A Secure and Efficient Routing and Aggregation Mechanism for Sensor Networks

Inshil Doh and Kijoon Chae

Ewha Womans University, Korea

Summary

For secure sensor network communication, secure and efficient routing mechanism is essential. However, due to the basic constraints of sensor network, conventional routing or security mechanisms cannot be applied. In this work, cluster-based on-demand routing mechanism is proposed. The mechanism also provides security by attaching multiple MAC values and increases the security level by computing keys on-demand. Instead of finding optimal path, it randomly picks a next clusterhead among several candidates. The robustness against energy depletion also rises in this way. Secure aggregation is further suggested.

Key words : *Sensor network, routing, security, clustering, aggregation*

1. Introduction

Sensor networks will be the core technology towards new network appliances such as ubiquitous computing. They can be applied at a low cost and cover large area sensing various data. Because of their low cost, sensor networks can be applied in various application area. However, sensor networks also introduce severe resource restraints due to lack of data storage and power, and the conventional routing protocols cannot be applied. A lot of routing mechanisms have been proposed so far. However, most of them are not considering the security, which is critical for sensor network because the basic constraints make the network more vulnerable to various attacks than the other wireless networks. So security must be justified and ensured before the large scale deployment of sensors. In this paper, we propose a secure routing mechanism for efficient sensor communications. One of our contributions is that we propose secure and efficient routing mechanism which is based on hexagonal shaped clustered network architecture. Our proposal is dynamic routing path selection made by clusterheads and it also provides message authentication with keys which are computed on-demand manner. It lengthens the lifetime of the network by diffusing the energy consumption and makes the network more resilient against node capture attack by preventing the attacker from stealing the raw keys. We also suggest a secure aggregation method by letting the base station verify the initial MAC values.

The organization of this paper is as follows. Section 2 gives an overview of related work. Assumptions and network model for our mechanism are given in section 3. Section 4 describes our proposal for secure routing mechanism. Secure aggregation mechanism is presented in section 5. Security and overhead analyses are described in section 6. In section 7, we conclude the paper and present some future research directions.

2. Related work

A lot of sensor network routing protocols have been proposed so far. They can be classified into two categories, one is flat routing and the other is hierarchical routing. Our proposal is included in the second category. Here, we mention several representative protocols for each category.

In flat routing, every sensor node equally participates in routing mechanism. [1][2][3][4] can be classified in the flat routing protocols. Sensor Protocols for Information via Negotiation (SPIN)[1] assumes all of the sensor nodes are potential sinks. Every node uses meta-data, i.e. high-level data descriptors, and before any data is really transmitted, a node performs meta-data negotiations. This assures that there is no redundant data sent throughout the network. In the Minimum cost forwarding approach[2], the sink broadcasts an ADV message containing its own cost to its neighbors, and each node receiving the message sets a timer. Once the timer expires, the node changes its cost to the new one, and rebroadcast the ADV message containing the new cost. When a source has data to send, it simply broadcasts it, and only nodes having a cost that matches the difference between the cost contained in the message and the consumed cost, rebroadcast the data. This process is continued until the data arrive at their destination.

Hierarchical routing is also called cluster based routing protocols[5][6][7][8][9]. They cluster the network, and in every cluster, a clusterhead collects the data from their own members and deliver the data to base station or upper layer clusterheads. LEACH[9] is a representative clustering-based protocol that utilizes randomized rotation of the clusterheads to evenly distribute the energy load among the sensor nodes in the network. The protocol uses dynamic clusterheads mechanism to avoid the energy depletion of selected clusterhead, employs localized coordination to improve the scalability and robustness. It also uses data fusion to reduce the amount of information transmitted between sensor nodes and the base station. Power-Efficient Gathering in Sensor Information Systems (PEGASIS)[6] is a chain-based power efficient protocol based on LEACH. It assumes that all nodes have location information about all other nodes and each of them has the capability of transmitting data to the base station directly. Because each node has global knowledge of the network, the chain can be constructed easily by using a greedy algorithm. To balance the overhead involved in communication between the leader and sink, each node in the chain takes turn to be the leader. Nodes fuse the received data with their own data when data are transmitted in the chain. TEEN[8] is another cluster-based routing protocol based on LEACH. It has Hard Threshold(HT) and Soft Threshold(ST). A node which has a sensed value ready determines whether to report it or not based on the values of HT and ST. Data are reported only when the sensed value exceeds HT or the value's change is bigger than ST. TEEN employs LEACH's strategy to form clusters. There are a lot of other routing protocols which have been proposed considering efficiency and energy consumption. However, they are usually not considering the security aspect which is especially important when sensor nodes are deployed in hostile environment. In this research, we consider the security problem as well as the efficiency and energy consumption problems.

For authentication, Zhu et al.[10] proposed a mechanism which enables the base station to verify the authenticity of event reports when compromised nodes exist and also filters the false data packets injected into the network before they reach the base station. This mechanism has the overhead of forming associations between nodes and all the nodes should keep pairwise keys with their associated nodes which are t-hops away.

3. Network model

3.1 Network architecture

Sensor field is clustered in hexagonal shape and in every cluster a clusterhead is located at the center of each cluster and the clusters are interconnected by several gateway nodes. Clusterheads are deployed in the predefined positions and normal sensor nodes are deployed randomly in their own clusters. Network architecture is shown in Fig. 1.

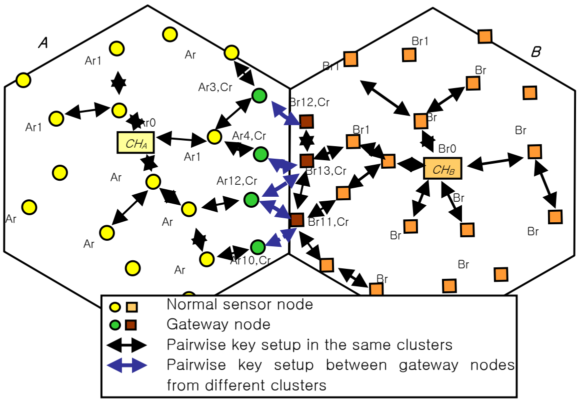


Fig. 1. Network clustering and pairwise keys between nodes

3.2 Assumptions

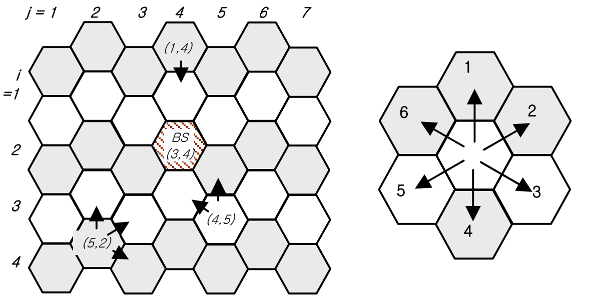
In our mechanism, all nodes are static and clusterheads are located in each cluster areas. They are more powerful in computation and have longer lifetime.

We adopts Blom's scheme[11] to establish pairwise keys between all pairs of neighboring nodes and all pairs of clusterheads. In every cluster, there is a dedicated matrix for establishing pairwise keys between neighboring nodes. Between neighboring clusters, shared matrices are used by gateway nodes to setup pairwise keys with neighboring nodes belonging to different clusters. Clusterheads are responsible for distributing key materials to their gateway nodes' requests and they become to know which nodes are their own gateways. For the base station and clusterheads, other shared matrices are assigned, and each row from respective matrices are predistributed to all the clusterheads, i.e, every clusterheads and the base station carry as many rows as the number of shared matrices for themselves. Every pair of clusterheads including the base station can compute its own pairwise keys on-demand way using the key material they have been predistributed.

4. Secure Routing

4.1 Routing mechanism

Every cluster is assigned to a coordinate (*i*,*j*) on the network field. As in Fig 2, the coordinates look a bit different from those of rectangular shape because we cluster the field in hexagonal shape. Every clusterhead plays an important role in routing. They gather the sensed data from their own member sensor nodes and then deliver the aggregated data to the base station through the other clusterheads. Routing is made by the clusterheads which can compute the other clusterhead's position and the data is moved toward the base station through the clusterheads. We located base station at the center of the field. However, wherever the base station may be located, it has its own coordinate and every clusterhead can compute the difference between its own coordinate and that of the base station. The steps of transmitting the data are as follows.



(a) Clusters' coordinates (b) Possible movements

Fig. 2. Clusters' coordinates and possible movements

– Step 1 : Sensor nodes sensing an event generate packets and deliver the packet to their own clusterheads with MAC values computed by the pairwise keys between each sensor node and their clusterheads. Intermediate sensor nodes just deliver the data to their clusterhead without verifying the MAC values.

– Step 2 : When clusterheads receive the report packet, they verify the MAC values and then generate a new report packet which has three destinations. Destination 1 is base station, destination 2 is next clusterhead toward the base station, and destination3 is the gateway node which is responsible for transmitting the report packet to the next cluster. Gateway node is randomly selected among the multiple gateway nodes by the clusterhead and the next cluster is selected according to the routing path selection rules. In this step, the clusterhead adds three MACs. MAC_{BS} is computed using pairwise key with the base station, MAC_{CH} using the key with next clusterhead on the routing path to the base station, and MAC_{GW}, using the key beetween clusterhead and the gateway node. In this way, not only outsider attacks, but also the insider attacks by compromised nodes can be detected. The outsider attacks are impossible because the adversaries don't know the pairwise keys and cannot generate legitimate MACs. Insider attackers, i.e. compromised nodes, can modify the data and generate fake MAC values, but this will be detected by the next gateway node and then clusterhead, and even when the gateway node or clusterhead cannot filter the false data, it can be finally detected by the Base station. Security and overhead is further described in section 6.

– Step 3 : Report packet generated at step 2 is broadcasted in the cluster where the generating clusterhead is located. When the gateway node in the report packet receives the report packet, it transfers the report packet to neighbor cluster after it verifies the MACGW and generate a new MACGW with its partner gateway node belonging to other cluster.

– Step 4 : The gateway in the neighbor cluster broadcasts the report packet in its own cluster after it verifies and generates MACGW if the verification is true.

– Step 5 : When the clusterhead in the neighbor cluster gets the report packet, it checks the two MAC values, generates two new MAC values, one for the next clusterhead, and the other for the gateway node. And then the clusterhead broadcasts the new report packet.

– Step 6 : When the report packet is delivered to the base station through step 1 to step 5, base station finally checks three MAC values, one from the beginning clusterhead, the other from neighbor clusterhead, and the other from gateway node. When the MAC values are true, the base station accepts the report packet. Blom's scheme[11] has the property of λ-security, i.e. when more than λ rows are compromised, the secret matrix is derived and all keys can be computed by adversaries. So, λ is very critical for the security and should be properly determined. We omit further descriptions about Blom's scheme here.

4.2 Routing path selection rules

Every cluster including base station is assigned to a coordinate(i,j). When a clusterhead tries to send a report packet, it computes the coordinate difference between itself and the base station. It needs to increment or decrement i or j values. As in Fig. 2, when the cluster is located at the same row or column with the base station, it needs to increment or decrement the column or row to get to the base station. When it is located at different column and different row, it needs to move by changing x or y coordinate under some rules. They pick the operation randomly every time they need to make the movement to decrease the energy consumption of certain nodes on the route to the base station. The routes may not be the optimal path to the base station. However, in this way, we can provide resilience against energy consumption attack.

Table 1. Movement directions

Columns	1	2	3	4	5	6
Odd column	r— c 00	r00 c++	r++ c++	r++ c00	r++ c—	r00 c—
Even column	r— c 00	r— c++	r00 c++	r++ c00	r00, c—	r— c—

Each clusterhead can choose one of the movement in the Table 1 according to directions in Fig.2(b). (++) means increment of the coordinate, (—) the decrement, and (00) is to keep the same coordinate value. Through some of the options, we expect efficient movement which is similar to diagonal movement in rectangular clusters through just one step. This is possible because we choose hexagonal shape and coordinates. By choosing one of the movements, each clusterhead can deliver the packet to the next cluster on the route, and the next cluster can choose another movement through the same process. To decide which movement to choose, we set three rules.

– Rule 1 : Clusters in the same row as the base station can choose one of the four movements, direction 2 or 3 for the clusters on the left side of the base station and direction 5 or 6 for the clusters on the right side of the base station.

– Rule 2 : clusters in the same column as the base station can choose two directions, direction 1 for the clusters on the down side of the base station and direction 4 for the clusters on the upper side of the base station.

– Rule 3 : clusters in different rows and different columns from the base station can choose one of the six movements, direction 1,2,3, or 2,3,4, or, 4,5,6, or 5,6,1 according to their positions.

4.3 Consideration for security enhancement

In step 2 of routing processes, clusterheads generate MACs using pairwise keys with base station and next clusterhead on the route to the base station, and the gateway node respectively. For keeping the pairwise keys between clusterheads secret, they don't keep the keys themselves, but just carry the rows from shared matrices and compute the pairwise keys every time they need to generate MAC values. If they keep the key values, all the clusterheads carry six pairwise keys with their respective neighbor clusterheads and one pairwise key with the base station. In this case, when adversaries capture the clusterheads, the whole network could be in danger because pairwise keys between clusterheads are very important. The security analysis is further discussed in 6.1.

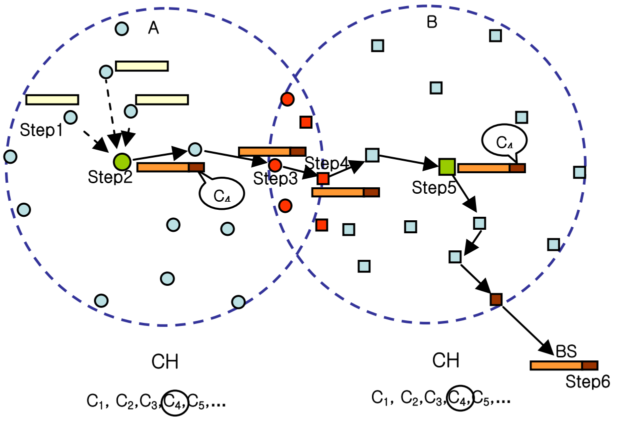


Fig. 3. Security enhancement method

In Fig.3, CHA chooses a row from shared matrix C4 and generates MACCH with a row value from C4. CHB knows that the MAC was generated with a pairwise key using shared matrix C4 by checking the value in the packet, and it verifies the value with the pairwise key after key computation. Base station verifies the MAC value MACBS in the same manner. By computing pairwise key on-demand manner, even if nodes are captured, the attackers cannot get the keys.

As mentioned in 4.1 briefly, gateway node is also selected randomly every time data need to be transferred from one cluster to another. This is for decreasing the energy consumption of a particular gateway node and protecting the node from energy depletion.

5. Secure aggregation

In some cases, an event is sensed in large area. In this case, if respective event report is delivered to the base station, the energy is consumed unnecessarily. This fact could be used for energy consumption attack. So data aggregation is needed. For example, as in Fig. 4, when an event 1 is sensed in the cluster area 1, 2, 3, 4, normal sensor nodes create and send a data packets to their own clusterheads.

When each clusterhead aggregates the data it generates a new report packet and relays it along the routing path. After an intermediate clusterhead receives these reports, it waits a predefined time period, gathers and compares the delivered packets, their event IDs and source IDs. When the clusterhead decides that the packets are created by the same event, it just increments the count value and delivers only one packet to the next clusterhead to the base station. In Fig.4, after each clusterhead gathers the sensing packets from its own member nodes, first aggregation is made by

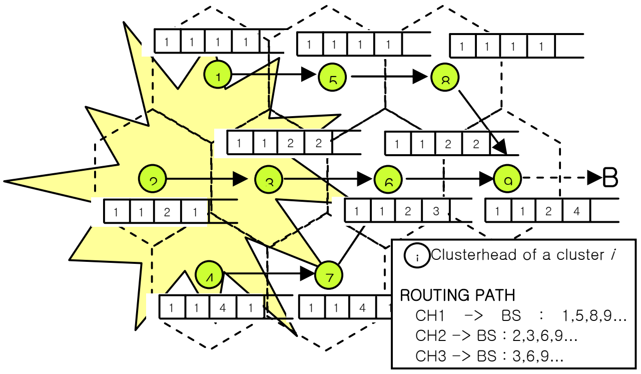
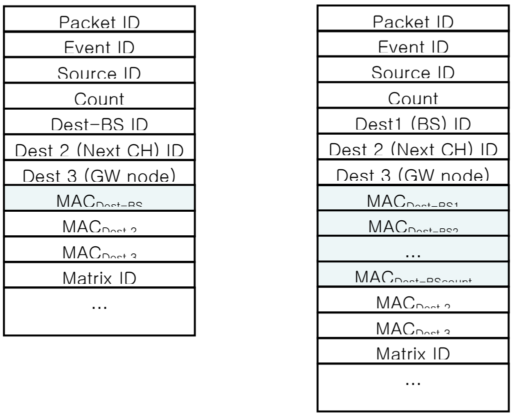


Fig. 4. Data aggregation by clusterheads

clusterhead 3(aggregating report packets from cluster 2 and 3), second aggregation is made by clusterhead 6(aggregating report packets from 3 and 7), and finally clusterhead 9 aggregates two event report packets(from 6 and 8) and generates one report packet which says the event ID is 1, the originating cluster is cluster 2, and the number of reporting packets it is aggregating is four. In this way, redundant traffic can be decreased drastically. In this case, the aggregating clusterhead needs to deliver the original MACBS computed by the source clusterhead(cluster 1,2,3, and 4) for the base station to check if they are true or false. The source clusterhead which generates the report packet should not be compromised, and we need further research to guarantee the authenticity of the first packet generated. This is one of our future research. Packet format is shown in Fig. 5.



Packet ID
Event ID
Source ID
Count
Dest-BS ID
Dest 2 (Next CH) ID
Dest 3 (GW node)
MAC$_{Dest-BS}$
MAC$_{Dest 2}$
MAC$_{Dest 3}$
Matrix ID
…

(a) Packet by initiating clusterhead

Packet ID
Event ID
Source ID
Count
Dest1 (BS) ID
Dest 2 (Next CH) ID
Dest 3 (GW node)
MAC$_{Dest-BS1}$
MAC$_{Dest-BS2}$
…
MAC$_{Dest-BScount}$
MAC$_{Dest 2}$
MAC$_{Dest 3}$
Matrix ID
…

(b) packet by other clusterhead

Fig. 5. Packet format of the event report packet

6. Performance evaluation

6.1 Security analysis

6.1.1 Routing security

In routes finding process, every clusterhead randomly chooses next clusterhead toward the base station according to the route selecting rules. They can use different routes every time they need to deliver the data. The farther the cluster is located from the base station, the more number of routes exist to the base station. In this way, it is not easy for the adversaries to guess the routes to the base station. Additionally, we can divide the traffic load to multiple routes and make the lifetime of the whole network much longer. The mechanism further resists against the similar energy depletion attacks. Not only the clusterheads, but also the gateway nodes are randomly chosen. The gateway nodes' energy should be also saved because they need to consume a lot of energy to deliver the data from a cluster to another. For efficiency, routes can proceed in the diagonal directions because the clusters have hexagonal shapes. Our routing rules do not guarantee the optimal paths the to base station. They just suggest rough direction. However, because of the basic characteristics of sensor network, we should not use only one optimal paths repeatedly, and sometimes we need roundabout ways because of problems such as energy consumption or attacks. So our mechanism can provide easy and efficient way to find the dynamic routes. Data can be securely delivered using attaching MAC values. We use three MAC values some of which are dynamically computed, too. By choosing key material and compute the key dynamically, each clusterhead don't have to keep important pairwise key with them. If they keep the keys, the keys can be used by the attackers when clusterheads are captured by adversaries. In our mechanism, even though the clusterheads are captured, the attackers cannot recover the pairwise keys with the other clusterheads or base station. They need to capture more clusterheads than the value λ to recover the keys, which is impossible if we keep the number of rows of the shared matrix over the value, λ. In this way, we can prevent keys from being disclosed.

6.1.2 Aggregation security

Aggregation can make the network very dangerous because if the aggregating nodes are compromised and modify or forge the report packets. To prevent this kind of attack, we put every MACBS from every clusterheads when aggregating the report packets from each clusterheads. Base station finally verifies the MAC values when it receives the aggregated packet. When an event is sensed in large area and data aggregation is made by the intermediate clusterheads, there will be as many MAC_{BS} as the number of initiating clusterheads(count field in the packet format). more than one in the event report packet. Base station can decide which clusterheads are delivering fake messages.

6.2 Overhead analysis

6.2.1Storage Overhead

We don’t need space for keeping the routes to the base station because the clusterheads compute the routes on-demand way, and in every cluster, the data is broadcasted to all the sensor nodes. Packet size increases for the MAC fields, which is one byte long. We add three MAC values and additional field such as count, next clusterhead, gateway node. They can be tolerable considering the decrease of the redundant traffic generated by normal sensor nodes. Another major storage overhead is the key material that the clusterheads need to carry to compute the keys dynamically. Based on Blom’s scheme, they carry as many rows as the number of shared matrices for clusterheads. One row has (λ+ 1) keys, and when the key is 128 bits long, clusterheads need additional (16 × n) bytes, where n is the number of rows the clusterheads carry. This can be tolerable because we assumed that the clusterheads have larger memory than the normal sensor nodes, and considering the importance of disclosure of the pairwise keys between clusterheads by node capture attack, applying on-demand key computation is critical for security. For intermediate clusterheads aggregating data from other clusterheads need to keep additional MACs for the base station to verify. In the aggregated event report packet, there is as many MAC_{BS} as the number of clusters which detected the event. This overhead can be also neglected.

6.2.2 Computation Overhead

Computation overhead is a bit higher for clusterheads than for normal sensor nodes. They verify and generate two MACs, one for the next clusterhead and the other for the gateway node, while normal sensor nodes compute one MAC generation for their own clusterhead when they sense an event. Only the initiating clusterhead generates three MACs, one for the base station, another for the next clusterhead, and the third for the gateway node. This can be ignored because the energy for computing one MAC is about the same as that for transmitting one byte[12]. Intermediate clusterheads processing aggregation need several comparisons for checking if the data is from the same event or not. Verification and generation of new packet for the intermediate clusterheads are as same as the other clusterheads. Base station has to verify as many MACs as the number of initiating nodes and two more MACs, one from the previous clusterhead and the other from the gateway node. Usually, base station has a lot of computation and communication ability, and this fact does not make any problem.

6.2.3 Communication overhead

Communication overhead is very important because it consumes a lot of energy compared to other overhead. The amount of traffic can be decreased drastically by two step aggregation, once by every clusterhead when they get the report by their own member sensor nodes and the other by the intermediate clusterheads. In this way, redundant traffic can be decreased while keeping the required security level.

7. Conclusion and Future work

In this paper, we propose efficient routing mechanism based on a hexagonal shaped cluster architecture. In the mechanism, every clusterheads play important roles by verifying and generating MAC values to prevent the attackers from modifying or forging data packets. They choose one of the clusters which are located closer to the base station. This choice is made on-demand manner, and we can decrease the energy consumption of routing nodes. We also propose security enhancement mechanism by computing pairwise keys between clusterheads when they need to transfer the report packets. In this way, our proposal increases the resilience against node capture attacks because the clusterheads don’t keep the raw keys. Finally, by aggregating the data, we can further decrease the redundant traffic and hence, reduce the communication overhead.

For our further research, we are going to simulate our routing mechanism and make comparison with the other routing protocols. We will also model possible attacks in our mechanism and design attack detection and prevention mechanisms.

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-2006-C1090-0603-0028).

References

[1] W. Heinzelman, J. Kulik, and H. Balakrishnan, Negotiation-based Protocols for Disseminating Information in Wireless sensor Networks, Proc. of the 5th Annual ACM/IEEE International Conf. on Mobile Computing and Networking, 1999.

[2] F. Ye, A. Chen, S. Liu, and L. Zhang, A Scalable Solution to Minimum Cost Forwarding in Large Sensor Networks, Proc. of the 10th International Conf. on Computer Communications and Networks, 2001.

[3] K. Sohrabi, J. Gao, V. Ailawadhi, and G. Pottie, Protocols for Selforganization of a Wireless Sensor Network, IEEE Personal Communications, vol. 7, Issue 5, 2000.

[4] D. Estrin, R. Govindan, and J. Heidemann, Next Century Challenges: Scalable Coordination in Sensor Networks, Proc. of the 5th Annual ACM/IEEE International Conf. on Mobile Computing and Networking, 1999.

[5] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, Span: an Energy-efficient Coordination Algorithm for Topology Maintenance, Proc. of the 7th Annual International Conf. on Mobile Computing and Networking, July 2001.

[6] S. Lindsey and C. Raghavendra, PEGASIS: Power-Efficient Gathering in Sensor Information Systems, International Conf. on Communications, 2001.

[7] D. Estrin, R. Govindan, and J. Heidemann, Next Century Challenges: Scalable Coordination in Sensor Networks, Proc. of the 5th Annual ACM/IEEE International Conf. on Mobile Computing and Networking, 1999.

[8] A. Manjeshwar and D. Agrawal, TEEN: A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks, Proc. of the 15th Parallel and Distributed Processing Symposium, 2001.

[9] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, Energy efficient Communication Protocol for Wireless Micro Sensor Networks, Proc. of the 33rd Annual Hawaii International Conf. on System Sciences, 2000.

[10] 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, Proc. of IEEE Symposium on Security and Privacy, 2004.

[11] R. Blom, An optimal class of symmetric key generation systems. Advances in Cryptology, Proc. of EUROCRYPT 84, LNCS 209, 1985.

[12] F. Ye, H. Luo, S. Lu, and L. Zhang, Statistical En-route Detection and Filtering of Injected False Data in Sensor Networks, Proc. of IEEE INFOCOM 2004.

Inshil Doh received the B.S and M.S degrees in Computer Science from Ewha Womans University in 1993 and 1995, respectively. During 1995 – 1998, she worked for Samsung SDS.
She is now at Ph.D course at Computer Science and Engineering from Ewha Womans University.
<Research area> Network security, Ad-hoc network, sensor network, Ubiquitous computing security
e-mail: isdoh@ewhain.net

Ki-Joon Chae received the B.S degree in Math from Yonsei University and M.S degree in Computer Science from Syracuse University in U.S.A in 1982 and 1984, respectively. He received Ph.D degree in Computer Engineering from North Carolina State University in U.S.A.
During 1990 – 1992, he was a professor at Computer science from the Naval Academy in U.S.A. He is now a professor at Computer Science and Engineering from Ewha Womans University.
<Research area> Network security, Internet/Wireless communications network/High speed network protocol design and performance evaluation, sensor network, Home network, Ubiquitous computing security
e-mail: kjchae@ewha.ac.kr