Statistical En-Route Filtering of Fabricated Reports in Ubiquitous Sensor Networks Based on Commutative Cipher

Hae Young Lee and Tae Ho Cho

Sungkyunkwan University, Suwon 440-746, South Korea

Summary
In ubiquitous sensor networks, sensor nodes are deployed in hostile environment, and thus vulnerable to the fabricated report injection attacks in which attackers inject fabricated reports into networks through compromised nodes to deceive sink nodes or deplete the limited energy resource of forwarding nodes. In this paper, we propose an enhanced version of the commutative cipher-based filtering scheme (CCEF), which strengthens the detection power of CCEF by combining CCEF and the statistical en-route filtering scheme (SEF). Every report is verified by intermediate nodes with a certain probability in the fashion of both schemes. Such combined approach can provide early detection of fabricated reports, which results in energy-efficiency against the massive fabricated report injection attacks. The effective of the proposed scheme is shown with the simulation results at the end of the paper.

Key words:
Ubiquitous sensor networks, fabricated report injection attacks, fabricated report filtering, commutative cipher, security.

1. Introduction

Recent advances in low power wireless networking have accelerated the development of ubiquitous sensor networks (USNs) [1]. USNs consist of a large number of sensor nodes that monitor the environment, and a few sink nodes that collect the sensor readings [2]. Typical applications for USNs include sending messages to a node at a given location, retrieving sensor data from nodes in a give region, and finding nodes with sensor data in a given range [3]. In many applications, nodes are vulnerable to physical attacks, potentially compromising the node's cryptographic keys since they are deployed in open environments and are not unattended [4]. Attackers may use compromised nodes to inject fabricated reports into the network (Fig. 1) [5]. Fabricated reports will cause false alarms that waste real world response efforts, and drain the finite amount of energy in a battery powered network [6]. To minimize the grave damage, fabricated reports should be dropped en-route as early as possible, and the few eluded ones should be further rejected at sink nodes [7]. The early dropping of fabricated reports leads to significant savings of energy [6].

Recently, several security solutions [5,6,8-13] have been proposed to detect such fabricated reports before the reports consume a significant amount of energy. While each of their designs has its own merits, they may be inefficient or even useless if the number of compromised nodes exceeds a certain threshold value [7]. The fundamental reason is that their designs follow the symmetric key sharing approach in achieving the en-route filtering capability. CCEF [7] was proposed to defend against the fabricated report injection attacks without symmetric key sharing among nodes. CCEF can provide much stronger security protection against compromised nodes than the symmetric key sharing-based designs.

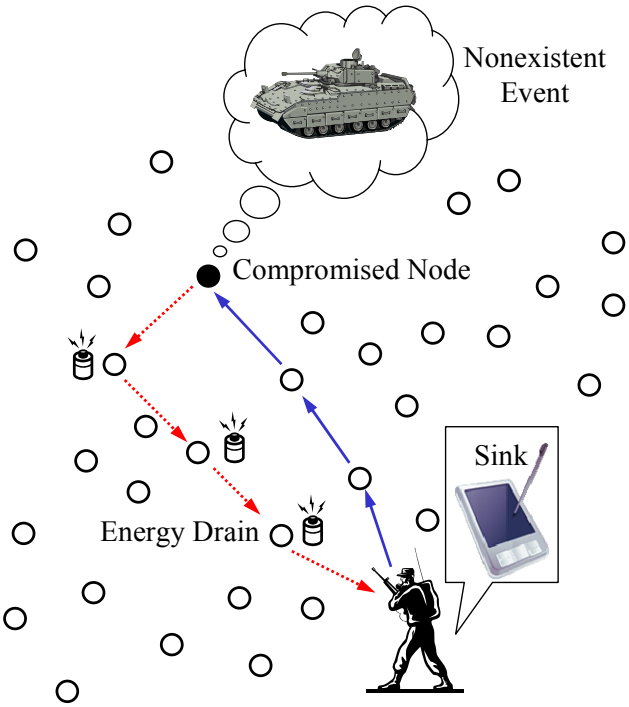

Fig. 1 Fabricated reports injection attacks in USN.

However, CCEF cannot filter out fabricated reports in case of the cluster head (CH) compromising. In this paper, we propose an enhanced version of CCEF, which can detect and drop fabricated reports in case of CH compromising. To achieve the goal, we combine CCEF with SEF [8] in which a report is forwarded only if it contains the message authentication codes (MACs)

Manuscript received March 10, 2008
Manuscript revised March 20, 2008

generated by multiple nodes, by using keys from different partitions in a global key pool. As a result, the proposed scheme can detect fabricated reports earlier than both schemes before they consume a significant amount of energy. The effectiveness of the proposed scheme against the fabricated report injection attacks is shown with the simulation results.

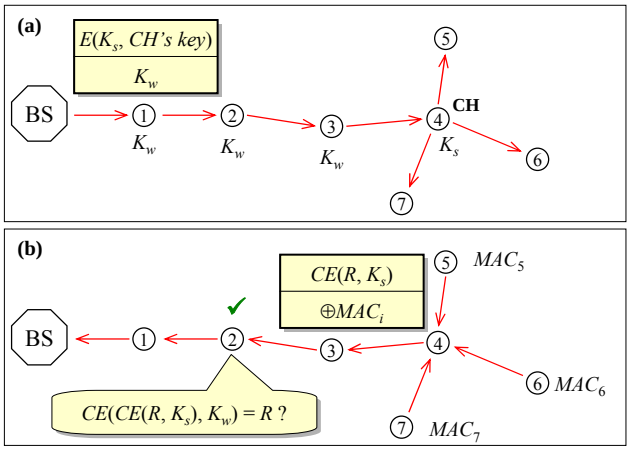

Fig. 2 Query-response in CCEF.

2. Background

2.1 Commutative Cipher-Based En-Route Filtering (CCEF)

Yang and Lu proposed CCEF [7] to defend against the fabricated report injection attacks without symmetric key sharing among sensor nodes. Every node loads a unique *authentication key*, which is shared only with sinks. For each session, a sink prepares the *session key* K_s and the *witness key* K_w that satisfy:

$$CE(CE(R, K_s), K_w) = R, \quad (1)$$

where CE is a commutative cipher. It then selects one sensor node within the interest region as CH and sends a query to CH (Fig. 2(a)). The query includes K_s encrypted by the CH's authentication key and K_w as plaintext. Each intermediate node stores K_w for future verification purpose. A sensing report is produced by CH. The report is endorsed with a MAC generated by CH using K_s (i.e., $CE(R,K_s)$) and multiple MACs generated by its neighboring nodes using their authentication keys. The report is forwarded to the sink along the reversed path as the query traverses (Fig. 2(b)). Every intermediate node can verify the report based on Eq. (1). The report is finally verified by the sink. In order to reduce the commutative cipher computation overhead, CCEF adopts a probabilistic approach in which a forwarding node verifies a report with a probability of:

$$p_{CCEP} = \frac{1}{\alpha \cdot h}, \quad (2)$$

where is a system parameter and h is the number of hops from CH to the sink. One of the major drawbacks of CCEF is that fabricated reports cannot be filtered during the forwarding process if CH is compromised.

2.2 Statistical En-Route Filtering (SEF)

SEF [8] is the first paper that addresses the fabricated report injection attacks in the presence of compromised nodes [6]. SEF can detect fabricated reports probabilistically. In SEF, sinks maintain a global key pool which is divided into multiple partitions. Every node loads a small number of keys from a randomly selected partition in the global key pool before the node is deployed. When an event occurs, one of the detecting nodes collects MACs for the event from its neighboring nodes (Fig. 3(a)). It then produces a sensing report and forwards the report to a sink. A report is forwarded if and only if it has multiple MACs generated by multiple nodes, using keys from different partitions in the global key pool (Fig. 3(b)). The overhead of SEF is relatively small [14]. However, it does not guarantee that a fabricated report can be always detected in forwarding. Moreover, it may be inefficient or even useless if the number of compromised nodes exceeds a fixed threshold value [7].

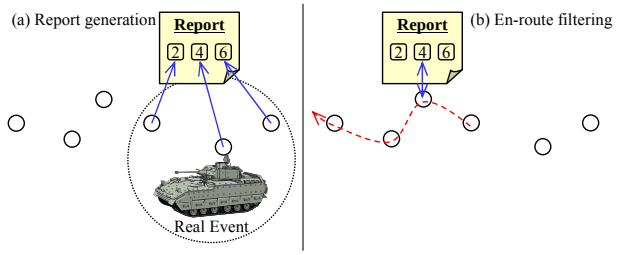

Fig. 3 Report generation and en-route filtering in SEF.

3. Commutative Cipher-Based, Probabilistic Filtering Scheme (CCPF)

3.1 System Models and Assumptions

We consider a sensor network composed of a large number of small sensor nodes. Nodes are similar to the current generation of sensor nodes (e.g., MICAz [15,16]) in their computational and communication capability and power resources. Nodes may be compromised or

physically captured. Due to cost constraints, we assume that each sensor node is not equipped with tamper-resistant hardware. Once compromised, a node can be used to inject fabricated reports into the network. We assume that sinks cannot be compromised. We also assume that sink have a mechanism to authenticate broadcast messages (e.g., based on μTESLA [17]), and every node can verify the broadcast messages.

3.2 Scheme Overview

Before node deployment, every sensor node is preloaded with an authentication key and some keys from a global key pool (Fig. 4(a)). Sinks initiate query-response sessions. For each session, a sink randomly selects one sensor node at the location of interest as CH. The sink constructs a query and sends it to CH (Fig. 4(b)). CH responses to the query by generating and endorsing a sensing report (Fig. 4(c)). The report is forwarded along the reversed path, as the query traverses. The report is verified by forwarding nodes in the fashion of both CCEF and SEF. That is, each intermediate node verifies the report using its keys (Fig. 4(d)) loaded from the key pool or using the witness key (Fig. 4(e)) with a certain probability. The report is finally verified by the sink.

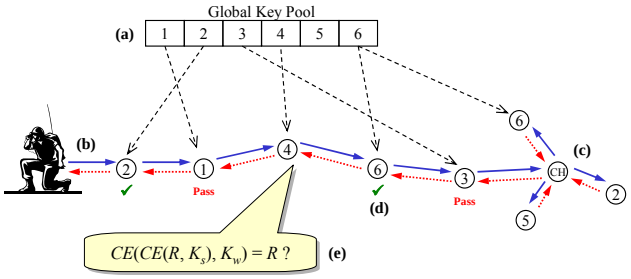

Fig. 4 CCPF overview.

3.3 Bootstrapping and Session Setup

Sinks maintain a global key pool of k keys, $\{K_0, \cdots, K_{k-1}\}$, divided into p non-overlapping partitions, $\{P_0, \cdots, P_{p-1}\}$. Each key has a unique key index. Every sensor node has a unique ID. It loads an authentication key that is shared only with sinks, and n keys ($n < k / p$), called *secret keys*, from a randomly selected partition in the global key pool together with the associated key indices, before node deployment.

For each session, a sink randomly selects one sensor node within the interest region as CH to be tasked through a query. It then prepares the session key K_s and the witness K_w that satisfy Eq. (1). The sink produces a query that includes the application-specific interests and the four fields required in CCEF (i.e., a query ID *QID*, the CH's ID, K_s encrypted by the CH's authentication key, and K_s as plaintext). The query is authenticated by an authentication scheme such as μTESLA [17]. The sink sends out the query, which is forwarded hop-by-hop to CH. Each intermediate node stores the query ID and K_w for further verification purpose.

3.4 Report Generation

The tasked node responses to the query by collaborative generation of a sensing report with its neighboring nodes. We ignore the collaborative report generation process here, and start with a scenario in which they have reached an agreement on the event description R, i.e., the content of the report. In CCPF, each report is endorsed by: 1) a MAC generated using the session key, 2) t MACs generated by t nodes using their secret keys from different partitions in the global key pool, where t ($\leq p$) is a security threshold value determined by the network designer. The former MAC is called the *session MAC*, and the latter is called the *secret MACs*. The indices of the secret key used to generate the secret MACs are also attached in the report. Thus, the report may have the following form:

$$\{QID, R, CE(R, K_s), i_1, i_2, \cdots, i_t, \\ M(R, K_{i_1}), M(R, K_{i_2}), \cdots, M(R, K_{i_t})\}, \quad (3)$$

where i_1, i_2, $\cdots$, i_t are key indices, $M(D,K)$ is the MAC of a message D generated using a secret key K. Finally, CH disseminates the report towards the sink.

3.5 En-Route Filtering

The report is forwarded along the reversed path as the query traverses. When an intermediate node receives a report, it first checks whether *QID* of the report is stored. If not, it drops the report. A legitimate report should include one session MAC, p key indices of distinct partitions, and p MACs generated using the keys indicated by the key indices. Reports with no session MAC or less than p key indices or less than p MACs or more than one key index in the same partition are dropped.

The session MAC of a report is verified by a node using the witness key with a probability p_{CCEF} in Eq. (2). Upon successful verification (i.e., the result of the commutative cipher computation for the session MAC matches with R), the node forwards the report toward the sink. If failed to verify, it drops the report immediately.

If a node does not have a chance to verify the session MAC of a report, it has a chance to verify the secret MACs. If it has any of the t keys indicated by the key indices in the report, it reproduces the MAC using its own key and compares the result with the corresponding MAC attached in the report. The report is dropped if the attached one differs from the reproduced. If they match exactly, or

the node does not have any of the t keys, the node passes the report to the next hop.

In CCEF, each intermediate node has a detection probability of Eq (2). In SEF, each node has a detection probability of:

$$p_{SEF} = \frac{n \cdot t}{k \cdot p}. \quad (5)$$

Thus, in CCPF, for a report, each intermediate node has a detection probability of:

$$p_{CCPF} = \frac{1}{\alpha \cdot h} + \left(1 - \frac{1}{\alpha \cdot h}\right)\frac{n \cdot t}{k \cdot p}. \quad (6)$$

3.6 Key Sharing and Security Analysis

Fig. 5 shows the essential concept of the key sharing in the three schemes. In CCEF, the en-route filtering capability is achieved by the key sharing between CH and intermediate nodes (Fig. 5(a)). Thus, to launch fabricated report attacks successfully, attackers have to compromise CH and at least t-1 nodes among CH's neighboring nodes. In real-world, it is very difficult to compromise such a number of nodes without being detected. However, it cannot detect and drop fabricated reports if CH is compromised (but, they are detected by sinks). In SEF, the en-route filtering capability is achieved by the key sharing among nodes (Fig. 5(b)). However, its detection power decreases with node compromising. Moreover, if t keys in distinct partitions are compromised, any fabricated reports cannot be detected during the forwarding process, even at sinks.

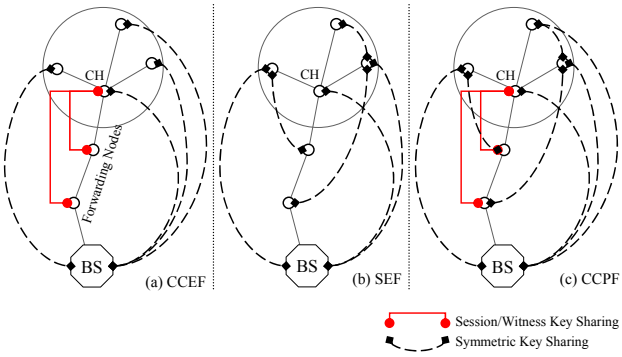

Fig. 5 Key sharing in three filtering schemes.

CCPF uses both key sharing approaches. That is, the en-route filtering capability is achieved by the key sharing among intermediate nodes, CH, and other nodes (Fig. 5(c)). CCPF can provide almost equivalent resilience against node compromising since it is an enhanced version of CCEF. CCPF also adopts the key sharing approach of SEF. Thus, it can filter out fabricated reports in case of CH compromising. As a result of the combination, it can provide early detection of fabricated reports. However, if attackers compromise CH and t keys from different partitions, CCPF cannot filter out any fabricated reports during the forwarding process. We can strengthen the resilience by assigning key indices to each query. For each session, a sink randomly selects t key indices from district partitions and attaches the key indices to a query. A legitimate report should contains t secret MACs using the keys indicated by the key indices attached in the query. Thus, it may be almost impossible to launch the fabricated report injection attacks successfully. Another benefit of such approach is that it can also be used to prevent the denial-of service (DoS) attacks.

4. Simulation Results

To show the effectiveness of CCPF, we compare CCPF with CCEF and SEF through the simulation. We use a field size of 500×30m^2, where 1,500 nodes are uniformly distributed. Each node takes 16.25, 12.5μJ to transmit/receive a byte [8]. Each MAC generation consumes 15μJ and one commutative cipher computation consumes 9mJ [7]. The size of an original report is 24 bytes. The size of a MAC is 1 byte. We use a global key pool of 1,000 keys for SEF.

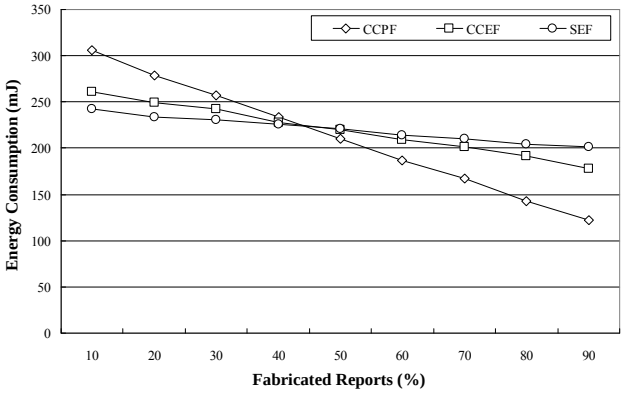

Fig. 6 Energy consumption per report.

Fig. 6 shows the average energy consumption per report in CCPF, CCEP, and SEF when the portion of false traffic takes from 10% to 90%. As shown in the figure, SEF (empty circles) can save energy when false traffic is a very small proportion of the total. CCEF (empty rectangles) is more energy-efficient than SEF if the portion of false traffic exceeds 50%. The dual verification approach of CCPF (empty diamonds) can lead to early detection of fabricated reports. Thus, it is the most efficient filtering scheme if most traffic is composed of fabricated report, in terms of energy saving.

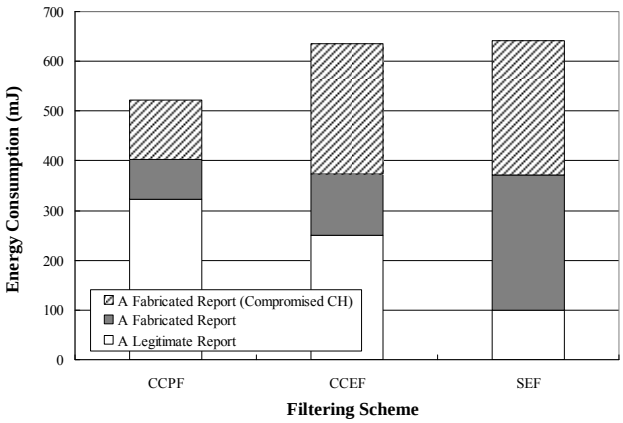

Fig. 7 Energy consumption per three kinds of reports.

Fig. 7 shows the average energy consumption caused by three kinds of reports – a legitimate report, a fabricated report, and a fabricated report generated by compromised CH – in the three schemes. As shown in the figure, for legitimate traffic, SEF is the most efficient solution in terms of energy saving. On the other hand, CCPF consumes much more energy than the others. However, CCPF is energy-efficient against false traffic due to its early detection capability. In terms of energy saving, CCEF may be a fair solution against false traffic, but it is very inefficient in case of CH compromising.

5. Conclusion and Future Work

In this paper, we proposed CCPF, in which can detect and drop fabricated reports in case of CH compromising. To achieve the goal, we combined CCEF with SEF. Every report is verified by forwarding nodes with a certain probability in the fashion of both schemes. As a result, CCPF can detect fabricated reports earlier than both schemes before they consume a significant amount of energy. The effectiveness of the proposed scheme against the fabricated report injection attacks was shown with the simulation results.

CCEP is very energy-efficient against false traffic, but consumes too much energy in delivering legitimate reports. To reduce energy consumption for legitimate traffic, we plan to apply the adaptive determining methods [18-20], which can result in energy saving by the adaptive determination of the security parameters such as a threshold value t or a in Eq. (2), to CCPF. In order to achieve more energy saving, we also plan to apply the adaptive filtering scheme method [14], which switches filtering schemes with the consideration of network status, to the CCPF-based networks. A filtering scheme for each session may be chosen among the three filtering schemes – CCPF, CCEF, and SEF – by considering network status.

Acknowledgments

This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Advancement). (IITA-2008-C1090-0801-0028)

References

[1] D. Kramarev, I. Kim, and K. Kim, "Type-Based Detection with a Fusion Center Performing the Sequential Test in Wireless Sensor Networks," *IEICE Transactions on Communications* **E90-B**(12), pp. 3354-3361, 2007.

[2] L. Buttyan, L. Dora, and I. Vajda, "Statistical Wormhole Detection in Sensor Networks," *Lecture Notes in Computer Science* **3813**, pp. 128-141, 2005.

[3] S. Li and D. Zhang, "A Novel Manifold Learning Algorithm for Localization Estimation in Wireless Sensor Networks," *IEICE Transactions on Communications* **E90-B**(12), pp. 3496-3500, 2007.

[4] I. Przydatek, D. Song, and A. Perrig, "SIA: Secure Information Aggregation in Sensor Networks," in *Proc. SenSys*, pp. 255-265, 2003.

[5] S. Zhu, S. Setia, S. Jajodia, and P. Ning, "An Interleaved Hop-by-Hop Authentication Scheme for Filtering of Injected False Data in Sensor Networks," in *Proc. S&P*, pp. 259-271, 2004.

[6] F. Li and J. Wu, "A Probabilistic Voting-based Filtering Scheme in Wireless Sensor Networks," in *Proc. IWCMC*, pp. 27-32, 2006.

[7] H. Yang and S. Lu, "Commutative Cipher Based En-Route Filtering in Wireless Sensor Networks," in *Proc. VTC*, pp. 1223-1227, 2003.

[8] F. Ye, H. Luo, and S. Lu, "Statistical En-Route Filtering of Injected False Data in Sensor Networks," *IEEE Journal on Selected Areas in Communications* **23**(4), pp. 839-850, 2005.

[9] Y. Zhang, J. Yang, and H.T. Vu, "The Interleaved Authentication for Filtering False Reports in Multi-path Routing based Sensor Networks," in *Proc. IPDPS*, 2006.

[10] H.Y. Lee and T.H. Cho, "Key Inheritance-Based False Data Filtering Scheme in Wireless Sensor Networks," *Lecture Notes in Computer Science* **4371**, pp. 116-127, 2006.

[11] W. Zhang and G. Cao, "Group Rekeying for Filtering False Data in Sensor Networks: A Predistribution and Local Collaboration-based Approach," in *Proc. INFOCOM,* pp. 503-514, 2005.

[12] Y. Zhang, W. Liu, W. Lou, and Y. Fang, "Location-Based Compromise-Tolerant Security Mechanisms for Wireless Sensor Networks," *IEEE Journal on Selected Areas in Communications* **24**(2), pp. 247-260, 2006.

[13] Z. Yu and Y. Guan, "A Dynamic En-route Scheme for Filtering False Data Injection in Wireless Sensor Networks," in *Proc. SenSys*, pp. 294-295, 2005.

[14] H.Y. Lee and T.H. Cho, "Fuzzy Adaptive Selection of Filtering Schemes for Energy Saving in Sensor Networks," *IEICE Transactions on Communications* **E90-B**(12), pp. 3346-3353, 2007.

[15] http://www.xbow.com/

[16] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, “System Architecture Directions for Networked Sensors,” in *Proc. ASPLOS*, pp. 93-104, 2000.
[17] A. Perrig, R. Szewczyk, J.D. Tygar, V. Wen, and D.E. Culler, "SPINS: Security Protocols for Sensor Networks," *Wireless Networks* **8**(5), pp. 521-534, 2002.
[18] H.Y. Lee and T.H. Cho, "Fuzzy-Based Adaptive Threshold Determining Method for the Interleaved Authentication in Sensor Networks," *Lecture Notes in Artificial Intelligence* **4293**, pp. 112-121, 2006.
[19] H.Y. Lee and T.H. Cho, "Fuzzy Adaptive Threshold Determining in the Key Inheritance Based Sensor Networks," *Lecture Notes in Artificial Intelligence* **4570**, pp. 64-73, 2007.
[20] H.Y. Lee and T.H. Cho, "Fuzzy Security Parameter Determining Method for the Commutative Cipher Based Filtering in Sensor Networks," *Lecture Notes in Computer Science* **4706**, pp. 573-583, 2007.

Hae Young Lee received a B.S. degree in electrical and computer engineering from Sungkyunkwan University, South Korea, in 2003. From 1998 to 2001, he worked for several companies as a programmer. He is currently a doctoral student at the School of Information and Communication Engineering, at Sungkyunkwan University, South Korea. His research interests include ubiquitous sensor networks, intelligent systems, computer-aided design, artificial intelligence, and modeling & simulation.

Tae Ho Cho received a Ph.D. degree in electrical and computer engineering from the University of Arizona, USA, in 1993, and B.S. and M.S. degrees in electrical engineering, from Sungkyunkwan University, South Korea, and the University of Alabama, USA, respectively. He is currently a Professor in the School of Information and Communication Engineering, at Sungkyunkwan University, South Korea. His research interests include ubiquitous sensor networks, intelligent systems, modeling & simulation, and enterprise resource planning.