PAPER

DIRECT: Dynamic Key Renewal Using Secure Cluster Head Election in Wireless Sensor Networks

Gicheol WANG[†a)], *Member*, Kang-Suk SONG[††], *and* Gihwan CHO[†††], *Nonmembers*

SUMMARY In modern sensor networks, key management is essential to transmit data from sensors to the sink securely. That is, sensors are likely to be compromised by attackers, and a key management scheme should renew the keys for communication as frequently as possible. In clustered sensor networks, CHs (Cluster Heads) tend to become targets of compromise attack because they collect data from sensors and deliver the aggregated data to the sink. However, existing key renewal schemes do not change the CH role nodes, and thus they are vulnerable to the compromise of CHs. Our scheme is called DIRECT (DynamIc key REnewal using Cluster head elecTion) because it materializes the dynamic key renewals through secure CH elections. In the scheme, the network is divided into sectors to separate CH elections in each sector from other sectors. Then, sensors establish pairwise keys with other sensors in their sector for intra-sector communication. Every CH election round, all sensors securely elect a CH in their sector by defeating the malicious actions of attackers. Therefore, the probability that a compromised node is elected as a CH decreases significantly. The simulation results show that our approach significantly improves the integrity of data, energy efficiency, and network longevity.

key words: *key management, cluster head election, wireless sensor networks, integrity, energy efficiency*

1. Introduction

In sensor networks, sensors are deployed in an unprotected environment and their data is delivered via wireless communication, so attackers can acquire data by eavesdropping and even fabricate data delivered from sensors to the sink. To defeat these threats, data from the sensors should be protected by means of cryptographic keys. This protection requires a key management scheme that generates cryptographic keys between the sensors and distributes them to the sensors securely. In other words, key management in sensor networks is an essential prerequisite for wide deployment of sensor networks. However, the constrained resources of sensors such as low bandwidth and computing power, small memory space, and limited battery power make key management a complicated problem [1], [2].

A lot of key management schemes have been proposed so far, and they are classified into two categories. The schemes in the first category have no key renewal mechanisms [3]–[7]. In these schemes, sensors typically have some administrative keys pre-distributed from the sink and they establish communication keys with neighbor sensors using these keys. The administrative keys are never renewed, and they are employed until the network is extinguished. Moreover, these administrative keys are taken by other sensors with a predefined probability, so these schemes severely degrade the security of the network as the number of compromised sensors increases.

The schemes in the other category have reactive key renewal mechanisms. That is, they renew the keys revealed to compromised sensors whenever a compromised sensor (s) is detected [8]–[12]. In most of these schemes, the physical network is divided into some logical groups, which are called clusters, to distribute the key management load to each logical group. Each logical group elects its own leader which is called CH and the key management duty is delegated from the sink to the CHs. If a CH is compromised, member sensors under the compromised CH are redistributed to non-compromised CHs. The non-compromised CHs distribute administrative keys used for key renewal to their new members and renew some of them if they are exposed to attackers. Then the CHs renew the group key used for intra-cluster communication using the renewed administrative keys. If only some members are compromised in a cluster, the CH renews the administrative keys of the compromised members and then renews the group key using the renewed administrative keys. Because these schemes employ one group key per cluster for communication, only one compromised sensor in a cluster can expose the group key. A more serious problem is that the CHs are likely to be targets of compromise attack because the CH role nodes are not changed.

In clustered sensor networks, member nodes as well as CHs are likely to be compromised since they are deployed in an unattended environment. If a member node is compromised by an attacker, its data is revealed to the attacker, and the attacker can send falsified data to the sink via the compromised node. The compromise of CHs has a more serious impact on the network than that of member nodes because CHs aggregate data from member nodes and deliver the aggregated data to the sink. An example showing the threat well is a military surveillance application. In this application, nodes monitor the movement of enemy troops and then notify headquarters of their invasion. However, compromised nodes can send forged information to the sink indicating that there is no suspicious activity. Especially, if all CHs are compromised, the control of the whole network is given to the enemies and their invasion is never detected.

Manuscript received September 28, 2009.

[†]The author is with the Dept. of Computing and Networking Resources, Korea Institute of Science and Technology Information, Korea.

[††]The author is with Corebell Corporation, Korea.

[†††]The authors is with CAIIT and Div. of Electronics and Information Engineering, Chonbuk Nat'l University, Korea.

a) E-mail: gcwang@kisti.re.kr

DOI: 10.1587/transinf.E93.D.1560

Therefore, a key management scheme which deals with the compromise of CHs is required. To cope with the compromise of CHs, a key management scheme should include a secure CH election mechanism which rotates the CH role nodes among non-compromised nodes.

In this paper, we propose a novel proactive key renewal scheme that resolves the security flaws of the reactive key renewal schemes. First, our scheme periodically changes CH role nodes in a cluster while preventing a compromised node from being elected as a CH. Second, in our scheme, each node employs a distinct key from other nodes in the same cluster for communication with its CH. Our scheme does not directly renew the communication keys in contrast with the reactive renewal schemes. Instead, our scheme performs key renewals using secure CH elections, so it is called DIRECT (DynamIc key REnewal using Cluster head elecTion) in this paper. After deployment, all sensors are grouped into some sectors to avoid the interference between CH elections in the network. Then, sensors establish pairwise keys with other sensors in their sector. These keys are employed for the communication between a CH and its members in the sector. Then, each sector elects a CH securely by defeating the malicious actions of compromised nodes. The secure CH election is periodically invoked until the network's extinction. As a result, the CH role nodes and the keys employed for communication between a CH and its members are renewed periodically. Therefore, our scheme is robust against the compromise of CHs and member nodes.

The rest of this paper is organized as follows. Section 2 gives a brief overview of the existing key renewal schemes and CH election schemes. Sections 3 describes the network and threat model that are assumed in this paper, and Sect. 4 describes the DIRECT in detail. We provide the simulation results and qualitative comparison in Sect. 5 and deal with the synchronization issue in Sect. 6. Finally, we conclude this paper in Sect. 7.

2. Related Work

Up to now, a number of key management schemes [3]–[7] have been proposed to establish the communication keys among sensor nodes. Eschenauer and Gilgor proposed a communication key establishment scheme using key predistribution for the first time [3]. In this scheme, any two neighbor sensors establish a communication key using common pre-distributed keys. If they have no common keys, then they establish the communication key indirectly through proxy nodes. Here, proxy nodes refer to the nodes that share at least one common key with the two nodes. The problem of this scheme is that any two nodes that share only one common key can establish a communication key. Therefore, it is very vulnerable to the increase of compromised nodes. Chan et al. resolved this problem by fixing the minimum number of common keys required for communication key establishment to $q(> 1)$ [4]. Du et al. proposed a scheme in which attackers scarcely obtain keys of a non-compromised sensor as long as the number of compromised nodes is less than a specific threshold [5]. They also proposed a location based key pre-distribution scheme [6]. In this scheme, nodes share more common keys when they are deployed in adjacent areas, but otherwise they share almost no common keys. This key pre-distribution makes the communication key establishment between neighbors easily succeed even if the number of pre-distributed keys in the sensors is small. Liu et al. proposed a communication key establishment scheme in which nodes are deployed in groups [7]. In this scheme, nodes belonging to the same group share many common keys. If any two adjacent nodes belong to different groups, then they act as a key establishment gateway which supports indirect key establishment between different groups.

Above schemes do not have any renewal mechanism in the key management. Therefore, administrative keys obtained from compromised nodes can be used for disclosing communication keys between the compromised nodes and their neighbors, and the administrative keys also exist inside many other nodes by a predetermined probability. As a result, the security of above schemes is deteriorated as the number of compromised nodes increases.

Recently, reactive key renewal schemes that renew the keys exposed to attackers using non-exposed keys are emerging. Eltoweissy et al. proposed EBS to protect a group key from compromised nodes [12]. In an EBS, the key distribution server randomly picks k keys from $k + m$ keys and distributes them to members. If a node is compromised, then the key distribution server renews the k compromised keys using the m keys unknown to the compromised node. In this way, the key distribution server can evict the compromised nodes from a communication group using only m messages. Eltoweissy et al. also applied the EBS to the key management in Wireless Sensor Networks (WSNs) in [10]. Sensors first determine the cluster to which they belong through the location information broadcasted by the sink. Each cluster maintains its own group key, and the EBS is applied to the whole network to renew these group keys. Jolly et al. presented a reactive key renewal scheme that does not rely on the EBS [8]. The sink pre-generates sensor-gateway keys and distributes them to sensors and gateways. Sensor-sink keys and gateway-sink keys are also distributed to them at this time. There are some sensors in each gateway's jurisdiction whose sensor-gateway keys are unknown to the gateway. Then, each gateway obtains the sensor-gateway keys through the communication with other gateways. While this scheme reduces the number of keys that each sensor should hold to two keys, it increases the communication overhead significantly. This is because a key renewal causes cluster reorganization and the redistribution of sensor-gateway keys. Key renewal schemes using the EBS are fragile to a collusion attack of adjacent nodes. To overcome this threat, Younis et al. proposed SHELL [11], which distributes the administrative keys with consideration of the locations of sensors. SHELL minimizes attackers' benefit from compromising adjacent sensors by making adjacent sensors share more common keys than distant sensors. Eltoweissy et al.

proposed LOCK (LOcalized Combinatorial Keying) which is an extension of SHELL [9]. LOCK applied the EBS to key renewal between CHs and the sink as well as to key renewal between sensors and the CH.

Above schemes have reactive renewal mechanisms in the key management. A CH is responsible for detecting a compromised node in its cluster and renews the administrative keys of the compromised node. Then, the CH renews the group key of the cluster using the renewed administrative keys. So, the reactive renewal mechanisms highly depend on a matured IDS (Intrusion Detection System). However, it is unrealistic that sensor networks have such a matured IDS. Besides, only one key that is referred as group key is employed for communication between a CH and its members so that one compromised node exposes the group key. Even worse, the CH role nodes are not changed with the lapse of time. Consequently, the CHs become targets of compromise attack and the increase of compromised CHs significantly impairs the security of network.

In a clustered sensor network, a CH election scheme is essential to transform a physical network into a cluster structure. Representative CH election schemes are LIDCP (Lowest ID Clustering Protocol) [14] and HCCP (Highest Connectivity Clustering Protocol) [14]. LIDCP prefers a lowest ID node among neighbors for CH elections while HCCP prefers a highest degree node among neighbors for CH elections. WCA (Weighted Clustering Algorithm) [15] considers various parameters for CH election, such as degree, transmission power, mobility, and residual energy. These parameters are assigned different weights, in line with the relative importance of the parameter in the network application. A final weight is generated by multiplying each parameter by the corresponding weight and summing them. The prominent problem of above weight based schemes is that a malicious node can broadcast a forged final weight as if it has a highest priority among neighbors. In that case, it can become a CH.

Recently, Sirivianos et al. proposed a different kind of CH election schemes, which elect a CH using an agreed random value and they are referred as SANE (Secure Aggregator Node Election) [16]. The authors presented three different schemes, that is Merkle's puzzle based scheme, a commitment based scheme, and a seed based scheme. In Merkle's puzzle based scheme, a current CH first establishes pairwise keys with its members. Then, a member generates its random value and encrypts it using the pairwise key with the current CH. It sums its encrypted random value with the accumulated sum which is sent from other node and forwards the sum to another node. This procedure is repeated until all nodes get the total sum of the encrypted random values. To decrypt this sum, each node should know all pairwise keys used for the generation of the sum due to the property of homomorphic encryption transformation [16]. For this purpose, the current CH distributes the pairwise keys to all nodes, and all nodes get the real sum of random values using the pairwise keys. They divide the real sum of random values by the number of nodes and get the remainder which indicates the position of the next CH node in the cluster. Because each node stores the IDs of nodes in an ascending order, they can easily reach an agreement on the CH election result. This conversion of an agreed random value to a CH position is also applied to two other schemes.

In the commitment based scheme, each sensor creates a random value and sends its encrypted random value to other sensors in the unicast manner. Then, each sensor sends the fulfillment value (that is, its random value) to other sensors. Receiving sensors verify the fulfillment values using the shared key, and sum them to make an agreed random value if the verification succeeds.

In the seed based scheme, each node generates its seed value and broadcasts it. This seed value is the initial random value for generation of sum of random values. Every CH election round, each node broadcasts its availability message. This availability message represents an intention for joining the CH election and it is similar to the fulfillment value of the commitment scheme. Sensors receiving the availability message keep the list of the senders. Then, all sensors make a sum of random values using the seed values of the senders and the number of CH election round. Merkle's puzzle based scheme causes a lot of overhead due to the pairwise key establishment, generation of sum of encrypted random values, and the pairwise key distribution. The commitment based scheme and the seed based scheme are vulnerable to transmission suppression and selective transmission of attackers. The transmission suppression causes arbitrary changes of CH election result. Besides, selective transmission causes the partition of clusters by separating one agreement of CH election into two or more agreements.

3. Network and Threat Model

3.1 Network Model

We assume that sensors are deployed by an aircraft without human intervention. To group all sensors into some clusters, we can employ a CH election scheme which is based on a weight value such as HCCP or LIDCP or WCA. However, a weight based scheme frequently makes a ripple effect where a CH election result in a region affects that of other regions. That is, in a weight based scheme, all nodes invoke the CH election process at the same time and some nodes make a subordinate relationship between them. Therefore, a weight based scheme necessarily requires the global synchronization among all nodes. This global synchronization requires a high communication and computation overhead. If the CH election is repeated periodically, this overhead increases heavily. To avoid this redundant overhead, we introduce the concept of sector, which plays a role of barrier between neighboring CH elections. Initially, because there are no sectors in the network, a CH election scheme should be invoked to make them. We assume that a weight based scheme is employed once for the sector formation. This sector formation makes one leader in each sector and the leader

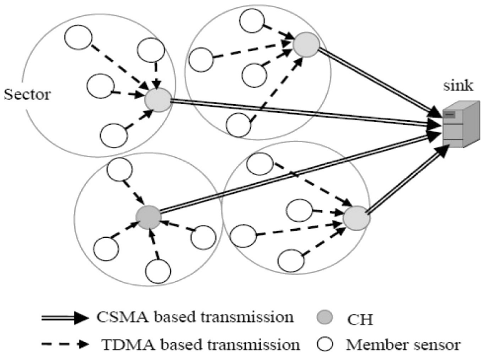


Fig. 1 Network model of clustered sensor networks.

is referred as sector manager hereafter. The sector manager helps a member establish a pairwise key with another member when the two members have no common keys. Section 4.1 provides the detailed information about the sector formation. Generally, sensors in a sector elect only one CH. In this case, a sector means a cluster. However, a sector sometimes has multiple CHs due to malicious behavior of attackers. In this case, a sector includes multiple clusters.

Our scheme employs a random value based scheme to elect a CH in a sector. The commitment based scheme and the seed based scheme described in Sect. 2 belong to the random value based scheme. The reason of using a random value based scheme is that it is more secure than the weight based schemes [16]. The network in this paper comprises a sink, some CHs, and many member sensors under the CHs. All sensors are quasi-stationary nodes and can play a role of CH. That is, the CH role sensors are changed with the lapse of time. Member sensors belong to only one CH and send their data to the CH nodes. The CHs aggregate data from member sensors and send the aggregated data to the sink using a fixed spreading code and Carrier Sense Multiple Access (CSMA). That is, a CH first sense the channel to see whether there is a transmission from a different CH or not. If the channel is occupied by any other transmission, it should wait to transmit its data. Otherwise, it sends its data to the sink using the sink spreading code. Note that each cluster employs a different spreading code for intra-cluster communication to minimize the inter-cluster interference.

To extend the lifetime of network, each member sensor sends its data in an allowed time slot and remains in a sleep state during the other time slots. To this end, each CH broadcasts a Time Division Multiple Access (TDMA) schedule for its members after settling its role, and sensors send their data to their CH directly.

The sink has a large amount of available resources and it is located in a sufficiently safe position to defeat various attacks. In contrast, CHs and member sensors have very limited resources and are located in positions where they can be compromised at anytime. Figure 1 shows the network model of the clustered sensor networks.

3.2 Threat Model

The aim of attackers is twofold. First, they aim to illegally obtain data that is going from the sensors to the sink. More importantly, they aim to cause a user to make a wrong decision by fabricating a large amount of data that is sent to the sink. To achieve these aims, attackers need to compromise sensors because all data and keys of the compromised sensors are revealed to the attackers. In a clustered sensor network, because CHs are the data collection points, smart attackers may target the CHs rather than member sensors for compromise. This is because they can get the control of the whole network by compromising a small number of CHs.

When there is a mixture of pure sensors and compromised sensors, we aim to minimize the illegal fabrication of data from the sensors by periodically renewing the CH role nodes. Actually, in most of sensor network applications, illegal fabrication of sensed data has a worse influence on the security than that of illegal acquisition. This is because the illegal fabrication of sensed data makes a wrong decision of a user while the illegal acquisition cannot do so. In other words, the aim of our key renewal scheme is to prevent compromised sensors from fabricating a large amount of data, even though they share their illegally obtained keys with each other. We also want to reduce the energy consumed for key renewal process.

Even though CH role nodes are changed periodically, the compromised nodes must try to become CHs by participating in the periodic CH elections. They may change the CH election result arbitrarily. If they keep changing the CH election result and there are many compromised nodes in the sector, a compromised node is likely to be elected as a CH. Also, the compromised nodes may produce several clusters in a sector. As the number of clusters in a sector increases, the number of members in a cluster decreases. Therefore, the transmission schedule of a cluster is shortened and the members in the cluster should send their data more frequently. Consequently, sensors deplete their energy in a short time. To materialize these attacks, attackers employ the following tricks. In a random value based scheme, all nodes cannot predict which node will become a CH except for one node which broadcasts its fulfillment value lastly. Therefore, a compromised node may delay it fulfillment value broadcast to recognize which node will become a CH. Then, it must suppress its fulfillment broadcast if a pure node is expected to become a CH. This action changes the CH election result. If a compromised node keeps changing the CH election results, the compromised nodes in the sector come to have many chances to become a CH. Especially, as the number of compromised nodes increases, they have more chances to achieve the goal. If a compromised node broadcasts its fulfillment value with a low transmission power, some nodes in the sector cannot receive the value. So, they have a different set of random values from other nodes which receive it. As a result, they make a different sum of the random values and elect a different node as their CH. In this case, one sector is separated into two clusters.

4. DIRECT Scheme

In this section, we describe our scheme, which is energy-efficient and resilient against the increase of compromised sensors. To begin with, we make the following assumptions.

- Each sensor is assigned a predefined number of administrative keys and an individual key before deployment. Each sensor can know which administrative keys assigned to other sensors if it knows their IDs because the key assignment employs the IDs of sensors. The administrative keys are used for each sensor to establish pairwise keys with other sensors within its two hop neighborhood. The individual key is employed for a CH role node to communicate with the sink.
- After the deployment, sensors invoke the sector formation step to group all sensors into some sectors, and then they establish pairwise keys with neighboring nodes which are at most two hops away. The sector formation and the pairwise key establishments are completed in a very short time so that an attacker cannot compromise a sensor within such a short period.
- The clocks of the sink and all sensors are initially synchronized. After the expiration of the synchronized timer, all sensors re-synchronize with other sensors within the same sector. Section 6 provides the detailed description about the synchronization problem.

Except for administrative key distribution, DIRECT consists of four steps; sector formation, pairwise key establishments within sectors, secure CH election, and transmission of sensed data. The former two steps are performed only once when the network boots. On the other hand, the other two steps are repeated periodically as long as the network is alive. Figure 2 shows the steps of DIRECT. Next subsections describe all steps of DIRECT respectively in detail.

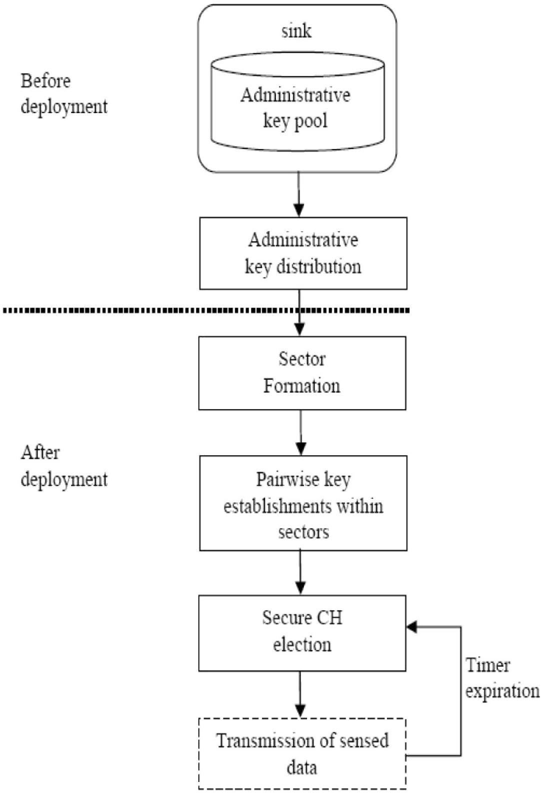


Fig. 2 Steps of DIRECT.

4.1 Sector Formation

At network boot-up time, each sensor exchanges its ID with neighbors. Then each sensor exchanges the neighbor list with its neighbors. Through these exchanges, each sensor recognizes other sensors which are at most two hops away and consequently recognizes their assigned keys.

After exchanging the ID and the neighbor list, sensors determine their sectors. In our scheme, HCCP [14] is used for the sector formation. A sensor compares its degree (number of neighbors) with its neighbors. If it is a highest degree node among neighbors, it becomes a sector manager and broadcasts the manager declaration message to its neighbors. The neighbors become the members of the sector and send a join message to the sector manager. Otherwise, it waits for a higher degree node to declare as a sector manager or join as a member to a different sector. Once a sensor joins a sector, it never joins other sectors even if it receives a manager declaration message from a different sector manager. Generally, HCCP creates some single sectors which consist of only one node. Because these single sectors weakens the advantages of grouping, they are incorporated into other sectors. After the sector formation, the sector managers register themselves into the sink. To reduce inter-sector interference, each sector communicates with the sink using direct-sequence spread spectrum (DSSS). Each sector employs a unique spreading code. Intra-sector communication is performed using the spreading code and the code is assigned when the sector manager registers itself into the sink. For instance, the first sector manager to register is assigned the first code on a predefined list, the second sector manager to register is assigned the second code, and so on.

4.2 Pairwise Key Establishments within Sectors

Because a CH is randomly elected in a sector and the CH and its members communicate directly, all sensors in a sector should share a pairwise key with each other. If any two sensors which are at most two hops away share at least one common administrative key, they can establish a pairwise key using the common key. However, some sensors do not share common administrative keys with each other. These sensors are considered as insecure sensors to each other. However, a sensor can indirectly establish a pairwise key with an insecure sensor using a helper node (that is, a node which shares common administrative keys with two sen-

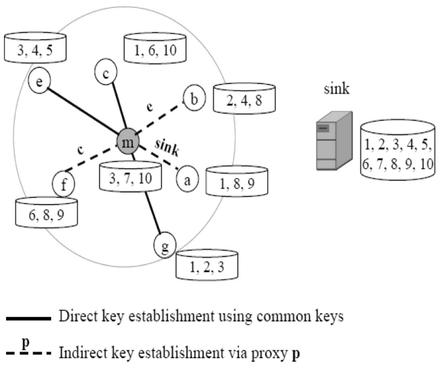


Fig. 3 Pairwise key establishment within a sector.

sors). However, if all sensors in a sector perform this indirect pairwise key establishment individually, it causes a lot of communication and computation overhead. To reduce this overhead, we use the sector manager. The reason why we use the sector manager is that it is located in the center position of the sector. The position enables any member in the sector to access the sector manager with minimum energy consumption. First, the sector manager establishes pairwise keys with its insecure members using its helper nodes. Note that each sector manager has many potential helper nodes through exchanges of the ID and the neighbor list. If all helper nodes do not share a common key with an insecure sensor, the sector manager establishes pairwise keys with the insecure sensor via the help of the sink. This is because the sink is the dealer of all administrative keys assigned to sensors in the network. Then the sector manager broadcasts the list of members so that each member establishes pairwise keys with its insecure members via the help of its sector manager. That is, if a member requests the sector manager to distribute a key to it and its insecure sensor, the sector manager generates a pairwise key and distributes it to the two nodes securely.

To facilitate the understanding of the pairwise key establishment, we introduce an illustrative example like Fig. 3. As shown in Fig. 3, the sector manager *m* has already established pairwise keys with members *c*, *e*, and *g* after exchanges of ID and neighbor list. However, the sector manager cannot establish pairwise keys with the sensors *b* and *f* because it shares no common administrative keys with them. In those cases, it establishes pairwise keys with *b* and *f* using helper nodes *e* and *c* respectively. That is, the helper nodes generates a pairwise key and distribute the key which is encrypted with common administrative keys to two nodes. In case of the sensor *a*, the sector manager has no common administrative keys and no helper nodes. So, the sector manager establishes a pairwise key with the sensor *a* via the help of sink.

4.3 Secure CH Election

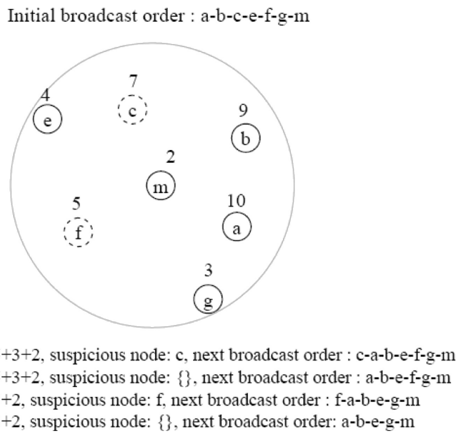
Before CH election, each sensor sets its timer interval to a predefined value. The timer interval is long enough to accommodate the CH election step and the data transmission step. Then, sensors should elect a node which plays the role of CH in this round. Our CH election scheme relies on an agreed random value like the commitment based scheme and the seed based scheme. So, a CH role node is likely to be changed at every election time. Owing to the periodic changes of CH role nodes, the communication keys between a CH and its members are periodically changed. For simplicity, we assume that there are no collisions in the MAC layers of sensors during the CH election. This assumption can be actualized using a broadcast order which is predetermined for broadcast of fulfillment values in each sector. Initial broadcast order in a sector is settled by the order of IDs of sensors. As shown in Sect. 3.2, the commitment based scheme and the seed based scheme enable a malicious node to change the CH election result. Our scheme prevents the arbitrary changes of CH elections result by forcing all sensors to follow the broadcast order of fulfillments. Besides, a malicious node can make some sensors have a different sum of random values by selectively transmitting its fulfillment value. A malicious node can easily implement the selective transmission by lowering the power level of its fulfillment transmission. Our scheme employs the received signal strength to defeat this selective transmission attack.

4.3.1 Commitment Broadcast

First, each sensor generates its random number and encrypts it with pairwise keys shared with other sensors in its sector to make commitments. The commitments are generated as many as the number of other sensors. Each sensor makes a list of the commitments in the order of IDs and broadcasts the list. After initial sector formation, distance between any two sensors in a sector is two hops and it is extended to at most four hops after the join of single sectors. Therefore, each sensor broadcasts the list with the power with which a message can reach four hop distance nodes. Sensors receiving the list first check whether the sender is a member in their sectors or not. If the sender is not the member, the receivers discard the message. Otherwise, the receivers pick up its commitment from the list and decrypt it to store with the sender's ID.

4.3.2 Broadcast of Fulfillment Value

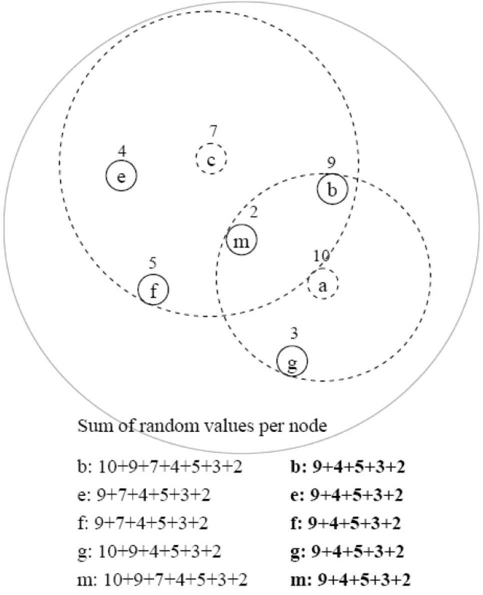
The commitments broadcasted by sensors can contribute to the generation of the sum of random values only if corresponding fulfillment values are received from the sensors. So, each sensor broadcasts the random number which was used for commitment generation to other sensors. Like the commitment broadcast, the transmission power level should be strong enough to reach the nodes which are at most four hops away. Besides, each sensor should follow the predetermined order for the broadcast of fulfillment values. If a sensor violates the order, the sensor is identified as a suspicious node and recorded in the suspicious node list. Besides, receivers discard the message from a suspicious node.



1. sum: 10+9+4+5+3+2, suspicious node: c, next broadcast order : c-a-b-e-f-g-m
2. sum: 10+9+4+5+3+2, suspicious node: {}, next broadcast order : a-b-e-f-g-m
3. sum: 10+9+4+3+2, suspicious node: f, next broadcast order : f-a-b-e-g-m
4. sum: 10+9+4+3+2, suspicious node: {}, next broadcast order: a-b-e-g-m

Fig. 4 Detection of malicious changes in CH election results.

If a fulfillment value is received in the accurate order, receiving sensors compare it with the corresponding commitment to check the equality. If they are equal, the receivers store the sender into normal node list. If a suspicious node violates the broadcast order again, the legitimate nodes exclude it from the member list and the suspicious node list. Therefore, a compromised node can arbitrarily change the CH election result only once by suppressing its fulfillment transmission. If it tries the same misbehavior or delays its transmission, other legitimate sensors eliminate it from the member list as well as discard the fulfillment value. This technique occasionally makes an innocent victim because the suspension of fulfillment value transmission may be caused by a message loss. However, in the aspect of secure CH election, this technique is a necessary evil. To help understanding the scheme, we introduce an example in Fig. 4. In Fig. 4, solid circles represent the legitimate nodes and dashed circles represent the compromised nodes (that is, *c* and *f*). Besides, the digits above the circles depict the fulfillment values of the nodes. In the first election, *c* delays its fulfillment transmission until all other sensors broadcast their fulfillment values. Because this action violates the broadcast order, other legitimate nodes record *c* into suspicious node list and ignores its fulfillment value. Nota that *c* is forced to broadcast its fulfillment value at the very first. In the second election, because *c* violates the order again, other legitimate nodes exclude *c* from its member list. As a result, *c* loses its right to join the CH elections in the sector. In the next two elections, *f* also loses its right to join the CH elections in the sector due to its consecutive violation against the broadcast order.



Sum of random values per node

b: 10+9+7+4+5+3+2	**b: 9+4+5+3+2**
e: 9+7+4+5+3+2	**e: 9+4+5+3+2**
f: 9+7+4+5+3+2	**f: 9+4+5+3+2**
g: 10+9+4+5+3+2	**g: 9+4+5+3+2**
m: 10+9+7+4+5+3+2	**m: 9+4+5+3+2**

Fig. 5 Detection of selective transmissions of fulfillment value.

Although a sensor transmits its fulfillment value, some distant sensors cannot receive it if the power level of the message is lower than a specific level. That is, we consider a node as an attacker trying to split the sector if its message cannot reach at most four hop nodes. Receivers can recognize this trial using the received signal strength. However, the received signal strength can be different at some sensors due to an obstacle or a propagation error. To deal with this problem, we set a specific level of signal strength to a threshold. The distance of the threshold ranges from three hops to four hops and it approximates to four hops. As a matter of course, this technique cannot perfectly prevent the sector split attack. However, we can limit the benefits of attackers significantly by using this technique. Note that attackers should transmit a message with a power level which reaches over three hop distance nodes. Otherwise, other legitimate nodes must discard the message. Therefore, the number of clusters which are abnormally generated in a sector is reduced. We assume that the energy model in [13] is employed in the energy consumption of transmitters and receivers. Assuming that the two-ray ground reflection model is used for radio propagation, a receiving node can calculate the transmission power of a sender transmitting a fulfillment value(P_t) by the Eq. (1), where P_r is the received power, d is the Euclidean distance, and L is the system loss. Besides, G_t and G_r are antenna gains and h_t and h_r are antenna heights.

$$P_t = \frac{P_r d^4 L}{G_t G_r h_t^2 h_r^2} \tag{1}$$

If a receiving node can know the transmission power of a sender, it can estimate the maximum reachable distance by the power(d_r) by Eq. (2).

$$d_r = \sqrt[4]{\frac{P_t}{E_{two\_ray\_amp} \times b}} \tag{2}$$

Here, $E_{two\_ray\_amp}$ is the energy consumed by the amplifier and b is the bandwidth of the channel. If d_r is smaller than a predetermined threshold, the receivers discard the received fulfillment value. To facilitate understanding the technique, we introduce another illustrative example in Fig. 5. In Fig. 5, the gray circle depicts the threshold of transmission range and a dashed big circle represents the transmission range of a compromised node. As shown in Fig. 5, the compromised nodes *a* and *c* transmit their fulfillment values with a low transmission range. In this case, the commitment based scheme and the seed based scheme divides the sum of random values into three kinds as shown in the left bottom of Fig. 5. However, in our scheme, the legitimate nodes discard the fulfillment values of *a* and *c* because their transmission range is shorter than the threshold. As a result, all legitimate nodes have the equal sum of random values in the sector as shown in the right bottom of Fig. 5.

4.3.3 Random Value Generation and CH Election

If a sensor receives fulfillment values (that is, random numbers) from all other sensors in the sector, it generates a sum of the random numbers and divide the sum by the number of normal nodes to get the remainder. Note that all sensors keep the list of normal nodes which follow the broadcast order of fulfillments values. The remainder means the position of the CH node in the normal node list.

4.3.4 Adjustment of Broadcast Order

After electing a CH among the normal nodes, each legitimate sensor adjusts the broadcast order of fulfillment values. First, each legitimate node moves the suspicious nodes to the front of the broadcast order and moves the normal nodes to the tail of the broadcast order. In other words, the broadcast order of the next round is generated by concatenating the suspicious node list and the normal node list.

The elected CHs generate a TDMA schedule and broadcast it. All members compute their transmission time and rest time in line with this schedule. They transmit their sensed data to the CH in their allowed time slots, and the CH transmits the aggregated data to the sink. This procedure is repeated until the timer which was set at the beginning of secure CH election step expires. If the timer expires, each sensor restarts the secure CH election.

5. Performance Evaluation

We built the simulation environment using the ns-2 simulator (version 2.27) [17] to evaluate the security and efficiency of DIRECT. In the simulation environment, 100 nodes were randomly distributed in the area of 100 meters×100 meters. The sink was situated in the position of (50 meters, 175 meters). The energy consumption model employed in the simulations adopted that of [13]. During the simulations, we executed three different schemes 30 times for each number of compromised nodes and we averaged the results to produce a representative value. In addition, we randomly selected the compromised nodes. Table 1 shows the parameters and their values employed in the simulations. In the EBS (Exclusion Basis System) [12] parameter, $k + m$ refers to the size of key pool in a cluster and k refers to the size of the key ring in a member sensor. We compared DIRECT with a scheme which has no renewal mechanisms and a reactive key renewal scheme. We selected Chan's scheme [4] as a representative of the schemes which have no renewal mechanisms and SHELL [11] as a representative of the reactive key renewal schemes.

Table 1 Simulation parameters.

Parameter	Value
Simulation time	3600 sec.
Initial energy	10 Joules/batery
Bandwidth	1 Mbps
Data packet size	500 bytes
Packet header size	25 bytes
Number of compromised nodes (CHs)	10∼50 (0∼3: SHELL)
Compromise time distribution	Random, 0∼900 sec.
Number of clusters	5 (SHELL)
EBS parameters ($k + m$)	7+3 (SHELL)
Key renewal period	20 sec. (SHELL)
Neighbor range	30 meters
Expiration time of timer	60, 120, 180 seconds (DIRECT)

In Chan's scheme, sensors employed the MTE (Minimum Transmission Energy) routing protocol [18] for delivery of their data to the sink. We made the routing protocol consider a security aspect as well. That is, a node first selects some candidates among neighbors that share a communication key with the node. Then the node determines the next hop node among the candidates by considering the energy consumption. If a node cannot find the next hop node that shares a communication key, it establishes a direct communication key with the sink by XOR'ing all pre-distributed keys. Because the sink is the distributor of all pre-distributed administrative keys, it can easily agree on the key with the node, and then the node sends the data to the sink directly.

To evaluate the security and efficiency, we introduced the following metrics:

- Exposure rate: the rate at which data from legal sensors are exposed to compromised nodes. This metric is used to measure the confidentiality of a key renewal scheme.
- Fabrication rate: the rate at which data is fabricated by compromised nodes. This metric is used to measure the integrity of a key renewal scheme.
- Energy consumption: amount of energy that the key renewal process consumes per unit time. This metric is used to measure the energy-efficiency of a key renewal scheme.
- Network lifetime: the time when the network is extinguished. In Chan's scheme, the network is extinguished when all nodes deplete their energy. In two other schemes, the network is extinguished when the number of active nodes is equal to the number of active CHs. This metric is used to show the impact of a key renewal scheme on the availability of the network.

5.1 Security Evaluation

Figure 6 shows the exposure rate of the sensed data as the number of compromised nodes increases. The increase in

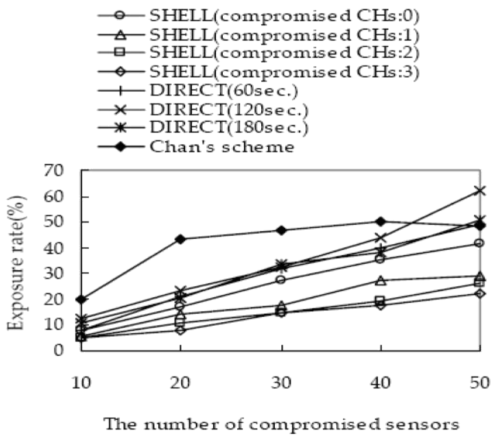


Fig. 6 Exposure rate vs. compromised sensors.

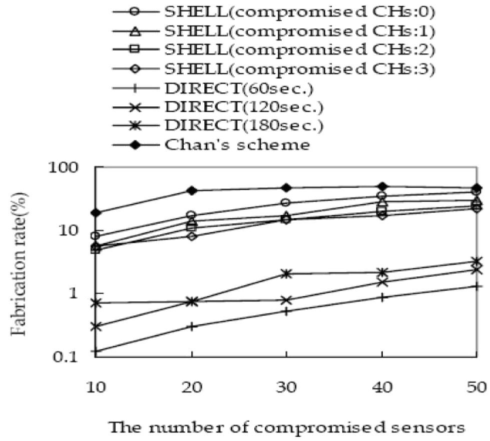


Fig. 7 Fabrication rate vs. compromised sensors.

the number of compromised nodes increases the exposure rate in all schemes.

In Fig. 6, SHELL seems to provide better confidentiality than two other schemes. However, the good confidentiality depends on the very good performance of an IDS (Intrusion Detection System) which operates in the network. In SHELL, compromised sensors are completely evicted from network by redistributing only non-compromised sensors to pure CHs. This action occurs whenever a compromised CH is detected by the sink. So, the increase of compromised CHs rather enhances the confidentiality as shown in Fig. 6. However, assuming the existence of such a wonderful IDS is unrealistic. If such an IDS exists and is available, our scheme can also benefit from its powerful function. If compromised sensors are all detected and they are known to all legitimate sensors, the legitimate sensors can easily expel them by joining a CH election without the compromised nodes. That is, in a CH election, sensors can exclude the compromised sensors by rejecting messages from them. In that case, our scheme must provide a better performance than SHELL because our scheme periodically invokes the CH election and expels the compromised nodes.

Figure 7 shows the fabrication rate of sensed data as the number of compromised nodes increases. Chan's scheme shows a much higher fabrication rate than two other schemes. This is because Chan's scheme delivers data from sensors to the sink through multiple intermediate nodes. That is, there are many nodes on the route from a sensor to the sink. Even if only one intermediate node is compromised, data from the sensor to the sink is fabricated by the compromised node. The fabrication rate increases with the increase in the number of compromised nodes. SHELL greatly decreases the fabrication rate compared to Chan's scheme. However, as the number of compromised nodes increases, the fabrication rate increases accordingly. This is because SHELL evicts the compromised sensors only when a compromised CH is detected. When a member sensor is compromised, SHELL just renews the keys. SHELL first renews the administrative keys known to the compromised nodes and then renews the group key using the renewed administrative keys. This eviction scheme is functionally useless, if all administrative keys employed in the cluster are exposed to attackers. Because the number of administrative keys employed for key renewals is small (10 keys), all of them are likely to be exposed to attackers under the increase of compromised nodes. In this case, attackers can keep fabricating the data of the compromised nodes.

In DIRECT, the number of clusters (CHs) is larger than SHELL and the compromised nodes are not evicted by an IDS. Because attackers randomly compromise the nodes, more CHs tend to be compromised. So, the fabrication rate in DIRECT highly depends on the number of compromised CHs. DIRECT reduces the probability that a compromised sensor is elected as a CH and rotates the CH role nodes. This makes the slope of the fabrication rate much gentler even if the number of compromised nodes increases.

5.2 Efficiency Evaluation

To evaluate the efficiency of the key renewal schemes, we first take energy-efficiency into consideration. This is because sensors deployed in the duty field are all battery-powered devices and there is no way to recharge them in the routine operation of the network. This fact forces all protocols on sensor networks to be energy efficient. Next, we investigate how the key renewal schemes influence the network lifetime.

Figure 8 shows the variation of energy consumption as the number of compromised nodes increases. In Chan's scheme, sensors do not engage in a key renewal process after they established communication keys with their neighbors. Therefore, sensors consume a constant amount of energy regardless of the increase in the number of compromised nodes.

On the other hand, SHELL evicts the compromised nodes and the eviction makes the variation in the energy consumption of sensors. In SHELL, sensor nodes consume more amounts of energy when the compromised nodes increase. This is because sensor nodes more frequently join the key renewal process. During the key renewal process,

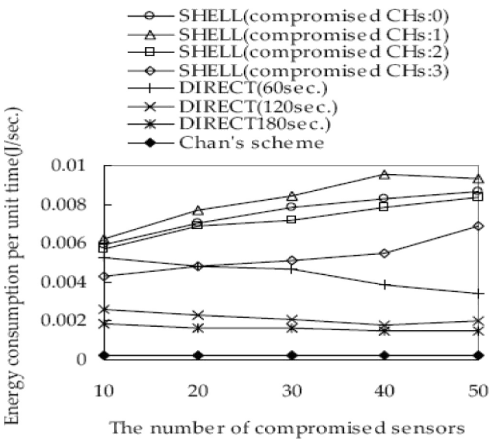


Fig. 8 Energy consumption vs. compromised sensors.

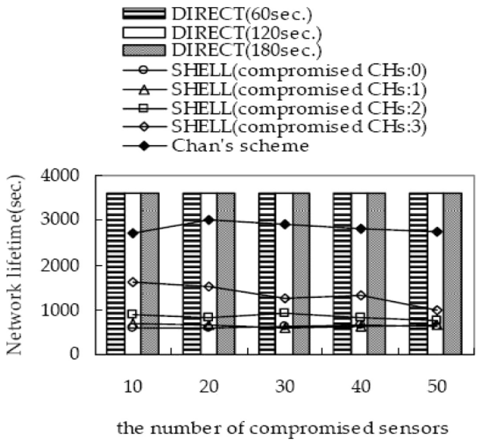


Fig. 9 Network lifetime vs. compromised sensors.

if a compromised CH is found, the compromised CH is expelled by the sink. Then, the orphan sensors are redistributed to the pure CHs, and the pure CHs distribute their administrative keys to the orphan sensors before key renewal. The preparation phase greatly increases the energy consumption of sensors. However, further increase of compromised CHs rather reduces the energy consumption of sensors. This is because the compromised nodes are expelled from the network whenever a compromised CH is found by an IDS. The expelled nodes do not consume energy any more for the key renewals.

On the other hand, main work for key refreshment in DIRECT is the secure CH election because it is periodically invoked. In the secure CH election procedure, all sensors consume their energy for two transmissions and two receptions. All other work such as sector formation and pairwise key establishments consume only a small amount of energy in the whole energy consumption perspective. This is because they are only invoked just one time at network bootup time. Therefore, our scheme consumes less energy for key renewals as shown in Fig. 8.

Figure 9 shows how the key renewal schemes affect the network lifetime as the number of compromised nodes increases. Chan's scheme renews no keys and evicts no nodes from the network, so there is little impact on the network lifetime as the number of compromised nodes increases. However, two other schemes evict the compromised nodes. SHELL evicts all compromised nodes when a compromised CH is detected and the number of evicted nodes is large. In DIRECT, one compromised node is evicted only when it violates the broadcast order of fulfillment values more than one time. Therefore, the rate of evicted nodes is much smaller than that of SHELL.

In SHELL, whenever the sink detects the compromised CHs, it evicts the compromised nodes as well as the compromised CHs from the network through cluster reorganization. Therefore, an increase in the number of compromised CHs causes the number of evicted nodes to increase as well. Nevertheless, the network lifetime lengthens as shown in Fig. 9. This is because the network lifetime is dominated not by the number of evicted nodes but by the number of remaining pure CHs. That is, if the number of compromised CHs is small (for example, one), then the sensors are redistributed to the remaining pure four CHs. In this case, all sensors are properly distributed to four remaining CHs so that the number of sensors served by a CH is small. This distribution makes the transmission schedule in a cluster short, and sensors frequently transmit their data to the CH. That is, because sensors consume large amounts of energy, the number of active nodes rapidly decreases. On the other hand, if only two pure CHs (that is, three compromised CHs) exists in a network, then the CHs should take charge of many sensors. This makes the transmission schedule in the clusters very long and the transmission frequency of the sensors decreases greatly. This slows the energy consumption of the sensors and the number of active nodes decreases very slowly.

As shown in Fig. 9, DIRECT significantly lengthens the network lifetime compared to SHELL in all timer intervals. There are two reasons why DIRECT extends the network lifetime significantly. First, DIRECT evicts only a small number of compromised nodes every CH election time. Second, DIRECT consumes much less energy for key renewals than SHELL as shown in Fig. 8.

5.3 Qualitative Comparison

Even though SHELL seems to provide the best confidentiality among three schemes, it is not the result of SHELL's function but the result of a matured IDS function. Furthermore, the integrity protection of SHELL deteriorates significantly as the number of compromised nodes increases. DIRECT effectively protects the integrity of data via the secure CH election while it reduces the energy consumption and lengthens the network lifetime. To provide more thorough comparison among the key management schemes, we list various properties of three schemes and compare them qualitatively in Table 2.

Table 2 Qualitative comparison of three key management schemes.

	Chan's scheme	SHELL	DIRECT
Key distribution	Pre-distribution before deployment	Redistribution as needed	Pre-distribution before deployment
Employment of cluster structure	N/A	Employ Large sized clusters	Employ Medium sized clusters
Key renewal type	No renewal	Reactive key renewal	Proactive key renewal
Key renewal method	N/A	EBS (Exclusion Basis System)	Secure CH election
Communication overhead	Overhead for initial key establishments	Overhead for reactive key renewals and cluster reorganization	Overhead for periodic CH election
Memory space overhead per node	Pre-distributed keys and generated pairwise keys	Fewer pre-distributed keys and and one group key	Pre-distributed keys, generated pairwise keys, and one individual key
Robustness against compromised nodes	Low	Very low (if the number of compromised nodes exceeds a threshold, key renewal is useless)	Medium
Robustness against compromised CHs	N/A	Low (CH role nodes are not changed)	High
Dependence on a matured IDS	N/A	Very high	N/A

6. Synchronization Issue

Generally, a periodic CH election in a network requires the time synchronization between sensors. As described before, a weight based CH election requires the global synchronization between sensors. However, DIRECT does not need the global synchronization because a CH election in a sector does not affect other sectors. Therefore, in DIRECT, only a local synchronization within a sector is needed. A number of local and global synchronization schemes have been proposed so far. Recently, Sun et al. propose TinySerSync [19] which implements a local synchronization scheme between any two sensors which share a pairwise key. Because any two sensors in a sector share a pairwise key with each other in DIRECT, they can synchronize with each other using the local synchronization scheme of TinySerSync.

7. Conclusion

In this paper, we have presented a key renewal scheme for wireless sensor networks, and it is based on periodic and secure CH election. For a secure CH election, our scheme employs two techniques: estimation of received signal strength and ordered transmission. Using these two techniques, our scheme prevents the compromised sensors from being elected as CHs. The simulation results show that our scheme remarkably enhances the integrity of the sensed data in the presence of compromised nodes as compared to other schemes. Other simulation results show that our scheme nevertheless consumes less energy for key renewals and provides a longer network lifetime than other schemes.

Our future research focuses on the design of a pairwise key establishment scheme between different generation sensors for enhancing the scalability of our scheme. Another interesting research item is to design a well-functioning IDS for clustered sensor networks.

Acknowledgements

This work was supported by the Korea Science and Engineering Foundation (KOSEF) grant funded by the Korea government (MEST) (No.2009-0083985).

References

[1] I.F. Akyidiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: A survey,” Comput. Netw., vol.38, no.4, pp.393–422, March 2003.

[2] S. Zhu, S. Setia, and S. Jajodia, “LEAP: Efficient security mechanisms for large-scale distributed sensor networks,” Proc. 10th ACM Conf. Computer and Comm. Security (CCS ’03), Oct. 2003.

[3] L. Eschenauer and V.D. Gilgor, “A key management scheme for distributed sensor networks,” Proc. 9th ACM Conf. Comp. and Comm. Sec., pp.41–47, Nov. 2002.

[4] H. Chan, A. Perrig, and D. Song, “Random key predistribution schemes for distributed sensor networks,” Proc. IEEE Symp. Security and Privacy, May 2003.

[5] W. Du, J. Deng, Y.S. Han, S. Chen, and P.K. Varshney, “A key management scheme for wireless sensor networks using deployment knowledge,” Proc. IEEE Infocom ’04, March 2004.

[6] W. Du, J. Deng, Y.S. Han, and P.K. Varshney, “A pairwise key predistribution scheme for wireless sensor networks,” Proc. 10th ACM Conf. Computer and Communication Security (CCS ’03), Oct. 2003.

[7] D. Liu, P. Ning, and W. Du, “Group-based key pre-distribution in wireless sensor networks,” Proc. 2005 ACM Wksp. Wireless Security (WiSe 2005), pp.11–20, Sept. 2005.

[8] G. Jolly, M.C. Kuscu, P. Kokate, and M. Younis, “A low-energy key management protocol for wireless sensor networks,” Proc. IEEE Int’l Symp. Comp. and Comm. (ISCC ’03), pp.335–340, June 2003.

[9] M. Eltoweissy, M. Moharrum, and R. Mukkamala, “Dynamic key management in sensor networks,” IEEE Commun. Mag., vol.44,

no.4, pp.122–130, April 2006.

[10] M. Eltoweissy, A. Wadaa, S. Olariu, and L. Wilson, "Group key management scheme for large-scale sensor networks," Ad Hoc Networks, vol.3, no.5, pp.668–688, Sept. 2005.

[11] M. Younis, K. Ghumman, and M. Eltoweissy, "Location-aware combinatorial key management scheme for clustered sensor networks," IEEE Trans. Parallel Distrib. Syst., vol.17, no.8, pp.865–882, Aug. 2006.

[12] M. Eltoweissy, M.H. Heydari, L. Morales, and I.H. Sudborough, "Combinatorial optimization of group key management," J. Network and Systems Management, vol.12, no.1, pp.33–50, March 2004.

[13] W. Heinzelman, A.P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," IEEE Trans. Wirel. Commun., vol.1, no.4, pp.660–670, Oct. 2002.

[14] M. Gerla and C. Chiang, "Multicluster, mobile, multimedia radio network," ACM-Baltzer J. Wireless Networks, vol.1, no.3, pp.255–265, 1995.

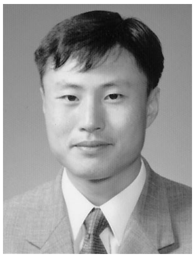
[15] M. Chatterjee, S. Das, and D. Turgut, "An on-demand weighted clustering algorithm (WCA) for ad hoc networks," Proc. IEEE GLOBECOM 2000, vol.3, pp.1697–1701, Nov. 2000.

[16] M. Sirivianos, M. Westhoff, F. Armknecht, and J. Girao, "Non-manipulable aggregator node election protocols for wireless sensor networks," Proc. Int'l Sympo. On Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt '07), pp.1–10, Cyprus, April 2007.

[17] The network simulator-ns-2, http://www.isi.edu/nsnam/ns/

[18] M. Ettus, "System capacity, latency, and power consumption in multihop-routed SS-CDMA wireless networks," Proc. Radio and Wireless Conf. (RAWCON), pp.55–58, Colorado Springs, Aug. 1998.

[19] K. Sun, R. Ning, C. Wang, A. Liu, and Y. Zhou, "TinySeRSync: Secure and resilient time synchronization in wireless sensor networks," Proc. 13th ACM Conf. on Computer and Communications, pp.264–277, Alexandria, VA, USA, Nov. 2006.



Gicheol Wang received the B.E. degree in Computer Science from Gwangju University, Gwangju, S. Korea in 1997 and the M.E. degree in Multimedia Engineering from Mokpo University, Mokpo, S. Korea in 2000. Currently, he is working toward the Ph.D. degree with the Department of Computer and Statistics Information, Chonbuk National University, Chonju, S. Korea. His current research interests include Ad-Hoc networks, sensor networks, distributed computing, security of wireless network, and mobile computing.

Kang-Suk Song received the B.S. degree from Kongju National University, Kongju, Korea, in 1999, and the M.S. degree from Kongju National University, Kongju, Korea, in 2002, in Electrical Engineering. He is currently serving as a senior researcher for Corebell Systems Inc. His current research interests include hierarchically image processing and context-awareness on Ubiquitous-Safety (U-Safety).

Gihwan Cho received the B.S. degree in Computer Science and Statistics from Chonnam University, Gwangju, Korea in 1985 and the M.S. degree in Computer Science and Statistics from Seoul National University, Seoul, Korea in 1987. He received Ph.D. degree in Computer Science from Newcastle upon Tyne, England in 1996. He worked for ETRI (Electronics and Telecommunications Research Institute), Daejeon, S. Korea, as a senior member of technical stuff, and from September 1997 to February 1999, for the Dept. of Computer Science at Mokpo National University, Mokpo, S. Korea, as a full time lacture. From March 1999, he joined to the Division of Electronic & Information Engineering at Chonbuk National University, Chonju, S. Korea, as an assosiate professor. His current research interests include mobile computing, computer communication, security of wireless networks, sensor networks, and distributed computing system.