. Figure 2.1 depicts the probability of sharing at least one key for two nodes. The authors noted that 50% is proper connectivity level, and even if a large pool which of number is 10000, the number of $k$ to maintain 50% connectivity is 100 and it is very reasonable. However, in real application, EG02 may not provide complete connectivity. In addition, if there are

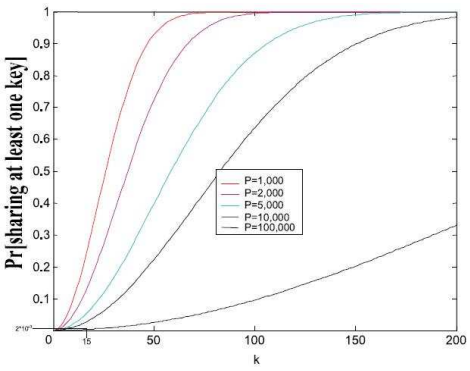


Figure 2.1: Probability of sharing at least one key when two nodes choose $k$ keys from a pool of size $P$

unexpected obstacles to disturb communication, the connectivity can be lower.

### 2.1.2 Chan, Perrig, and Song's scheme

Chan, Perrig, and Song proposed the improved version (denoted by CPS03) of EG02. CPS03 reduced the effect of key exposure problem, when nodes are compromised, by establishing a key of two nodes with $q$ keys which are shared by the two nodes. CPS03 has similar three phases with EG02 because it is modified version of EG02.

In the key pre-distribution phase, TA performs key pre-distribution work as the key pre-distribution of EG02. TA generates a key pool and loads keys into each node. However, TA should generate smaller key pool or load more keys into node than EG02 so that any two nodes can share $q$ keys as same probability as share a key in EG02.

In the shared-key discovery phase, sensors broadcast their key information and establish a $q$-composite key with neighbor nodes. Two nodes establish $q$-composite key by hashing the shared $q$ keys as follows. $H(key_0 \| ... \| key_q)$ where $H()$ is hash function, $key_0$ and $key_q$ are shared keys, and $\|$ means concatenation of bits.

In the path-key establishment phase, two nodes which do not establish a $q$-composite key try to share more keys through a secure link which is established in the shared-key discovery phase. If TA maintains the probability which two nodes share $m$ keys as much as two nodes share a key in EG02, CPS03 can guarantee the connectivity of secure links similarly.

It causes a trade-off to replace a shared key as a $q$-composite key. CPS03 achieves more

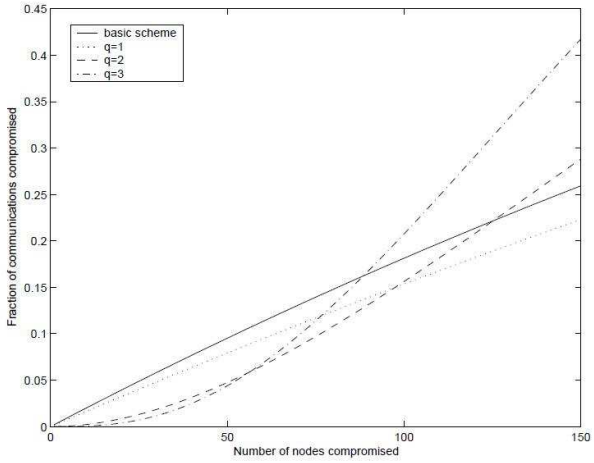


Figure 2.2: Comparing the effect of node compromise when the number of compromise nodes is increased

strengthened security under small scale attack which means attack with a small number of compromised nodes while it also increase vulnerability in the face of a large scale attack. Figure 2.2 depicts this trade-off relation. When the number of compromised nodes is smaller than 80, the compromised communication links are smaller than basic scheme which means EG02. The authors insist that the large scale attacks is not practical in real situation so CPS03 is more resilient to the node compromise.

### 2.1.3 Du, Deng, Han, and Varshney's scheme

Du, Deng, Han, and Varshney proposed a new improved key pre-distribution scheme (denoted by DDHV03) which is more resilient to node compromise compared with EG02 and CPS03. The essential idea of DDHV03 is to pre-distribute not random set of keys but random set of polynomials which nodes can utilize to establish a symmetric key. DDHV03 especially utilizes Blom's key pre-distribution scheme which is explained in the following.

We assume that a system manager generates $(\lambda + 1) \times N$ matrix $G$ over a finite field $GF(q)$, where $N$ is the size of the network and $q$ is a prime number which is bigger than $N$. $G$ is public information in Blom's setting. That is, all entities including an adversary know $G$. The system manager also creates a random $(\lambda + 1)(\lambda + 1)$ symmetric matrix $D$ over $GF(q)$ and computes an $N \times (\lambda + 1)$ matrix $A = (D \cdot G)^T$, where $(D \cdot G)^T$ is the transpose of $(D \cdot G)$. $D$ is the secret information and any entity except the system

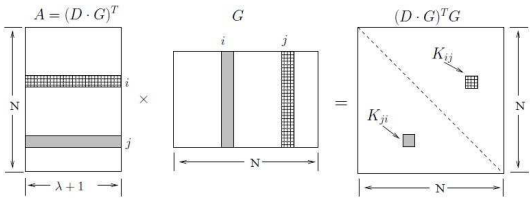


Figure 2.3: Generating keys in Blom's scheme

manager should not know the entire information. Due to symmetric property of $D$, the following equality holds.

$$A \cdot G = (D \cdot G)^T \cdot G = G^T \cdot D^T \cdot G = G^T \cdot D \cdot G = (A \cdot G)^T$$

The above equation means that $A \cdot G$ is a symmetric matrix, and we let $A \cdot G$ as $K$. We then know that $K_{ij} = K_{ji}$, where $K_{ij}$ is the element in the $i$th row and $j$th column of $K$. This result means that any two entities, where one has $i$th row in $K$ and other has $j$th column in $K$, can establish a symmetric key $K = K_{ij} = K_{ji}$ which can be generated with a row in $K$ and public information $G$. Furthermore, even though an adversary compromises an entity, it does not expose other keys which are established by other entities, until the adversary finds out the secret matrix $K$ or compromises entities as much as $\lambda$. DDHV03 applies probabilistic setting of EG02 to Blom's key pre-distribution.

Blom's key pre-distribution creates only a secret matrix $D$, but DDHV03 generates a set of $\omega$ secret matrixes $D = \{D_1, \ldots, D_\omega\}$. In the key pre-distribution phase, TA generates a public matrix $G$ and a set of secret matrixes $D$. TA also computes the matrix $K_i = (D_i \cdot G)^T$. We let $K_i(j)$ represent the $i$th row of $K_i$. TA then chooses $\tau$ $(2 \leq \tau < \omega)$ and randomly selects $\tau$ distinct matrixes from $\{K_1, \ldots, K_\omega\}$ for each node as TA selects randomly $m$ keys among a key pool in EG02. If $\tau$ matrixes for a node $i$ is noted as $K_\tau^i$, $i$ stores $i$th row from every $K \in K_\tau^i$.

After the key pre-distribution phase, the remain of DDHV03 is perform similar with other key pre-distribution schemes. Each node $i$ broadcasts $K_\tau^i$ and, when two neighbor nodes have row information in same matrix $K$, then establish a symmetric key. That is, two nodes $i$ and $j$ check $K_\tau^i \cap K_\tau^j \neq \phi$, where $\phi$ means empty set. If this equation is right, the two nodes perform Blom's key generation scheme in one of shared matrixes.

As seen in Figure 2.4, DDHV03 shows better resilience to node compromise compared with CSP03. The left figure is when the probability of sharing a key between any two nodes is 0.33, and the right one is 0.55. In Figure, $q$ means the number of keys used for

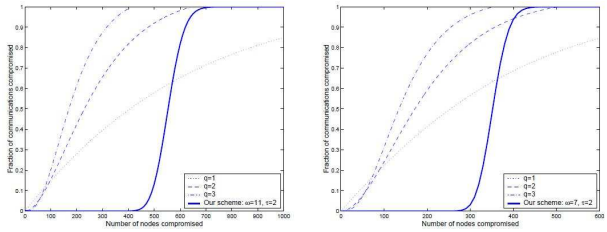


Figure 2.4: Comparing the resilience to node compromise

establishing $q$-composite key. We can confirm that DDHV03 is very resilient to node compromise when the number of compromised nodes is below a certain threshold. However, if the number of compromised node exceeds a certain threshold, it shows a rapid increase in the fraction of compromised link. The authors insist that this threshold is very reasonable because it is very difficult that an adversary compromises nodes as much as the threshold.

### 2.1.4 Liu and Ning's schemes

Liu and Ning proposed a key pre-distribution scheme (denoted by LN03a) which is very similar to DDHV03. LN03a utilizes a bivariate t-degree polynomial

$$f(x,y) = \sum_{i,j=0}^{t} a_{ij} x^i j^i$$

over a finite field $F_q$, where $q$ is larger than the size of a cryptographic key which will be used. The polynomial also has symmetric property of $f(x,y) = f(y,x)$. Let two nodes as $i$ and $j$. Node $i$ has a polynomial share of $f(x,y)$ which is $f(i,y)$, and node $j$ has $f(j,y)$, where $i$ and $j$ is a constant (you can think $i$ and $j$ are the identities of the two nodes) and $y$ is a variable. When the two nodes know other node's identity, they can compute a key $K_{i,j}$. Node $i$ computes $K_{i,j} = f(i,j)$ by putting $j$ as an input of $y$, and node $j$ computes $K_{i,j} = f(j,i)$ by putting $i$ as an input of $y$. Because of symmetric property of $f()$, the following equality holds $K_{i,k} = f(i,j) = f(j,i)$.

The function $f()$ is very similar to key generation used in Blom's key pre-distribution scheme. Actually, they are same, but $f()$ is generalized version of Blom's key pre-distribution. We can see that DDHV03 and LN03a are essentially same. The interesting fact is that two schemes were done separately and were presented at the same conference CCS'03. The contribution of LN03a is also same with it of DDHV03. LN03a remarkably increase the resilience to node compromise than EG02 and CSP03.

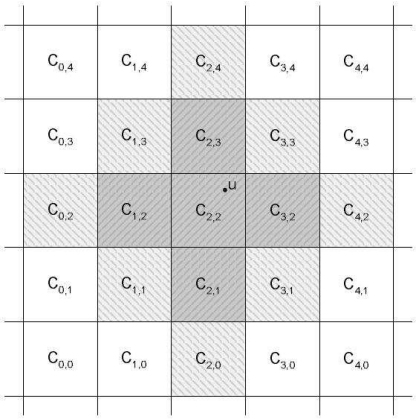


Figure 2.5: Partitioning a sensing field into several squares

LN03a is also performed as like DDHV03. In LN03a, TA first generates a set of $\omega$ polynomials and then randomly selects subset of $\tau$ distinct polynomials for each node. After deployment, each node broadcasts which polynomials and establishes a symmetric key with a shared polynomial.

The authors also modified LN03a with grid-based deployment and proposed modified scheme (denoted by LN03b), where a sensing field is separated by squares and each square has its own polynomial. Each node then stores the polynomial of the square where it is planed to deploy as well as neighbor squares which are adjacent to its home square. As seen in Figure 2.5, if we assume that a node $u$ will be deployed into the square $C_{2,2}$, $u$ should store the polynomials for both $C_{2,2}$ and adjacent squares which are $C_{1,2}$, $C_{2,1}$, $C_{2,3}$, and $C_{3,2}$. With location information, LN03b shows better performance and security than LN03a, in terms of the node connectivity, storage, and resilience to node compromise.

## 2.2 Security schemes for the mobile sink

### 2.2.1 Song, Zhu, Zhang, and Cao's scheme

Song, Zhu, Zhang, and Cao proposed a security scheme (denoted by SZZC08) for a mobile sink [30]. In SZZC08, it is addressed that how to give the least privilege which is need to perform a certain task to a mobile sink.

For solving the problem, they assumed that $BS$ already knows a route of a mobile sink and which sensors exist in the route before deployment of the sink. They also assumed

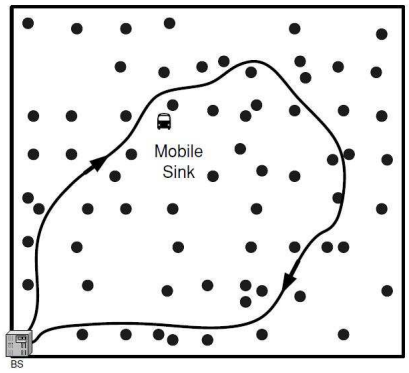


Figure 2.6: An example of a mobile sink node

that the sink will be come back after performing a pre-determined task. If we assume a route of the sink as seen in Figure 2.6, $BS$ will assign a list of accessible sensors and a task type into the sink. When the sink finishes the assigned task, it returns to $BS$.

The authors first proposed a basic scheme for the sink to access the pre-determined sensor and then improved the scheme in phases. Their first scheme is as follows: Before deployment, every node is pre-loaded with a key shared with $BS$. $BS$ generates a master key $K_m$ for a mobile sink and derives a pair-wise key between $BS$ and all nodes $u$ in a route of the sink as $K_u = G_{K_m}(u)$, where $G$ is a pseudo-random function and $u$ is an identity of sensors. $BS$ then generates a pair-wise key between the sink $m$ and $u$ as $K_u(MS) = H(TT|MS|K_u|T_s|T_e)$ where $H$ is a collision-resistant one-way hash function, $\|$ denotes the concatenation of messages, $TT$ is the task type, $T_s$ and $T_e$ are the starting time and the ending time of a task, and $MS$ is the identity of the sink. To establish a pair-wise key with a node $u$, the sink sends $MS$, $TT$, $T_s$, and $T_e$ to $u$. On the receipt of the message, $u$ computes $K_u(MS)$. After establishing a pair-wise key, $MS$ and $u$ exchange a message authentication code (MAC) with their identity and random nonce to protect replay attack.

The basic scheme can restrict the privilege of the mobile sink for the only pre-determined task with the pre-determined nodes. However, it is not scalable in terms of storage if the mobile sink is expected to access a large number of nodes. The authors then improved the basic scheme with the Blundo scheme which is used in [18, 19]. Before deployment of sensors, $BS$ determines a random symmetric bivariate polynomial $f(x, y)$ of degree $t$ with coefficients over a finite field $GF(q)$. $BS$ then loads every node $n$ with $f(n, y)$ which is a polynomial obtained by evaluating $f(x, y)$ at $x = n$. Their first scheme which is based on

the Blundo scheme is operated as follows. $BS$ first determines $TT$, $T_s$, and $T_e$ of a mobile sink $MS$ and then constructs the identity of $MS$ as $MS(u) = H(TT|T_s|T_e|u)$ for sensors $u$ which $MS$ can access. $BS$ then pre-loads $MS$ with a polynomial share of $f(MS(u), y)$. $MS$ and $u$ can share a key $f(MS(u), u) = f(u, MS(u))$ when $MS$ sends $(TT, T_s, T_e)$ to $u$. After establishment of a pair-wise key, $MS$ and $u$ can authenticate each other as the basic scheme.

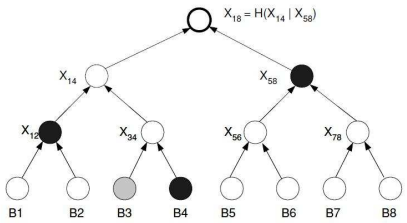


Figure 2.7: An example of the Merkle hash tree

The first scheme, based on the Blundo scheme, also has a big overhead of storages because $MS$ should store $m(t + 1)$ coefficients to communicate with $m$ sensors. To reduce the number of polynomial shares, the authors assumed the grid-based deployment, where before deployment of sensors, a set of sensors is pre-determined and is deployed to each grid. BS then generates the identity of $MS$ not for a sensor but for all sensors in a grid. The identity of $MS$ for a grid which is in $i$th row and $j$th column is computed as $MS(i, j) = H(TT|T_s|T_e|i|j)$. The rest of the second scheme is same with that of the first scheme.

To optimize the second scheme, they introduced the block compression which makes adjacent grids as one rectangle block. BS then generates the identity of $MS$ for each block as $MS = H(TT|T_s|T_e|X_{1m})$ where $X_{1m}$ is root value of a Merkle hash tree. The Merkle hash tree is generated as seen in the Figure 2.7. In Figure 2.7, $B_i$ means $i$th block, and $MS$ can access eight blocks $(B_1, \ldots, B_8)$. The root value $X_{18}$ will be computes as $X_{18} = H(X_{14}|X_{58})$, $X_{14} = H(X_{12}|X_{34})$, and $X_{12} = H(B_3|B_4)$. $BS$ loads $MS$ with a polynomial share $f(MS, y)$. $MS$ then can generate a pair-wise key with every sensor in $(B_1, \ldots, B_8)$. For example, to establish a key with a node $u$ in $B_3$, $MS$ sends $MS, TT, T_s, T_e$ and $X_{18}$ as well as several values $B_3, B_4, X_{12}$ and $X_{58}$ of the Merkle tree allowing $u$ to verify $X_{18}$. On the receipt of this message, $u$ first checks whether itself is in $B_3$ and then confirms $X_{18} = H(H(X_{12}|H(B_3|B_4))|X_{58})$. After this confirm is done well, $u$ generates the identify of $MS$ and shares a pair-wise key $f(u, MS) = f(MS, u)$. The rest of this scheme is same

with that of other schemes.

### 2.2.2 Zhou and Ravishankar's scheme

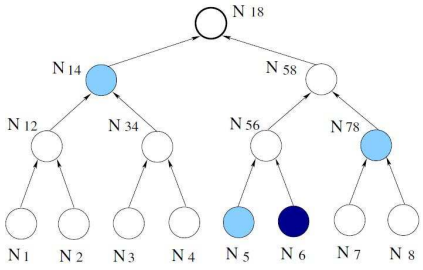


Figure 2.8: A dynamic credential tree

Zhou and Ravishankar proposed a security scheme (denoted by ZR06) based on dynamic merkle trees to allow more efficient access control of mobile sinks [40]. They assumed the followings. First, grid-based deployment of sensors and broadcast authentication of $BS$ is supported. Second, each sensor knows its own grid, time of nodes is synchronized. Finally, mobile sinks and all sensors already share a pair-wise key.

They utilized a dynamic credential tree (DC-tree) as seen in Figure 2.8. In the dynamic credential tree, there are $m = |M| \times |T|$ leaf nodes, where $|M|$ is the number of mobile sinks, and $|T|$ is the number of transaction types. Leaf node $N_i$ in the DC-tree is initially given the contents $\widehat{N_i} = H(C_{\alpha_i}(0), \alpha_i)$, where $\alpha_i$ is an activity done by a mobile sink, $C_{\alpha_i}(x)$ is a one-way hash chain, which is made as $C_{\alpha_i}(i) = C_{\alpha_i}(i+1)$, for $\alpha_i$, $C_{\alpha_i}(0)$ is the first value of $C_{\alpha_i}(x)$, and $H$ is a one-way hash function. $\alpha_i$ is made as $\alpha_i = H(ms, tt, r)$, where $ms$ is an identity of mobile sinks, $tt$ is a transaction type, and $r$ is a region which may be a set of grids.

After deployment of sensors, $BS$ generates the initial DC-tree and broadcasts its corresponding root value $\tau$ and hash values $C_\alpha(0)$ for all activities to all nodes. In Figure 2.8, $\tau$ is same with $N_{18}$. We show how the scheme is performed in the case of $\alpha_6$. First, a mobile sink, which will perform $\alpha_6$, is loaded with $\alpha_6$, $C_{\alpha_6}(1)$, and hash values to prove the new root value $N'_{18}$, $\{\widehat{N_5}, \widehat{N_{78}}, \widehat{N_{14}}\}$. The mobile sink then moves to the region $\alpha_6.r$ and broadcasts the message which consists of the above tuples. On the receipt of the message, sensors in $\alpha_6.r$ first confirm $C_{\alpha_6}(0) = H(C_{\alpha_6}(1))$ and further compute new $\widehat{N_6}$ as $H(C_{\alpha_6(1)}, \alpha_i)$. The nodes then compute the root value as $H(\widehat{N_{14}}, H(H(\widehat{N_5}, \widehat{N_6}), \widehat{N_{78}}))$ and confirm whether this value is same with $N'_{18}$. If it is right, the nodes believe that the mo-

bile sink is valid and send an accept message with MAC made by pre-distributed key. If this is wrong, they ignore the mobile sink.

In addition to above process, $BS$ separately should perform the following process. After an activity $\alpha_i$ is done, a set of nodes $\overline{N}$ should update their root value of DC-tree as a new root value, where $N$ is all nodes in $\alpha_i.r$, and $\overline{N}$ is the complementary set of $N$. $BS$ broadcasts a new root value with a broadcast authentication scheme.

## 2.3 Security schemes for WSAN

### 2.3.1 Cao, Huang, Chen and Chen's scheme

Cao, Huang, Chen, and Chen proposed an Authentication and Key Management (AKM) scheme (denoted by CHCC05) for WSAN[7]. Their scheme consists of three layers, Actor-Actor ($A-A$) layer, Actor-Sensor ($A-S$) layer, and Sensor-Sensor ($S-S$) layer. In each layer, they provided authenticated key transport between two entities. AKM of $A-A$ layer utilized public key cryptosystem by considering $BS$ as Certificated Authority ($CA$). AKM of $A-S$ layer is based on private key cryptosystem with a pre-distributed key which is shared by all entities including actor and sensors. AKM of $S-S$ layer utilized secure links which are generated in AKM of $A-S$ layer.

AKM of $A-A$ layer works as conventional public key based schemes. Each actor is loaded with a public key, a private key, the public key of $CA$ and a certificate of its public key. For AKM, two actors first generate a random nonce and generate a digital signature with their private key, their identity, and the random nonce. The actors then exchange the massage consists of the random nonce, the identity, and the digital signature. If the digital signature is verified, they generate a key with a hash function using the two random nonces as inputs. The following shows detail process of AKM of $A-A$ layer. $E_{PK}(M)$ is encryption of message $M$, $E_{SK}(M)$ is signing of message $M$, $N_A$ is a random nonce made by $A$, $H$ is a hash function, and $F$ is a key generation function.

$A \longrightarrow B : E_{PK_B}(A||N_A||E_{SK_A}(H(A||N_A)))$

$A \longleftarrow B : E_{PK_A}(N_A||N_B||E_{SK_B}(H(N_A||N_B)))$

$A \longrightarrow B : E_{PK_B}(N_B||E_{SK_A}(H(N_B)))$

$A$ and $B$ : so far, if verification is ok, $A$ and $B$ share $K_{AB} = F(N_A||N_B)$

AKM of $A-S$ layer works as follows: The authors assumed that no attack is occurred until some time $T_{no}$ and $T_{no}$ is longer than a time which is required to perform AKM

of $A-S$ layer. Before deployment of WSAN, $BS$ loads all nodes with a key $K_{all}$. After deployment, each actor $A$ broadcasts its identity, and each sensor $B$ chooses the closest actor among them. Then, $A$ and $B$ perform the following. When all sensors have at least a secure link with an actor, all entities remove $K_{all}$.

$$A \longleftarrow B : E_{K_{all}}(B||N_B||H(B||N_B))$$

$$A \longrightarrow B : E_{K_{all}}(A||N_A||B||N_B||K_{AB}||H(A||N_A||B||N_B||K_{AB}))$$

$$A \longleftarrow B : E_{K_{AB}}(N_A)$$

AKM of $S-S$ layer utilized the secure links of $A-S$ layer. So far, other AKM has only two entities, but AKM of $S-S$ also has three entities. When two sensors $B_1$ and $B_2$ authenticate each other and share a key $K_{B_1B_2}$, one of them requests an actor which shares a key with the two nodes. The actor then transport $K_{B_1B_2}$ with secure keys, $K_{AB_1}$ and $K_{AB_2}$. The detail is performed as follows:

$$A \longleftarrow B_1 : (B_1||B_2||N_1)$$

$$A \longrightarrow B_1 : E_{K_{AB_1}}(K_{B_1B_2}||B_1||B_2||N_1||E_{K_{AB_2}}(K_{B_1B_2}||B_1))$$

$$B_1 \longrightarrow B_2 : E_{K_{AB_2}}(K_{B_1B_2}||B_1)$$

### 2.3.2 Yu, Ma, Wang, Mao, and Gao's scheme

Yu, Ma, Wang, Mao, and Gao proposed an Authenticated Key Transport (AKT) scheme (denoted by YMWMG06) between an actor node and a sensor in WSAN [37]. Their scheme assumed a grid-based deployment where a grid consists of $M$ sensors and a actor node, and the number of grid is same with the number of actors as $N$. As seen in Figure 2.8, $BS$ generates a merkle tree for actors and $N$ merkle trees for sensors in $N$ grids.

The merkle tree of actors is made as follows: A leaf of the tree is made by hashing an identity and a public key of an actor. In the case of $A_1$, the leaf $L_1$ is made as $L_1 = H(A_1||PK_{A_1})$. The root value of the tree is noted as $R_A$ and the least number of hash values to reconstruct $R_A$ with $L_1$ are noted as $\Phi(L_1)$. The merkle tree of sensors is also made in similar way. A different point is that each grid has its own Merkle tree. We note a sensor $i$ in the grid 1 as $S_i^1$ and the leaf of $S_i^1$ as denoted $L_i^1 = H(S_i^1||PK_{S_i^1})$, where $S$ is some secret information which is only known to $BS$. The root value of the grid $i$ is denoted as $R_i$.

After generating the trees, $BS$ loads each actor $A_i$ with a public key, a private key, $L_i$, $\Phi L_i$, and $R_{allow}^i$, where $R_{allow}$ denotes a set of the root value of grids are allowed for

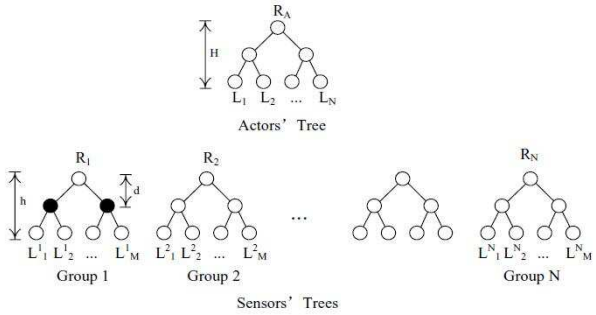


Figure 2.9: Merkle forests

$A_i$ to access. $BS$ also loads each sensor $S_j^i$ with $L_j^i$ and $\phi L_j^i$. After deployment, an actor $A_1$ and a sensor $S_1^1$ perform the following:

$A_1$ : generates a random nonce $N_{A_1}$

$A_1 \longrightarrow S_1^1 : (\Phi(L_1), PK_{A_1}, A_1, E_{SK_{A_1}}(N_{A_1}, H(\Phi(L_1)||PK_{A_1}||A_1)))$

$S_1^1$ : confirms $PK_{A_1}$ with $\Phi(L_1)$, $R_A$, and $L_1 = H(A_1||PK_{A_1})$

$S_1^1$ : generates $N_{S_1^1}$

$A_1 \longleftarrow S_1^1 : E_{PK_{A_1}}(\Phi(L_1^1), S_1^1, N_{S_1^1})$

$A_1$ : checks that $R_1$ can be reconstructed by $\Phi(L_1^1)$ and $S_1^1$

## 2.4 Access control schemes

### 2.4.1 Wang and Li's scheme

Wang and Li proposed two user access control schemes (denoted by WL06) in sensor networks [33]. The first scheme is based on Elliptic Curve Cryptography (ECC), and the other scheme is based on blundo's polynomial scheme. Their schemes enable a new sensor to authenticate existing nodes and establish a key.

The first scheme works as follows: $BS$ chooses a particular elliptic curve over a finite field $GF(p)$ and publishes a point $P$ with order $q$, where $p$ and $q$ are prime numbers. $BS$ then picks a random number $x \in GF(q)$ as the system private key and generates the corresponding public key $Q = x \times P$. For an user $A$, $BS$ picks a random number $c_A \in GF(q)$ and then calculate the user's public key constructor $C_A = c_A \times P$. $BS$ also

issues a proper access control list $ac_A$ and generates $T_A = (C_A||ac_A)$ and $e_A = H(T_A)$, where $H$ is a hash function. $BS$ further generates $A$'s private key $q_A = e_A c_A + x$ and public key $Q_A = q_A \times P = e_A \times C_A + Q$. $BS$ sends $q_A, Q_A$, and $T_A$ to user $A$ through a secure link. The user $A$ can authenticate and share a key with node $s_i$ as the followings, where $X(Z_r)$ is the $X$ coordinate of point $Z_r$, and $MAC(K, M)$ is a message authentication code algorithm with $K$ as a key and $M$ is a message.

$A \longrightarrow s_i : T_A = (C_A||ac_A)$

$s_i$ computes : $Q_A = e_A \times C_A + Q$

: picks a random $r \in GF(p)$

: $Z_r = H(r) \times Q_A$

: $Y_r = H(r) \times P$

: $z_r = r \oplus X(Z_r)$

: $MAC(r, N_A)$

$A \longleftarrow s_i : z_r, Y_r, MAC(r, N_A)$

$A$ computes : $q_A \times Y_r = q_A \times H(r) \times P = Z_r$

: $X(Z_r) \oplus z_r = r$

: decrypt $MAC(r, N_A)$

$A \longrightarrow s_i : MAC(r, (N_A||ac_A))$

$A \longleftarrow s_i : MAC(r, MAC(r, (N_A||ac_A)))$

For the second scheme, the authors assumed a grid-based deployment, where a set of sensors is deployed into each grid. Before the deployment, sensors are divided into $k$ groups $\{g_1, g_2, \ldots, g_k\}$, where $g_j$ $(1 \leq j \leq k)$ is a group ID. We note a sensor $s_i$ in a grid $j$ as $s_i^j$. Before the deployment, $s_i^j$ is loaded with two shares of polynomial,$f(x, s_i)$ and $f(x, g_j)$. In this scheme, when a user $A$ and a sensor $s_i^g$ perform the scheme to authenticate each other and share a key, $A$ should obtain confirmation from $k$ sensors $s_i^j$, where $j \neq g$ and $k$ is a system parameter. The detail of the scheme works as follows:

$A$ finds $k$ local sensors $^j_i$ with different $j \neq g$

$A \longrightarrow k$ sensors $s_i^j$ : broadcast the request message

$A \longleftarrow s_i^j$ : group id

$A \longrightarrow s_i^j$ : confirm request

$s_i^j$ authenticate user access list $T_A$

$A \longleftarrow s_i^j : mac_i = MAC(f(s_i, g_i), ac_A)$

$A$ computes : $mac = H(mac_1 || \ldots || mac_k)$

$A \longrightarrow s_i^g$ : $(MAC(mac, ac_A || N_A) || ac_A || grouplist)$

$s_i^g$ : computes $f(g_1, s_r), \ldots, f(g_k, s_r)$

$s_i^g$ : reconstruct $mac = H(mac_1 || \ldots || mac_k)$

$s_i^g$ : decrypt and verify $ac_A$

$A \longleftarrow s_i^g : MAC(mac, reqply || N_A || N_B)$

### 2.4.2 Zhou, Zhang, and Fang's scheme

Zhou, Zhang, and Fang proposed an access control scheme (denoted by ZZF07) with Elliptic Curve Digital Signature Algorithm (ECDSA) [41]. Before the deployment, $BS$ performs the followings. $BS$ first chooses a set of network parameters: a finite field $\mathbb{F}_q$, where $q$ i a large odd prime of at least 160 bits; an elliptic curve $E$ over $\mathbb{F}_q$ denoted by $E(\mathbb{F}_q)$; a cyclic group $\mathbb{G} =< G >$ of points over the elliptic curve $E(\mathbb{F}_q)$, where $G$ is the generator of the group and has an order $n$ of at least 160 bits, with $n > 4\sqrt{q}$; the $BS$'s private key $\kappa \in \mathbb{Z}_n^* = \{1, 2, \ldots, n-1\}$; the $BS$'s public key $Q = \kappa G \in \mathbb{G}$.

$BS$ then pre-loads a sensor $N_i$ with a set of node parameters: $E(\mathbb{F}_q)$; $\mathbb{G}$; $Q$; the bootstrapping time $T_i$ when $N_i$ bootstraps itself to join the sensor networks; the length of bootstrapping phase $L_i$ during which the node is allowed to join the sensor networks; $N_i$'s private key $s_i \in \mathbb{Z}_n^*$; $N_i$'s public key $P_i = s_i G = (x_{pi}, y_{pi}) \in \mathbb{G}$, where $x_{pi}, y_{pi} \in \mathbb{F}_q$; the signature $< C_i, c_i >$, where $C_i \in \mathbb{G} = k_i G = (x_{ci}, yci)$ and $c_i \in \mathbb{Z}_n^* = k_i^{-1}(H(N_i || T_i || L_i || P_i) + \kappa x_{ci}) \pmod{n}$, where $k_i$ is a random number of $\mathbb{Z}_n^*$ chosen by $BS$; a hash function $H : \{0,1\}^* \rightarrow \mathbb{Z}_n^*$.

After the deployment, every new node broadcasts the message $\{N_i, T_i, L_i, P_i, C_i, c_i\}$. On the receipt of this message, a node $N_j$ first checks that $T_i$ and $L_i$ are valid and then computes a verifier $V_i$ as follows:

$u_1 = H(N_i||T_i||L_i||P_i)$

$u_2 = c_i^{-1}u_1 \pmod{n}$

$u_3 = c_i^{-1}x_{ci} \pmod{n}$

$V_i = u_2G + u_3Q$

If $V_i = C_i$, $N_j$ believes that $N_i$ is a valid new node and then sends $\{N_j, T_j, L_j, P_j, C_j, c_j\}$. On the receipt of the message, $N_i$ generates $V_j$ and confirms $V_j = C_j$. After this procedure, they can share a key $K_{ij} = s_iP_j = s_jP_i = s_is_jG$. After establishing $K_{ij}$, they confirm sharing $K_{ij}$ by exchanging messages including mac with $K_{ij}$.

### 2.4.3 Liu's scheme

Liu proposed an access control scheme (denoted by Liu07) which can delegate a right of an user to other users and support a broadcast-based access control [20]. First, we introduce his basic scheme where an user can access a sensor at once. In the basic scheme, $BS$ determines privilege of an user $A$ and generates a key for each sensor $i$ which $A$ can access as $K_{A,i} = H(ID_A||C_A||K_i)$, where $ID_A$ is the identity of $A$, $C_A$ is the constraint of the privilege of $A$, and $K_i$ is a key shared by $BS$ and sensor $i$. $BS$ then loads $A$ with these keys. To obtain access from sensor $i$, $A$ sends the message $C_A, Q(A), MAC_{K_{A,i}}(C_A||Q(A))$, where Q(A) is a query message of $A$, and $MAC_K(M)$ is a mac algorithm using $K$ as a key and $M$ as an input message. On the receipt of this message, sensor $i$ constructs $K_{A,i}$ from $K_i$ and confirms the privilege of $A$.

Liu modified the basic scheme for an user $A$ to enable delegate its privilege. The idea is quite simple. When an user $A$ delegates its privilege to an user $B$, $A$ first determines the constraint $C_B$ for $B$. $A$ then assign a key $K_{B,i} = H(B||C_B||K_{A,i})$ for each sensor $i$. With these keys, $B$ can obtain form $i$.

Liu also proposed the access scheme to support a broadcast query. Before deployment, an user $A$ obtains a pair of private and public keys $(SK_A, PK_A)$ and a certificate $Cert_A$. To perform the broadcast query, $A$ first picks out a sensor $i$ among several nodes and then sends the message $M_A$ which consists of $C_A$, $PK_A$, $Cert_A$, $Q(A)$, and a signature $Dig_{SK_A}(Q(A))$ which is made with $SK_A$ as the private key and $Q(A)$ as an input message. On the receipt of $M_A$, the sensor $i$ confirms that it is valid and broadcast $M_B$ with a commitment $H(M_B||K_{i,j})$ for each neighbor sensor $j$. Sensor $j$ first checks $H(M_B||K_{i,j})$ to confirm this message is from sensor $i$ and then confirms $Dig_{SK_A}(Q(A))$.

# 3. Proposed scheme

## 3.1 Preliminaries

### 3.1.1 Denial of Service (DoS) Attacks

WSAN consists of a lot of sensors and actors in large scale area and aggregates data with ad-hoc routing. These properties of WSAN enable DoS attacks which obstruct valid network operation. There are various DoS attacks, but their mechanism is same. They utilize that sensors need to transmit their data to an actor with multi hop routing and interrupt that the actor obtains sensed data from sensors [16]. [1] In this subsection, we introduce three famous attacks, Sybil attack, Identity replication attack, and Wormhole attack.

- **Sybil attack:** Sybil attack [16, 24] is performed by a malicious sensor which behaves as if it were a large number of nodes. That is, a node impersonates other nodes or simply claiming multiple forged identities. Sybil attack is extremely harmful to many important tasks of WSAN such as routing and data aggregation of actors.

- **Identity replication attack:** Identity replication attack [24] happens when an adversary loads multiple replicas of a compromised sensor in different geographic locations. This attack makes actors recognize that replicated nodes are valid.

- **Wormhole attack:** In wormhole attacks, two malicious sensors, which are connected with low-latency communication link, are deployed in a little distant location. By collecting messages and relaying them, they make that their neighbor nodes are confused as if they coexist in closed range where they communicate each other. This attack can jeopardize routing and data aggregation.

[1] There are other DoS attacks, but, in this thesis, we do not concern these attacks.

### 3.1.2 Network Model

In this subsection, we describe a network model of WSAN for the proposed scheme. Generally, WSAN consists of three kinds of nodes, a sink node, sensors, and actors. In the network model, we assume that sensors are resource-limited sensors such as Mica2 motes. Sensors have a wireless transmitter to enable short-range communication, a low computation unit, a small storage, and a sensing unit to obtain data. Sensors also have no mobility. On the other hand, actors have more resources than sensors. Actors have two wireless transmitters, one for communication of sensors and the other for communication of actors, a computation unit, a storage unit, and a decision unit, which is to determine whether an event happens or not from data of sensors. Depend on applications, the capability of actors will be changed, but, in this thesis, we consider that an actor has resources as much as a laptop computer or a PDA. The sink node is desktop computer and protected by service administrator.

WSAN is operated as follows: First, sensors obtain data from the sensing filed and transmit this data to the closest actor. The actor then decides some event is occurred. If the event happens, the actor lets other actors know that the event is occurred. They discuss that how many actors should deal with it. Then, actor(s) move to the location where the event happens and handle it. As above, actors should move when an event is occurred. When it is over, actors should receive data from sensor. Actors must have the right to access for almost all sensors because of their mobility.

### 3.1.3 Treat Model

We assume that the adversary can compromise multiple both actor(s) and sensor(s) and further assume that if a node is compromised, all the information it holds will also be compromised. The adversary can take full control of compromised nodes and thus can manipulate compromised nodes to drop or alter messages going through them with pretending valid nodes. However, we assume that the sink node is secure against the adversary because it is usually well protected and under the direct control of the network owner. We also assume that the adversary can eavesdrop on all traffic, inject packets, and replay old packets.

### 3.1.4 Security Requirements

In the subsection, we present the security requirements for AKA schemes in WSAN. The AKA schemes should guarantee these requirements. Note that we only consider AKA be-

tween sensor and actors. For AKA between actors, existing AKA schemes [6, 9] are useful, so we do not address it. AKA between sensors is an important research topic, but we remain it as our future work.

1. **Authentication:** The scheme should provide mutual authentication of two entities. That is, an attacker cannot impersonate a valid sensor or actor node without compromising the node. Even if a node is compromised, the scheme should guarantee that the attacker cannot impersonate other nodes except the compromised node.

2. **Key security:** After some two entities agreed a key, the scheme should guarantee that every entity except themselves and BS cannot compute the agreed key. The compromised node should not expose agreed keys of other nodes.

3. **Resilient to DoS attacks:** Karlof *et al.* introduced several DoS attacks for WSN[16]. They identified sybil attack, hello flood attack, and wormhole attack. The scheme should be secured against these attacks.

### 3.1.5 Bilinear Map

Let $\mathbb{G}_1$ be a cyclic additive group of prime order $q$ and $\mathbb{G}_2$ be a cyclic multiplicative group of same order $q$. We assume that the discrete logarithm problem (DLP) in both $\mathbb{G}_1$ and $\mathbb{G}_2$ is intractable. We call $e$: $\mathbb{G}_1 \times \mathbb{G}_1 \longrightarrow \mathbb{G}_2$ an bilinear map if it satisfies the following properties:

1. Bilinearity: $e(aP, bQ) = e(P, Q)^{ab}$ for all $P, Q \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_q^*$.
2. Non-degenerancy: If $\mathbb{G}_1 =< P >$, then $\mathbb{G}_2 =< e(P, P) >$.
3. Computability: There exists an efficient algorithm to compute $e(P, Q)$ for all $P, Q \in \mathbb{G}_1$.

The Weil [5] and Tate [3] pairings in elliptic curve are examples of such a bilinear map.

### 3.1.6 Hard Problems

We assume that the following hard problems are intractable similar to [10, 17]. That is, there is no polynomial time algorithm solving these problems with non-negligible probability.

- **Computational Diffie-Hellman (CDH) problem:** The CDH problem is to compute $abP$ when given, $P$, $aP$ and $bP$ for some $a$, $b$ *in* $\mathbb{Z}_q^*$.
- **Modified Inverse Computation Difiie-Hellman (mICDH) problem:** The mICDH problem is to compute $(a+b)^{-1}P$ when given $b, P, aP$ and $(a+b)P$ for some $a, b \in \mathbb{Z}_q^*$.
- **Bilinear Diffie-Hellman (BDH) problem:** the BDH problem is to compute $e(P,P)^{abc}$ when given $P, aP, bP$ and $cP$ for some $a, b, c \in \mathbb{Z}_q^*$.
- **Modified Bilinear Inverse Diffie-Hellman (mBIDH) problem:** The mBIDH problem is to compute $e(P,P)^{\frac{1}{a+b}c}$ when given $b, P, aP$ and $cP$ for some $a, b, c \in \mathbb{Z}_q^*$.

## 3.2 Scheme Description

In the section, we propose location-based AKA scheme between actor and sensors in WSAN. We consider asymmetric resource of sensor and actors. Generally, actors have more resources than sensors, so, we try to assign light overheads for sensors in the proposed scheme.

### 3.2.1 Assumptions

For the proposed scheme, we assume that actors are resource-rich in terms of computation, storage and battery and have mobility. We further assume that the actors have Global Positioning System (GPS) capability. These assumptions are general in WSAN, and most security schemes for WSAN [37, 7] also assumed them. We assume that sensors are low-power, low-cost devices such as MICA2 mote. The sensors have no mobility, so they are static after deployment.

For deployment of sensors, we assume that a practical approach such as [11, 28] is used. In the approach, mobile robots, which are similar to actors, are used to deploy and localize individual sensors. Before deployment, actors (mobile robots) are equipped with several sensors. Then, during deployment phase, the actors drop the sensors according to the predetermined plan. At that time, the actors transmit the x and y coordinate values of the deployment position. During the deployment phase, we also assume that there is no compromise of the actors.

### 3.2.2 Setup

Before deployment of sensors, a trusted authority (TA) (*e.g.,* the system administrator or network planner) performs the following operations.

1. TA determines two groups $\mathbb{G}_1, \mathbb{G}_2$ and a bilinear map $e$ as described in preliminaries.
2. TA chooses three cryptographic hash functions $h : \{0,1\}^* \longrightarrow \mathbb{Z}_q^*$, $h_1 : \{0,1\}^* \longrightarrow \{0,1\}^t$ and $h_2 : \mathbb{Z}_q^* \longrightarrow \{0,1\}^t$ where $t$ is the size of session key.
3. TA computes $g = e(P, P)$, where $P$ is a generator of $\mathbb{G}_1$.
4. TA picks a secret value $\kappa \in \mathbb{Z}_q^*$ and then sets the public key of TA as $P_{pub} = \kappa P$.
5. For each actor node $A_i$, TA computes a public key as $PK_{A_i} = h(ID_{A_i})P + P_{pub}$ and a private key as $SK_{A_i} = (h(ID_{A_i}) + \kappa)^{-1}P$ where $ID_{A_i}$ is an identity of the actor $A_i$.
6. For each sensor $S_i$, TA computes Id-Based Key as $IBK_{S_i} = h_2(\kappa h(ID_{S_i}))$.

TA then loads the public system parameters $< p, q, \mathbb{G}_1, \mathbb{G}_2, e, h, h_1, h_2, P, P_{pub}, g >$, $ID_{A_i}$, key pair $(PK_{A_i}, SK_{A_i})$ and $\kappa$ into each actor node $A_i$. TA also loads the public system parameters $< p, q, \mathbb{G}_1, \mathbb{G}_2, e, h, h_1, P, P_{pub}, g >$, $ID_{S_i}$ and $IBK_{S_i}$ into each sensor $S_i$.

### 3.2.3 Generation of location-based keys

For deployment of sensors, we assume the approach proposed in [11, 28]. This approach uses mobile robots (actors) to deploy and localize sensors. After pre-deployment, each actor node equips several sensors to deploy and receives deployment information from TA. The actors then deploys sensors according to the deployment information.

The proposed scheme utilizes the geographic information of a sensor to generate its public key and private key pair. Therefore, actor node transmits a proper key pair to a sensor when the sensor is just deployed. They execute the protocol in Figure 3.1.

An actor $A_i$ transmits a hello message to a sensor $S_i$ which is just deployed. After receiving this message, $S_i$ replies its id $ID_{S_i}$. $A_i$ then makes $IBK_{S_i}$ and $pos_{S_i}$ respectively as $IBK_{S_i} = h_2(\kappa h(ID_{S_i}))$ and $pos_{S_i} = (x_{S_i}||y_{S_i})$ where $x_{S_i}$ and $y_{S_i}$ are x and y coordinate values of the deployment position of $S_i$. $A_i$ further generates location-based public and private key pair of $S_i$ as $LPK_{S_i} = h(pos_{S_i})P + P_{pub}$ and $LSK_{S_i} = (h(pos_{S_i}) + \kappa)^{-1}P$. Finally, $A_i$ encrypts $ID_{S_i}$, $pos_{S_i}$, $PK_{S_i}$ and $SK_{S_i}$ with a symmetric encryption scheme

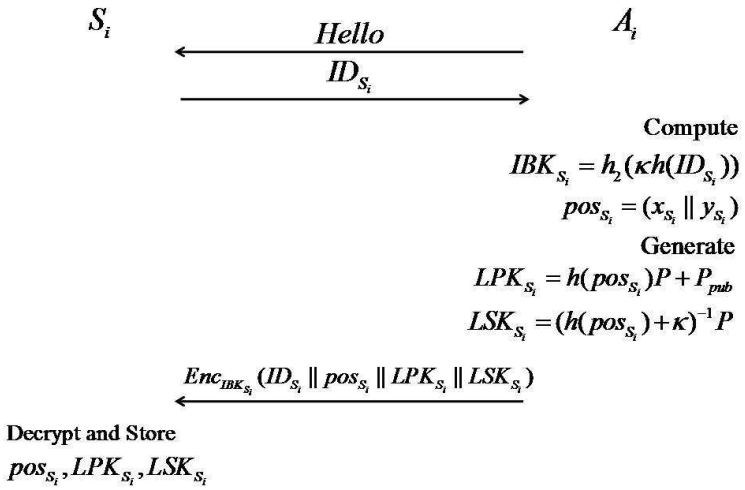


Figure 3.1: Generation of location-based keys

(*e.g.,* AES and DES) and the symmetric key $IBK_{S_i}$ and transmits the encrypted message to $S_i$. After receiving the massage, $S_i$ decrypts this message using the preloaded $IBK_{S_i}$, checks that it is valid and then stores $pos_{S_i}$, $LPK_{S_i}$ and $LSK_{S_i}$. Note that when $A_i$ finishes deployment process for all sensors which $A_i$ equips, $A_i$ removes the secret value $\kappa$.

### 3.2.4 Location-Based Authenticated Key Agreement

To authenticate and establish session keys, an actor $A_i$ and a sensor $S_i$ perform the protocol in Figure 3.2.

$A_i$ generates a random value $R_{A_i}$ from $\mathbb{Z}_q^*$ and then transmits a message which consists of its id $ID_{A_i}$ and $R_{A_i}$. After receiving it, $S_i$ generates a random value $R_{S_i}$ from $\mathbb{Z}_q^*$ and computes $sk, X$ and $Y$ as $sk = h(g^{R_{S_i}}||R_{A_i}||pos_{S_i}||ID_{A_i})$, $X = R_{S_i}PK_{A_i} = R_{S_i}h(pos_{S_i})P + R_{S_i}P_{pub}$ and $Y = (R_{S_i} + sk)LSK_{S_i}$. $S_i$ then sends $pos_{S_i}, X$ and $Y$ to $A_i$. When $A_i$ receives this message, $A_i$ first computes $e_{S_i} = e(X, SK_{A_i}) = g^{R_{S_i}}$ and $sk' = h(e_{S_i}||R_{A_i}||pos_{S_i}||ID_{A_i})$. After computing $e_{S_i}$ and $sk'$, $A_i$ verifies that the following equation holds :

$$e(Y, h(pos_{S_i})P + P_{pub}) = e_{S_i} g^{sk'}$$

The verification works as follows:

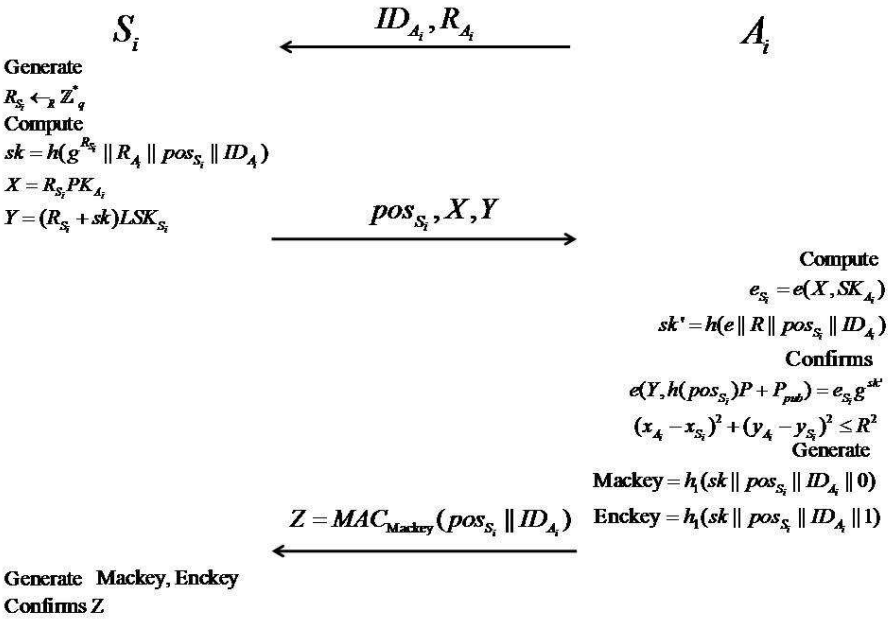


Figure 3.2: Location-based authenticated key agreement

$$\begin{aligned} e_{S_i} = e(X, PK_{A_i}) &= e(R_{A_i} h(ID_{A_i})P + R_{A_i} P_{pub}, (h(ID_{A_i}) + \kappa)^{-1} P) \\ &= e(R_{A_i}(h(ID_{A_i})P + \kappa P), (h(ID_{A_i}) + \kappa)^{-1} P) \\ &= e(R_{A_i}(h(ID_{A_i}) + \kappa)P, (h(ID_{A_i}) + \kappa)^{-1} P) \\ &= e(R_{A_i} P, P)^{h(ID_{A_i}+\kappa)h(ID_{A_i}+\kappa)^{-1}} \\ &= e(R_{A_i} P, P) = e(P, P)^{R_{A_i}} = g^{R_{S_i}} \end{aligned}$$

and

$$\begin{aligned} e(Y, h(pos_{S_i})P + P_{pub}) &= e((R_{S_i} + sk) LSK_{S_i}, (h(pos_{S_i}) + \kappa)P) \\ &= e((R_{S_i} + sk)(h(pos_{S_i}) + \kappa)^{-1} P, (h(pos_{S_i}) + \kappa)P) \\ &= e((R_{S_i} + sk)P, P)^{(h(pos_{S_i})+\kappa)^{-1}(h(pos_{S_i})+\kappa)} \\ &= e((R_{S_i} + sk)P, P) = e(P, P)^{R_{S_i}+sk} \\ &= g^{R_{S_i}+sk} = g^{R_{S_i}} g^{sk} = e_{S_i} g^{sk} = e_{S_i} g^{sk'} \end{aligned}$$

After this verification, $A_i$ also verifies that $S_i$ is really within the transmission range of $S_i$. That is, $A_i$ checks that $S_i$ is real neighbor node. $A_i$ first finds its position values $x_{A_i}$ and $y_{A_i}$ from $pos_{S_i}$ and then checks the following equation is valid.

$$(x_{A_i} - x_{S_i})^2 + (y_{A_i} - y_{S_i})^2 \leqq R^2 \text{ where R is transmission range of } S_i$$

If all the processes of verification are successful, $A_i$ believes that $S_i$ is valid and then computes two session keys, Mackey and Enckey as Mackey $= h_1(sk||pos_{S_i}||ID_{A_i}||0)$ and Enckey $= h_1(sk||pos_{S_i}||ID_{A_i}||1)$. $A_i$ also computes a message authentication code $Z$ as $MAC_{Mackey}(pos_{S_i}||ID_{A_i})$ where MAC is a message authentication code function and then sends $Z$ to $S_i$. After receiving $Z$, $S_i$ first computes Enckey and Mackey and then checks $Z$ is valid. If $Z$ is valid, the overall process of the scheme succeeds, and $A_i$ and $S_i$ share two keys, Mackey and Enckey. Otherwise, it fails.

## 3.3 Analysis

### 3.3.1 Security Analysis

**Security of IBK:** The proposed scheme utilizes IBK to distribute the LBK. For a sensor $S_i$, because we assume that the DLP is intractable in $\mathbb{G}_1$, an adversary cannot obtain $LBK_{S_i} = h_2(\kappa h(ID_{S_i}))$ without the secret value $\kappa$. After the deployment of sensors, all nodes including actor and sensors do not have $\kappa$, so the adversary cannot obtain $LBK_{S_i}$. So, we can say that the generation of IBK is secure until the adversary knows $\kappa$.

**Security of SK and LSK:** In the proposed scheme, the security of SK and LSK is based on mICDH problem. To obtain SK of an actor or LSK of a sensor without $\kappa$, the adversary can solve mICDH problem and we believe that there is no polynomial time algorithm solving mICDH problem with non-negligible probability. We can say SK and LSK is secure until the adversary knows $\kappa$.

**Authentication:** The proposed scheme provides mutual authentication between an actor node $A_i$ and a sensor $S_i$. $A_i$ checks whether $e(Y, h(pos_{S_i})P + P_{pub}) = e_{S_i} g^{sk'}$ holds. If it holds, $A_i$ can verify that $S_i$ has the knowledge of $sk'$ and $LSK_{S_i}$ and then believes that $S_i$ is valid. $S_i$ also can verify that $A_i$ has the knowledge of $sk$ and $SK_{A_i}$ by checking $Z = MAC_{Mackey}(pos_{S_i}||ID_{A_i})$. If it holds, $S_i$ believes that

$A_i$ is valid.

**Security of session key:** In the proposed scheme, the security of session keys is based on the intractability of the mBIDH problem. By eavesdropping, an adversary can obtain $h(ID_{A_i})$, $P$, $P_{pub} = \kappa P$ and $R_{S_i}(h(ID_{A_i}) + \kappa)P$. But, the attacker cannot compute $e_{S_i} = g^r = e(P,P)^{(\kappa+h(ID_{A_i}))^{-1}R_{S_i}(h(ID_{A_i})+\kappa)}$ and $sk = sk'$ because she do not know the secret value $\kappa$ and there is no polynomial time algorithm solving mBIDH problem with non-negligible probability [10].

**Resilient to DoS attacks:** In the following, we demonstrate how the proposed scheme can act as countermeasures against some most famous attacks which identified in [16, 24].

- **Sybil attack:** In the proposed scheme, sensors utilize their LBK which contains their location information. To perform sybil attack in the proposed scheme, an adversary should have to forge LBK of other nodes or compromise valid nodes. Because to forge LBK is depend on mBIDH problem, the adversary cannot impersonate other nodes. In addition, when the adversary compromise a valid node, she can utilize information of the compromised node in only the transmission range of the compromised node because the proposed scheme checks whether a sensor exists in its transmission range.

- **Identity replication attack:**

  As mentioned above, because LBK contains geographical information of sensors, actors can confirm whether a sensor really exist in the transmission range. If an adversary cannot forge LBK, we can reduce the effect of this attack. That is, the attack can be performed in the transmission range of the compromised node.

- **Wormhole attack:**

  In the proposed scheme, LBK which contains the geographical information of sensors is utilized, so an adversary can perform this attack only if she can forge LBK which depends on mBIDH problem.

### 3.3.2 Performance Analysis

In the performance, we evaluate the proposed scheme in terms of energy consumption compared with Yu *et al.*'s scheme [37]. The comparison is two-folds, one is communication cost and two is computation cost. Note out that we do not consider the energy consumption of actors because usually, actors have enough energy and the energy consumption of actors is mostly due to movement of actors [1].

In the communication cost, the proposed scheme is superior to [37]. We assume identity of entity and random nonce are 64 bit. In the proposed scheme, a sensor only needs 256 bit for receiving and 416 bit for sending. In [37], a sensor receives 280 byte for receiving and 280 byte. We utilize the result which is that Mica2 sensor consume 59.2 $\mu$J to send 1 byte and 28.6 $\mu$J to receive 1 byte [32]. Then, the proposed scheme only consumes 915.2 $\mu$J for receiving and 3078.4 $\mu$J for sending, but [37] consumes 8008 $\mu$J for receiving and 16576 $\mu$J. Totaly, the proposed scheme consumes 3993.6 $\mu$J (= 4 mJ) and [37] consumes 24584 $\mu$J (25 mJ).

For computation cost, in the proposed scheme, a sensor computes two scalar multiplications of a random point and two scalar multiplications of a fixed point. In [37], sensor computes two public key operations of RSA. The proposed scheme consumes 114.24 mJ and [37] consumes 24 mJ, respectively [25].

Then, the total energy consumption is that the proposed scheme consumes about 118 mJ and [37] consumes 49 mJ. The proposed scheme shows 2.4 times bigger energy consumption. But, note that the proposed scheme shows higher security than [37]. The proposed scheme shows resilience to both DoS attacks and node compromise. In [37], a sensor authenticates itself using not public key based cryptosystem but hash tree based approach. If the adversary compromises a certain number of sensors, she can pretend sensors which is not compromised. So, even if the proposed scheme has bigger overhead in energy consumption. We can say that it is reasonable because it has higher security level.

The proposed scheme satisfies our design goal in which sensors have light overheads. In the proposed scheme, the sensors do not perform pairing operation which is several times more costly than a scalar multiplication. Instead, the sensors need four scalar multiplication computations and one modular exponentiation computation for each AKA. Compared with other location-based scheme [38, 39] which each entity should compute one pairing operation, in the proposed scheme, only actors perform pairing operation and sensors need not compute it.

In the proposed scheme, actors should compute one Weil or Tate pairing. Because the actors have enough resources in terms of storage and battery, the only computation time

Table 3.1: Comparison summary

| | Criterion 1 | Criterion 2 | Criterion 3 | Criterion 4 | Criterion 5 |
|---|---|---|---|---|---|
| [CHCC05] | O | △ | X | X | low |
| [YMWMG06] | O | O | △ | X | medium |
| Our scheme | O | O | O | O | high |

of the pairing is an issue. In the recent implementation [29], the computation of pairing in a sensor only takes 1.93 sec. This result shows the feasibility to utilize pairing operation in actors.

### 3.3.3 Comparison summary

Here, we compare the proposed scheme with [CHCC05] and [YMWMG06]. For the comparison, we select five criterions as follows: mutual authentication (denoted by Criterion 1), key establishment (denoted by Criterion 2), resilience to node compromise (denoted by Criterion 3), resilience to DoS attack (denoted by Criterion 4). For the performance comparison, we make a criterion, energy consumption of sensors (denoted by Criterion 5). In Table 1, $O$ means satisfying the criterion perfectly, △ means satisfying the criterion in part, and $X$ means unsatisfying the criterion.

Table 1 shows the summary of comparison. Detail description is as follows: For Criterion 1, all the three schemes provide mutual authentication between actors and sensors. In Criterion 2, our scheme and [YMWMG06] provides the key agreement operation, where two entities contribute to establishment of a shared key, but, [CHHCC05] only provides the key transport operation, where an actor generates a key by itself and just transports the key to a sensor. Generally, we consider that the key agreement is stronger requirement than the key transport. So, we give △ to [CHCC05] for Criterion 2. For Criterion 3, only our scheme satisfies it perfectly. In [CHCC05], if a node is compromised, an adversary can impersonate other non-compromised sensors or actors. [YMWMG06] also shows an weakness to node compromise. When an adversary compromises with a sufficient number of nodes, she can take the control of an entire network. However, in our scheme, even if an adversary compromises a sufficient number of nodes, she cannot affect non-compromised nodes. Criterion 4 is one of motivations of our scheme. Only our scheme satisfies it. For Criterion 5, [CHCC05] utilizes lightweight primitives like symmetric encryption, MAC, and hash function. On the other hand, [YMWMG06] and our scheme utilize public key based cryptosystem. Especially, our scheme utilizes ID-based cryptosys-

tem, which requires the high computation cost. As we already said, even though we try to assign light overhead to sensors, the energy consumption of sensors in our scheme is twice than that of [YMWMG06].

In conclusion, our scheme shows stronger security than [CHCC05] and [YMWMG06], but requires the bigger overhead in the performance. Especially, for some applications which need the high level of security (*e.g., battle filed monitoring*), we argue that our scheme is more competitive solution than others.

# 4. Conclusion

Wireless Sensor and Actor Network (WSAN), which is the integrated network of sensors and actors, has been appeared. WSAN refers to a group of sensors and actors linked by wireless medium to perform distributed sensing and acting tasks. In WSAN, sensors gather information about the physical world, while actors take decisions and then perform appropriate actions upon the environment. The coexistence of sensors and actors introduces new challenges to realize WSAN. In order to secure WSAN, the new challenge is to provide the security mechanisms between actors and sensors because this problem is not addressed by previous work.

In this thesis, we focus on providing security mechanisms between actors and sensors. Specifically, we propose location-based Authenticated Key Agreement (AKA) scheme, operated over ID-based cryptosystem. The proposed scheme provides authentication and key establishment between actors and sensors and further shows the resilience to Denial of Service (DoS) attack for routing layer by utilizing the location information of sensors to generate location-based public and private key pair of sensors. We also analyze the security and performance of the proposed scheme. The proposed scheme shows the superior security than other schemes for WASN. We try to assign light overhead to sensors to make sensors compute no pairing operation. The consumption of the energy of sensors is about twice than that of [YMWMG06]. But, the proposed scheme shows the higher security than [YMWMG06], especially for the resilience to node compromise and DoS attack, so we argue that it is reasonable.

We leave the followings as the future work. In this thesis, we only consider AKA between actor and sensors, but location-based AKA between sensors can be considered. Also, it is possible to reduce the assumption where actors have the secret of TA during deployment of sensors. If this assumption is removed, the proposed scheme can be more secure and flexible.

# 요 약 문

## 무선 센서 및 액터 네트워크에서 위치 기반의 인증된 키합의 기법에 관한 연구

무선 센서 및 액터 네트워크는 무선 센서 네트워크의 일종으로 무선 센서 네트워크에 이동이 가능한 액터를 노드를 추가하여, 어떤 사건의 탐지뿐 아니라 실시간 대응이 가능하도록 고안된 네트워크이다. 무선 센서 및 액터 네트워크는 사회 주요 기반 인프라로 산불 감시, 전장 감시, 환경 오염 감시 등 여러 응용 프로그램에 활용될 수 있다.

본 학위 논문에서는 처음으로 서비스 거부 공격에 견고한 안전한 키 합의 및 인증 기법을 제안한다. 제안 기법은 센서의 위치정보를 활용하여 센서의 신원 기반 키를 생성하고 액터와의 인증 및 키 합의를 하는데 활용함으로써 여러 서비스 거부 공격으로부터 견고성을 가진다. 또한, 센서 노드가 오버헤드가 큰 페어링 연산을 수행하지 않도록 설계되었다. 기존의 기법과 비교해서 약 2배 정도의 에너지 소모가 늘었지만, 노드 탈취와 서비스 거부 공격에 견고성을 가지는 등 향상된 보안을 제공할 수 있기 때문에 의미있다고 할 수 있다.

# References


[1] I. F. Akyildiz and I. H. Kasimoglu, "Wireless sensor and actor networks: research challenges", Ad Hoc Networks, Vol. 2, No. 4, pp. 351-367, 2004.

[2] D. Balenson, D. A. McGrew, and A. T. Sherman. "Key management for large dynamic groups: One-way function trees and amortized initializtion," Internet Draft, Internet Engineering Task Force, March 1999. Work in progress.

[3] P. Barreto, H. Kim, B. Bynn and M. Scott, "Efficient algorithms for pairing-based cryptosystems" CRYPTO '02, LNCS 2442, pp. 354-368, 2002.

[4] B. Bloom, "Space/time trade-offs in hash coding with allowable errors," Communications of the ACM, Vol. 13, No. 7, 1970.

[5] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing", SIAM Journal of Computing, Vol. 32, No. 3, pp. 586-615, 2003.

[6] Muhammad Bohio and Ali Miri,"Efficient identity-based security schemes for ad hoc networks routing protocols", Ad Hoc Networks, Vol. 2, pp. 309-317, 2004.

[7] X. Cao, M. Huang, Y. Chen, and G. Chen, "Hybrid Authentication and Key management Scheme for WASNs", ISPA Workshops 2005, LNCS 3759, pp. 454-465, 2005.

[8] H. Chan, A. Perrig, and D. Song, "Random KeyPredistribution Schemes for Sensor Networks." in IEEE Symposium on Security and Privacy, pp. 42-51, May 2003.

[9] Hung-Yu Chien and Ru-Yu Lin, "Improved id-based security framework for ad hoc network", Ad Hoc Networks, Vol. 6, pp. 46-60, 2008.

[10] K. Choi, J. Hwang, D. Lee and I. Seo, "ID-based authenticated key agreement for low-power mobile devices", ACISP 2005, LNCS 3574, pp. 494-505, 2005.

[11] P. Corke, R. Peterson and D. rus, "Networked robots: flying robot navigation using a sensor net", ISRR'03, 2003.

[12] 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(CCS), pp. 42-51, October 27-31 2003.

[13] L. Eschenauer and V. D. Gligor, "A key management scheme for distributed sensor networks." in Procedings of the 9th ACM conference on Computer and Communications Security(CCS), pp. 41-47, November 2002.

[14] W. Heinzelman, A. 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, October 2002.

[15] J. Jung, T. Park and C. Kim, "A Forwarding Scheme for Reliable and Energy-efficient Data Delivery in Cluster-based Sensor Networks," IEEE Communication Letters , Vol. 9, No. 2, pp. 112-114, Feb 2005.

[16] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures", Ad Hoc Networks, Vol. 1, Issues 2-3,

[17] Y. Kim, H. Lee, J. Park and L. T. Yang, "Key establishemnt scheme for sensor networks with low communication cost", ATC 2007, LNCS 4610, pp. 441-448, 2007.

[18] D. Liu and P. Ning, "Establishing Pairwise Keys in Distributed Sensor Networks", Proc. of the 10th ACM Conference on Computer and Communication Security(CCS), 2003.

[19] D. Liu and P. Ning, "Location-Based Pairwise Key Establishments for Static Sensor Networks", Proc. of the 1st ACM Workshop on the Security of Ad Hoc and Sensor Networks(SASN), 2003.

[20] D. Liu, "Efficient and Distributed Access Control for Sensor Networks", Distributed Computing in Sensor Systems, LNCS 4549, pp. 21-35, 2007.

[21] D. A. McGrew and A. T. Sherman, "Key establishment in large dynamic groups using one-way function trees", May 1998. http://www.cs.umbc.edu/ sherman/Papers/itse.ps.

[22] T. Melodia, D. Pompili, I. F. Akyildiz, "A Communication Architecture for Mobile Wireless Sensor and Actor Networks", IEEE SECON 2006, pp. 109-118, 2006.

[23] T. Melodia, D. Pompili, V. C. Gungor, and I. F. Akyildiz, "Communication and Coordination in Wireless Sensor and Actor Networks", IEEE Transactions on Mobile Computing, Vol. 6, No. 10, pp. 1116-1129, October 2007.

[24] J. Newsome, E. Shi, D. Song and A. Perrig, "The sybil attack in sensor networks: Analysis & defenses", IPSN 2004, April 2004.

[25] K. Piotrowsk, P. Lagnedoerfer, S. Peter, "How public key cryptography influences wireless sensor node lifetime", Proceedings of the 4th ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN 2006), pp. 169.176, 2006

[26] K. Ren, W. Lou, and Y. Zhang, "LEDS: Providing Location-Aware End-to-End Data Security in Wireless Sensor Networks", IEEE Trasactions on Mobile Computing, Vol. 7, No. 5, pp. 585-598, May 2008.

[27] K. Ren, W. Lou, B. Zhu, and S. Jajodia "Secure and Efficient Multicast in Wireless Sensor Networks Allowing Ad hoc Group Formation", IEEE Transaction on Vehiclular Technology (TVT), Vol. 58, No. 4, May 2009.

[28] A. Sanfeliu and J. Andrade-Cetto, "Ubiquitous networking robotics in urban settings", Workshop on Netwrok Robot Systems, 2006.

[29] M. Shirase, Y. Miyazaki, T. Takagi, D. Han and D. Choi, "Efficient implementation of pairing based cryptography on a sensor", IEICE Trans. on Information and Systems, Vol. E92-D, No. 5, pp. 909-917, 2009.

[30] H. Song, S. Zhu, W. Zhang, and G. Cao, "Least Privilege and Privilege Deprivation: Toward Tolerating Mobile Sink Compromises in Wireless Sensor Networks," ACM Transactions on Sensor Networks (TOSN), Vol. 4, No. 4, Article 23, August 2008.

[31] R. Vedanthan, Z. Zhuang and R. Sivakumar, "Mutual exclusion in wireless sensor and actor networks", IEEE SECON 2006, pp. 346-355, 2006.

[32] A. Wander, N. Gura, H. Eberle, V. Gupta, and Sh. Shantz, "Energy Analysis of Public-Key Cryptography on Small Wireless Devices", IEEE PerCom, 2005.

[33] H. Wang and Q. Li. "Distributed User Access Control in Sensor Networks", Distributed Computing in Sensor Systems, LNCS 4026, pp. 305-320, 2006.

[34] J. Wu, S. Yang and M. Cardei, "On maintaining sensor-actor connectivity in wireless sensor and actor networks", IEEE INFCOM 2008, pp. 888-896, 2008.

[35] H. Yang, F. Ye, Y. Yuan, S. Lu, and W. Arbaugh, "Toward Resilient Security in Wireless Sensor Networks", ACM MOBIHOC 2005, 2005.

[36] F. Ye, H. Luo, S. Lu, and L. Zhang, "Statiscal Enroute Filtering of Injected False Data in Sensor Networks", IEEE infocom, 2004.

[37] B. Yu, J. Ma, Z. Wang, D. Mao and C. Gao, "Key Establishemtn Between Heterogenous Nodes in Wireless Sensor and Actor Networks", IWSN 2006, LNCS 3842, pp. 196-205, 2006.

[38] Y. Zhang, W. Liu, W. Lou and Y. Fang, "Securing sensor networks with location-based keys", IEEE WCNC 2005, pp. 1909-1914, 2005.

[39] Y. Zhang, W. Liu, W. Lou and Y. Fang, "Location-based compromise-tolerant security mechanisms for wireless sensor networks", IEEE JSAC, Vol. 24, No. 2, pp. 1-14, 2006.

[40] L. Zhou and C. V. Ravishankar, "Dynamic Merkle Trees for Verifying Privileges in Sensor Networks", IEEE ICC'06, 2006.

[41] Y. Zhou, Y. Zhang, and Y. Fang, "Access control in wireless sensor networks", Ad Hoc Networks, Vol. 5, No. 1, pp. 3-13, 2007.

# 감 사 의 글

이 논문을 완성하기까지 주위의 모든 분들로부터 수많은 도움을 받았습니다. 김광조 교수님께서는 틈틈히 연구 상황을 확인해 주셔서 체계적인 연구방향을 세울 수 있었습니다. 게다가 연구실 선후배와 동기들에게 연구실 생활 및 연구 주제 등 많은 부분에서 격려 및 조언을 받았습니다. 또한, 바쁘신 와중에도 학위논문심사를 위해 참석하셔서 진심어린 조언을 주신 김명철 교수님, 이병천 교수님께 감사드립니다.

끝으로 오늘의 제가 있을 수 있도록 사랑으로 키워 주신 어머니, 무뚝뚝하지만 항상 그 자리에 계셔주신 아버지에게 감사드립니다. 저의 이 작은 결실이 그분들께 조금이나마 보답이 되기를 바랍니다.

# Curriculum Vitae


Name : Imsung Choi

Date of Birth : February 18, 1985

Address : R504, 109 Munjiro, Yusong-gu, Daejeon, 305-714 KOREA

E-mail : TheShaki@kaist.ac.kr


## Educations

2004. 3. – 2008. 2. Computer Engineering, Information and Communication University (ICU) (B.S.)

2008. 2. – 2010. 2. Information and Communication Engineering, KAIST (M.S.)

## Career

2009. 06. – 2009. 08. Sony Internship Program, Japan.

## Publications

1. 신승목, **최임성**, 김광조, "무선 센서 네트워크에서의 안전한 네트어크 재프로그래밍 기법", CISC-S'08 Proceedings vol. 18, no.1, pp. 162-167, 2008.06.26, 순천향대학교 천안.
2. **최임성**, 김진, 김광조, "무선 센서 네트워크에서의 안전한 클러스터링 프로토콜들의 안전성 분석", 2008년도 한국정보보호학회 충청지부 학술발표회 논문집, pp. 85-91, 2008.1017, 배재대학교, 대전.
3. **최임성**, 현록, 김광조, "중앙서버가 없는 추적할 수 없는 RFID 인증 및 프로토콜들", 2009년도 한국정보보호학회 충청지부 학술발표회 논문집, pp. 91-105, 2009.10.23, 호서대학교, 아산.
4. **Imsung Choi**, Zeen Kim, and Kwangjo Kim, "DoS-Resilient Authenticated Key Agreement Scheme between Actor and sensor nodes in Wireless Sensor and Actor Network", Joint Workshop on Information Security 2009, Aug. 6-7, 2009, Kaohsiung, Taiwan.

5. Junhyung Yim, **Imsung Choi**, and Kwangjo Kim, “An Efficient Anonymous Authentication Protocol in Vehicular Ad-hoc Networks”, The 10th International Workshop on Information Security Applications, Aug. 25-27, 2009, Busan, Korea.
6. **최임성**, 김진, 김광조, “무선 센서 네트워크에서 안전하고 효율적인 방송형 인증 기법 연구”, CISC-W’09 Proceedings, pp. 206-213, 2009. 12. 5. 연세대학교, 서울 [우수논문]