Practical Attacks on Hybrid Group Key Management for SOHAN

Jiun-Hau Liew, Ivy Ong, Sanggon Lee, HyoTaek Lim and HoonJae Lee*, *Member, KIMICS*

***Abstract*— Lim et al. proposed a Hybrid Group Key Management scheme for Hierarchical Self-Organizing Sensor Network in 2008 to provide a secure way to pass down the group key for cluster-based communication. This paper presents two practical attacks on the scheme proposed by Lim et al. by tampering sensor nodes of a cluster to recover necessary secret keys and by exploiting the IDS employed by the scheme. The first attack enables a long-term but slow data fabrication while other attack causes more severe DoS on the access to cluster sensor nodes.**

***Index Terms*— Data Fabrication, Denial of Service, Hybrid Group Key Management, Wireless Sensor Network.**

I. INTRODUCTION

WIRELESS Sensor Network (WSN) is widely deployed nowadays to monitor physical or environmental conditions in various applications ranged from civilian to military purposes. A wide coverage area WSN is normally formed by many base stations with a large number of self-configuring and self-organizing sensor nodes, which are small and have limited resources such as battery, power, computations and memory spaces constraints [1]. Therefore, key management handling in WSN is differed from those conventional wired or wireless networks. The design of WSN key management schemes relies on some crucial considerations such as reliable key distribution procedures, energy consumption, scalability and tamper resistant properties, as well as computation and storage overhead trade-off among participated entities.

To date, there are many efforts made by researchers in establishing group-wise keys for sensor nodes to communicate securely. In 2004, Park and Shin have employed the symmetric cryptography approach and presented the Lightweight Security Protocol (LiSP) [2] that offers key broadcast, lost keys detection/recovery and seamless key refreshment services through the re-keying mechanism. In 2005, Burmerster et al. have extended the ideas from asymmetric Diffie-Hellman protocol and presented a novel scalable Group Key Exchange protocol [3] that is able to defend passive adversary attacks. Also, Carman et al. have proposed an energy efficient and low latency asymmetric key management approach, ID-STAR [4], which is built based on identity-based public key cryptography to fulfill the requirements of greater ranges, low probability of interception and anti-jamming in army sensor network.

In addition, Zhu et al. have introduced a symmetric key management protocol LEAP [5] for large scale sensor network. Instead of using a single keying mechanism, this protocol supports four diverse types of key establishments per sensor node: individual, pair-wise, cluster and group shared key. LEAP has achieved greater energy efficiency and is protected from various attacks, yet it does not sufficiently address the scalability problem in group key distribution and management. To enhance the scalability property, Lim et al. have presented a secure hybrid group key management (HGKM) [6] for hierarchical self-organizing sensor network in 2008.

In this paper, we show that HGKM is vulnerable to DoS attack and data fabrication by exploiting the features of IDS system [7] used in the scheme. The structure of this paper is organized as follows. We will first review the HGKM scheme in Section II. We then show how we could compromise the scheme in Section III. Finally we conclude by giving a summary.

II. LIM ET AL.'S SCHEME

HGKM is designed to operate on a sensor network that is based on a Hierarchical Self-Organizing Ad-hoc Network (SOHAN) [8]. In a SOHAN-based sensor network, there are three main components arranged in hierarchical architecture: Access Points (AP), Forwarding Nodes (FN) and Sensor Nodes (SN). The uppermost AP layer roles as a bridge between a wireless and wired environment, whereas the intermediate FN layer provides a wireless radio interface with the responsibility of routing sensed data from the lowest SN layer to the AP layer. Fig. 1 shows the model of SOHAN-based sensor network.

Manuscript received July 19, 2010; revised August 7, 2010; accepted August 11, 2010.

Jiun-Hau Liew is with the Graduate School Design and IT of Dongseo University, Busan, 617-716, Korea (Email: liew.jiun.hau@gmail.com)

* Corresponding Author

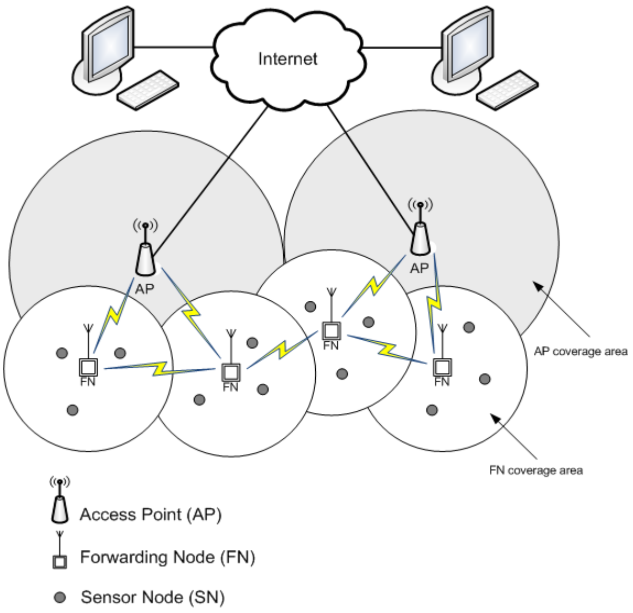

Fig. 1. SOHAN-based Sensor Network

In the proposed HGKM scheme by Lim et al., the group key between AP and FN is computed based on the Diffie-Hellman key agreement protocol. The group key is subsequently transported by FN to SNs that are being deployed to the clustered group. This scheme utilizes the higher power characteristic of APs and FNs to perform cryptographic computations without burdening the low power SNs that have limited computing capability and energy constraint. There are 3 phases in HGKM: Group Key Agreement (GKA), Group Key Transport (GKT) and Group Key Refresh (GKR) Phase.

A. Notations

ID_a	-	Identity of *a*
SC	-	Session Counter
K_{ab}	-	Shared secret key *K* between *a* and *b*
g	-	Primitive root in Diffie-Hellman
N_a	-	Nonce generated by *a*
CA	-	Certificate Authority
y_a/x_a	-	Long-term Public/Private key pair of *a* with a pairing certificate signed by *CA*
t_a/r_a	-	Ephemeral Public/Private key pair of *a*
$m_1 \| m_2$	-	Concatenation of m_1 and m_2
$MAC_K(m)$	-	Message Authentication Code of *m* using *K*
$E_K(m)$	-	Asymmetric encryption of *m* using *K*
$D_K(m)$	-	Asymmetric decryption of *m* using *K*
$\{m\}_K$	-	Symmetric encryption of *m* using *K*
$D\{m\}_K$	-	Symmetric decryption of *m* using *K*

B. Assumptions

- AP is always honest.
- FN or SN is not tamper resistant.
- Initial secret key K_S pre-installed in all SN and AP
- An intrusion detection system (IDS) is present
- An intruder can perform eavesdropping, fabrication, interception and modification attacks

C. Group Key Agreement Phase

GKA is executed prior to the deployment of the SNs. Whenever the group key requires an update, AP and FN will also perform GKA.

1) Firstly, AP and FN compute the initial shared key, $K_{AF} = y_{FN}{}^{x_{AP}} = y_{AP}{}^{x_{FN}} = g^{x_{AP}x_{FN}}$ respectively. After that, AP picks an ephemeral $t_{AP} = g^{r_{AP}}$ while FN computes $t_{FN} = g^{r_{FN}}$ for the current session.

$$\text{FN} \rightarrow \text{AP}:$$
$$ID_{FN}, t_{FN}, MAC_{K_{AF}}(ID_{FN} \| ID_{AP} \| t_{FN})$$

2) AP verifies $MAC_{K_{AF}}$. If succeeds, it selects an initial *SC* value and computes $\{SC\}_{K_{AF}}$ and $K_G = t_{FN}{}^{r_{AP}}$.

$$\text{AP} \rightarrow \text{FN}:$$
$$ID_{AP}, t_{AP}, \{SC\}_{K_{AF}},$$
$$MAC_{K_S}(ID_{AP} \| ID_{FN} \| K_G \| SC),$$
$$MAC_{K_{AF}}(ID_{AP} \| ID_{FN} \| t_{FN}{}^{x_{AP}} \| t_{AP} \| \{SC\}_{K_{AF}})$$

3) FN verifies $MAC_{K_{AF}}$ by substituting the computation of $t_{FN}{}^{x_{AP}} = y_{AP}{}^{r_{FN}}$ and generates a new MAC. If matches, FN decrypt $D\{\{SC\}_{K_{AF}}\}_{K_{AF}}$ and computes $K_G = t_{AP}{}^{r_{FN}}$.

$$\text{FN} \rightarrow \text{AP}:$$
$$ID_{FN}, MAC_{(K_{AF} \| K_G \| SC)}(ID_{FN} \| ID_{AP} \| t_{AP}{}^{x_{FN}} \|$$
$$t_{FN} \| MAC_{K_S}(ID_{AP} \| ID_{FN} \| K_G \| SC))$$

4) AP verifies $MAC_{(K_{AF} \| K_G \| SC)}$ by substitute the computation of $t_{AP}{}^{x_{FN}} = y_{FN}{}^{r_{AP}}$ and generates a new MAC. If matches, finally the GKA process is completed successful with the agreed group key to be used as follows:

$$K_G = t_{AP}{}^{r_{FN}} = t_{FN}{}^{r_{AP}} = g^{r_{FN}r_{AP}}$$

D. Group Key Transport Phase

Right after the SNs has been deployed, GKT will be request by each SN in order to fetch the group key from the FN which the SN wish to connect with. GKT is also perform when SN want to switch to a different FN for any other reasons.

1) Similar to GKA, SN and FN compute the initial shared key, K_{FS} and SN generates N_{SN} for the session and send initial group key request to FN:

$$\mathrm{SN} \rightarrow \mathrm{FN}:$$
$$ID_{SN}, E_{y_{FN}}(N_{SN}), MAC_{K_{FS}}(ID_{SN} \parallel ID_{FN} \parallel N_{SN})$$

2) FN decrypt $E_{y_{FN}}(N_{SN})$ verifies $MAC_{K_{FS}}$. If succeeds, FN replies:

$$\mathrm{FN} \rightarrow \mathrm{SN}:$$
$$ID_{FN}, E_{y_{SN}}(K_G \parallel SC), N_{FN}, ID_{AP},$$
$$MAC_{K_S}(ID_{AP} \parallel ID_{FN} \parallel K_G \parallel SC),$$
$$MAC_{K_{FS}}(ID_{FN} \parallel ID_{SN} \parallel N_{FN} \parallel N_{SN} \parallel$$
$$E_{y_{SN}}(K_G \parallel SC))$$

3) SN verifies $MAC_{K_{FS}}$ and performs decryption to recover $K_G \parallel SC$. After that, if verifies MAC_{K_S}. If verification passed, it sends acknowledgement to FN:

$$\mathrm{SN} \rightarrow \mathrm{FN}:$$
$$ID_{SN},$$
$$MAC_{(K_{FS} \parallel K_G \parallel SC)}(ID_{SN} \parallel ID_{FN} \parallel N_{SN} \parallel N_{FN})$$

4) Finally, FN verifies $MAC_{(K_{FS} \parallel K_G \parallel SC)}$ and confirmed that SN has received the correct key.

E. Group Key Refresh Phase

To reduce the risk of group key compromise, AP will first update the group key to FN via GKA. After that, FN will propagate the update to the SNs.

1) FN broadcast the new group key to its SNs by securing the new group key with current group key.

$$\mathrm{FN} \rightarrow \mathrm{SN}:$$
$$ID_{FN}, \{K'_G \parallel SC'\}_{(K_{FS} \parallel K_G \parallel SC)}, N_{FN}, ID_{AP},$$
$$MAC_{K_S}(ID_{AP} \parallel ID_{FN} \parallel K'_G \parallel SC'),$$
$$MAC_{(K_{FS} \parallel K_G \parallel SC)}(ID_{FN} \parallel ID_{SN} \parallel$$
$$\{K'_G \parallel SC'\}_{(K_{FS} \parallel K_G \parallel SC)})$$

2) SN performs decryption to recover $K'_G \parallel SC'$ and compute if $SC = SC + 1$. Then it verifies the MACs. If matches, the new group key is updated in SN and acknowledgement is sent to FN.

$$\mathrm{SN} \rightarrow \mathrm{FN}:$$
$$ID_{SN},$$
$$MAC_{(K_{FS} \parallel K'_G \parallel SC')}(ID_{SN} \parallel ID_{FN} \parallel N_{FN})$$

3) Finally, FN verifies $MAC_{(K_{FS} \parallel K'_G \parallel SC')}$ and confirmed that SN has received the correct key.

III. ATTACKS ON HGKM

Each AP, FN and SN is pre-assigned with *y*/*x* key pairs and signed by a trusted CA. However, FN and SN can be easily compromised physically by an intruder to obtain the underlying sensitive secret keys. HKGM depends on LEACH-based [7] or other similar IDS to detect and isolate the compromised nodes, as well as to trigger the GKR process to generate new group key. LEACH divides *n* nodes in a cluster into *n*/*m* groups with *m* member nodes (MN). LEACH then detects data packet jamming, dropping and duplicating in the network by using MN to ”overhear” in ”prominous” mode. In case of SOHAN, since there is no inter-cluster communication for SNs, after SNs detected a malicious FN in their cluster, those SNs will need to reconnect to another trusted FN before they can report to the IDS to blacklist the compromised FN. This allows a window of opportunity for attack.

A. Data Fabrication

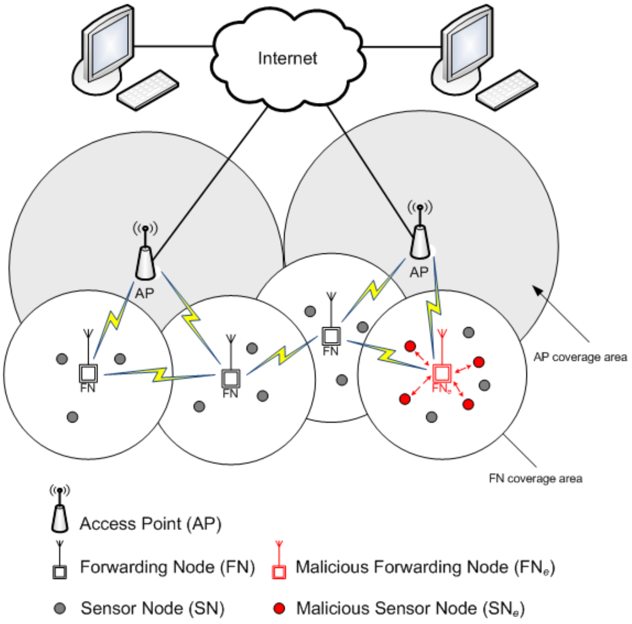

Fig. 2. Data Fabrication with Malicious SNs

In HGKM, the authors made an implicit naive assumption that GKT must be followed on all FNs, even the compromised one. However, in our case, we assumed custom-coded FN$_e$s are used. This allows GKT to be skipped when interacting with unknown SNs which do not have valid certificates signed by CA.

The first step of our attack is to compromise some FNs and recover y_{FN} /x_{FN} and the current K_G || SC. These keys are then loaded into FN$_e$ so that FN$_e$s can take over the compromised FN and continue to work as normal. This will allow the FN$_e$ to avoid detection by the IDS. Since FN$_e$ have the necessary keys, it can continue to receive K'_G updates from AP. Since FN$_e$ does not follow standard GKT in HKGM, we are able to deploy malicious SN$_e$s to connect to FN$_e$ as depicted in Fig. 2.

When FN$_e$ detected the existence of SN$_e$, it will pass K_G and subsequent K'_G directly to SN$_e$. This bypass Diffie-Hellman exchange and there will be no ID verification performed. With the valid group keys, SN$_e$ can then inject falsified data into the WSN passively as a long-term attack.

B. Denial of Service

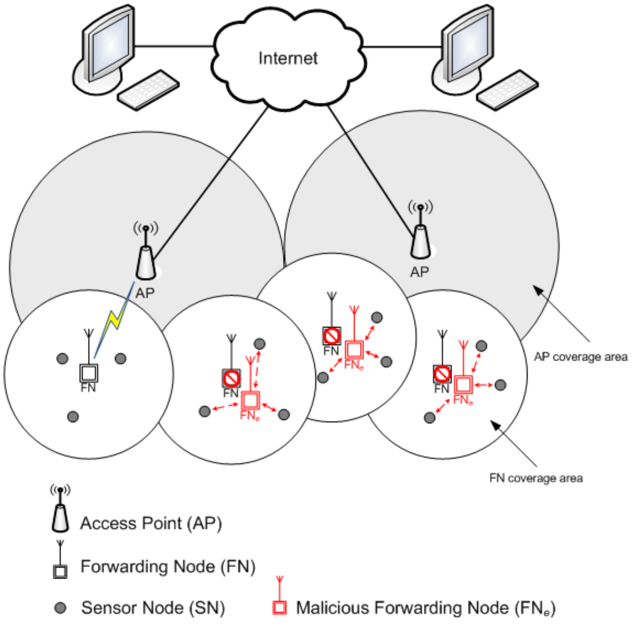

Fig. 3. Packet Duplicating Attack Causes SNs to Connect to Malicious FNs

Our first attack is purely passive and does not inflict serious damage on the WSN. In this section, we explain a DoS attack by exploiting the IDS. We only require the keys from a single FN and K_S from SN. Again, we assumed multiple custom-coded FN$_e$s are used and the same keys from a particular FN are pre-loaded into these FN$_e$s. These FN$_e$s are then deployed close to the FNs that we are going to attack. Recall that in GKA, GKT and GKR, none of the message exchanges include a time-stamp.

1) FN$_e$s will first capture a valid message from these protocols and start to broadcast the message repeatedly for awhile.

2) The nearby SNs will detect duplicated packets from the FN. Since SNs cannot differentiate these packets are actually old packet, this will trigger the intrusion detection at SNs.

3) SNs will start to look for other FN to reconnect and since FN$_e$ is placed closely to FN, there is high probability that SNs will request GKT with FN$_e$ as shown in Fig. 3.

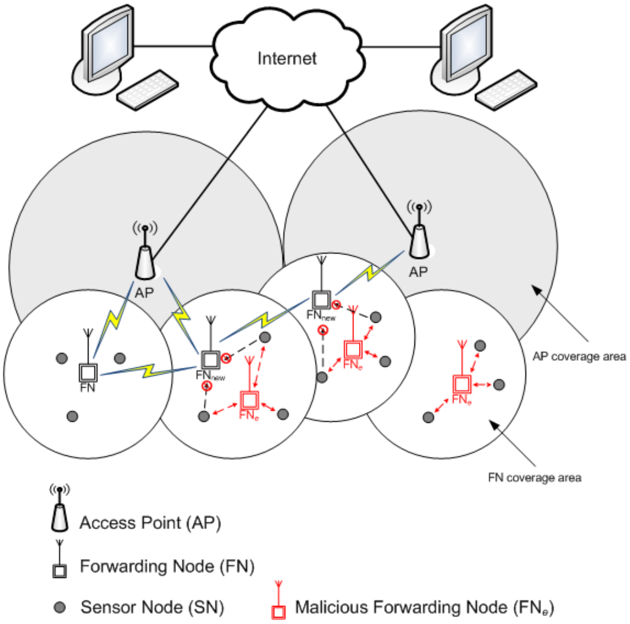

Fig. 4. SNs Remained Connected to Malicious FNs

Since FN$_e$ can forge a valid GKT session with y_{FN} /x_{FN} and K_S:

1) FN$_e$ is able to establish K_{FS} with the stolen y_{FN} /x_{FN} by masquerading as an authentic FN.

2) FN$_e$ can pick any K_G and SC and produce the valid MACs since it has the stolen K_S and K_{FS} .

FN$_e$ can also provide group key updates using GKR with the same keys:

1) FN$_e$ can encrypt K'_G || SC' by using the K_{FS} with the K_G || SC passed in the previous GKT.

2) FN$_e$ can produce the required MACs since it has K_S, K_{FS} and K_G || SC

3) K'_G does not necessary come from AP. As long as $SC' = SC+1$, SN will accept any arbitrary K'_G as a valid update.

After SNs is connected to FN$_e$, it will continue to forward data but the data will be dropped by the AP since it is using an invalid group key. This will go undetected by the SNs since SNs only monitor the FN in their cluster. Furthermore, FN$_e$ can perform valid GKR periodically and keep the SNs connected to it. After some time, new FNs maybe deployed to replace the blacklisted FNs.

However GKT is always initiated by SN, these new FNs cannot force SNs to join their cluster as shown in Figure 4. This results a DoS as the base stations can no longer get data from these SNs.

IV. CONCLUSION

We have shown that the scheme is not secure and can be practically attacked. HKGM has a dependency on IDS therefore security is not guaranteed. Furthermore, since K_S is subject to compromise, it should not be used for long-term verification. In addition, we feel that HKGM may not be very efficient in term of energy usage due to many exponentials and asymmetric encryption operations involved. On top of that, there is also a hidden energy cost of ID verification against the CA-signed certificate for each GKA/GKT/GKR process.

There is still much work to be done to improvise the scheme so that the attacks could be patch and to make the scheme more efficient by either simply the scheme or reduce the number of cryptographic computation required.

ACKNOWLEDGMENT

This work was supported in part by the 2010 National Research Foundation Project and 2009 Dongseo Frontier project.

REFERENCES

[1] J. Ibriq and I. Mahgoub, "Cluster-Based Routing in Wireless Sensor Networks: Issues and Challenges," Proc. of the 2004 Sym. on Performance Evaluation of Computer Telecommunication Systems, 2004, pp. 759-766.

[2] T. Park and K.G. Shin, "LiSP: A Lightweight Security Protocol for Wireless Sensor Networks," ACM Transactions on Embedded Computing Systems, vol. 3, no. 3, 2004, pp. 634-660.

[3] M. Burmerster and Y. Desmedt, "A Secure and Scalable Group Key Exchange System," Information Processing Letter, vol. 94, no. 3, 2005, pp. 137-143.

[4] D. Carman, B. Matt, and G. Cirincione, "Energy-efficient and Low-latency Key Management for Sensor Networks," Proc. of 23rd Army Science Conference, 2002.

[5] S. Zhu, S. Setia, S. Jajodia, "LEAP: Efficient Security Mechanisms for Large-Scale Distributed MSN Networks," Proc. of ACM conference on Computer and Communication Security, 2003, pp. 62-72.

[6] S.Y. Lim, M.H. Lim, S.G. Lee, and H.J. Lee, "Secure Hybrid Group Key Management for Hierarchical Self-Organizing Sensor Network," Proc. of the Fourth International Conference on Information Assurance and Security, 2008, pp. 43-49.

[7] C. Su, K. Chang, Y. Kuo, and M. Horng, "The New Intrusion Prevention and Detection Approaches for Clustering-Based Sensor Networks," IEEE Wireless Communications and Networking Conference, vol. 4, 2005, pp. 1927-1932.

[8] S. Ganu, L. Raju, B. Anepu, I. Seskar, and D. Raychaudhuri, "Architecture and Prototyping of an 802.11-based Self-Organizing Hierarchical Ad-Hoc Wireless Network (SOHAN)," Proc. of the 15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, vol. 2, 2004, pp. 880-884.

Jiun-Hau Liew
The author graduated with the B.S. degree in IT (Hons.) from the College of Information Technology of University Tenaga Nasional in 2007 and joined the Graduate School of Design and IT in Dongseo University in Sept 2009 as a M.S. student researching in the field of cryptography and computer network security.

Ivy Ong
Graduated from Multimedia University, Malaysia in 2005 with the B.S degree of IT (Hons.) major in Information Systems Engineering, currently she is pursuing M.S. of Engineering in Graduate School of Design and IT in Dongseo University, Korea. She research interest includes hard disk drive technology, reliability analysis, and network and database management

Sang-Gon Lee received his BEng, MEng, and PhD degree in electronics engineering from Kyungpook National University, Korea, in 1986, 1988, and 1993, respectively. He is a professor in the Division of Computer & Information Engineering, Dongseo University. He was an assistant/associate professor at Chang-shin College from 1991 to 1993 and a visiting scholar at QUT, Australia from 1993 to 1994. His research areas include information security, network security, wireless network and digital right managements.

HyoTaek Lim received his BS degree in Computer Science from Hongik University in 1988, the MS degree in Computer Science from POSTECH and the PhD degree in Computer Science from Yonsei University in 1992 and 1997, respectively. From 1988 to 1994, he had worked for Electronics and Telecommunications Research Institute as a research staff. Since 1994, he has been with Dongseo University, Korea, where he is currently a professor in the Division of Computer and Information Engineering. His research interests include computer network, protocol engineering, storage networking, IPv6 and mobile application.

HoonJae Lee received his BS, MS, and PhD Degrees in electronic engineering from Kyungpook National University, Daegu, Korea in 1985, 1987, and 1998, respectively. He is currently a professor in the School of Computer and Information Engineering at Dongseo University. From 1987 to 1998, he was a research associate at the Agency for Defense Development (ADD). His current research interests include developing secure communication system, side-channel attack and USN/RFID security.