Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 391986, 8 pages
http://dx.doi.org/10.1155/2013/391986

Research Article

Key Schemes for Security Enhanced TEEN Routing Protocol in Wireless Sensor Networks

Saewoom Lee,[1] Youngtae Noh,[2] and Kiseon Kim[1]

[1] *School of Information and Mechatronics, Department of Nanobio Materials and Electronics (DNE) World-Class University (WCU), Gwangju Institute of Science and Technology (GIST), 1 Oryong-dong, Buk-Gu, Gwangju 500-712, Republic of Korea*
[2] *Department of Computer Science, University of California, Los Angeles (UCLA), Los Angeles, CA 90095, USA*

Correspondence should be addressed to Youngtae Noh; ytnoh@cs.ucla.edu

Received 7 March 2013; Accepted 25 May 2013

Academic Editor: S. Khan

In wireless sensor networks (WSNs), hierarchical routing protocol is commonly used for energy efficiency. In particular, the TEEN (Threshold sensitive Energy Efficient sensor Network) protocol is used widely as a basic clustered multihop routing protocol. However, energy efficient routing protocols without proper security suffer from many security vulnerabilities. Hence, in this paper, we propose a hybrid key scheme specially for the TEEN protocol: a symmetric key scheme for the intracluster and a public key scheme for the intercluster. The simulation results show that network lifetime of the proposed hybrid key scheme decreases about 8% than the TEEN protocol and about 4% compared with the TEEN protocol with symmetric key scheme. On the other hand, a hybrid key scheme provides better probability of successful transmission than that of the symmetric key scheme.

1. Introduction

Wireless sensor networks (WSNs) can be applied to various applications such as safety monitoring of special spaces and buildings, traffic monitoring, environmental pollutant tracking, ocean and wildlife monitoring, home appliance management, and many military applications. In WSNs, one of the most significant constraints is the limited battery power of the nodes. Since the randomly deployed nodes are infeasible to be recharged. In this regard, it is worthwhile to pursue energy efficiency in WSNs. To acquire energy efficiency, various routing protocols are proposed in the literature such as SPIN, LEACH, and TEEN [1, 2]. Among the classified routing protocols as flat, hierarchical, and location-based routing protocol, the hierarchical routing protocol is proper in the view of energy efficiency due to the cluster head (CH) which performs data aggregation from non-CH nodes and directly communicates with the base station (BS). In particular, the TEEN (Threshold sensitive Energy Efficient sensor Network) protocol [3] is a basic routing protocol of hierarchical clustered multihop routing protocol. However, most routing protocols including the TEEN protocol in WSNs assume a trusted environment where all sensor nodes cooperate each other without any attacks. As a result, routing protocols suffer from many security vulnerabilities like as Denial of Service (DoS), injecting, and impersonating. Thus, an attacker can make the network useless [4, 5]. Therefore, designing a secure routing protocol is necessary for WSNs to provide secure data transmission regardless of opponent activities.

To cope with these problems, most security protocols are based on cryptographic operations that involve keys. Two types of key schemes are used in cryptography generally. The first one is the symmetric key scheme, which is computationally inexpensive, and can be used to achieve some of security goals. However, one major drawback with this scheme is the key exchange problem; that is, the two communication nodes must somehow know the shared key before communicating securely. Unfortunately, if an attacker can capture the symmetric key, the whole network can be broken because the attacker can decrypt every encrypted data by using the symmetric key [6, 7]. The other type of key scheme is public key scheme, which uses a pair of keys (p, q) where p means the public key and q indicates the private key corresponding to the public key p. Different from the symmetric key scheme, only the receiver can decrypt the encrypted data in the public key scheme [8]. So public key scheme allows for flexible key

management but requires a significant amount of computation due to the complex algorithm.

In this paper, we propose a hybrid key scheme that uses both symmetric and public key schemes in order to take the advantage of the rapid calculation times of the symmetric key scheme and flexible key management of the public key scheme. By considering a hybrid key scheme, we can offer security into the TEEN protocol. Usually, the encrypted data is transmitted from the sending node, and then it decrypts at the receiving node. In hybrid key scheme, two different key schemes are used depending on the types of communication for the TEEN protocol. That is, the symmetric key scheme is applied to intracluster communication, and the public key scheme is used for intercluster communication. Also, to provide the assurance of the identities among communication nodes, we create hashed value generated form the hash function. This hashed value is used to authenticate the origin of the messages as a message authentication code (MAC).

The rest of the paper is organized as follows. In Section 2, we briefly overview the TEEN protocol to explain the background for reactive routing protocol and related works. In Section 3, we describe the hybrid key scheme adapted to the TEEN protocol for the security of WSNs. In Section 4, we simulate a hybrid key scheme to evaluate the performance. We describe analysis of the security of the protocol in Section 5. Finally we conclude the paper in Section 6.

2. TEEN Protocol and Related Works

In this section, we briefly introduce the TEEN protocol and several routing protocols connected with the TEEN protocol for WSNs.

There are routing protocol groups based on their mode of functioning and the type of target application in WSNs: proactive and reactive routing protocols. In proactive routing protocol, once the cluster heads (CHs) are decided after cluster exchanging, the CH node creates a TDMA schedule and assigns each node a time slot when it can transmit. After setup phase, cluster members sense the phenomena and transmit the data to the CH. The CH aggregates this data and sends aggregated data to the higher level CH, or the BS depends on the network hierarchy. Low-Energy Adaptive Clustering Hierarchy (LEACH) [9] is a good example of a proactive routing protocol with some small differences.

On the other hand, the CH broadcasts in the following threshold values to its cluster members at every cluster setup phase in the TEEN protocol [3] which is the most typical protocol for reactive routing protocol.

Hard threshold (H_T): it is an absolute value for the sensed attribute. If the node senses this value, it turns on its transmitter and reports the data to the CH.

Soft threshold (S_T): it is a small variation in the value of the sensed attribute which causes the node to turn on its transmitter.

The nodes sense their surroundings continuously. The first time a sensed data reaches its hard threshold value, the node transmits the sensed data. The sensed value is stored to a variable called *sensed value* (SV). The node will transmit the data in current round only when both the following conditions are true:

(1) the sensed data is greater than the hard threshold,

(2) the sensed data differs from SV by an amount equal or greater than the soft threshold.

Thus, the hard threshold tries to reduce the number of transmission by sending only when the sensed data is in the range of interest. Also, the soft threshold reduces the number of transmission by excluding from the transmissions which have little or no change in the sensed data.

The TEEN protocol is succeeded by the APTEEN (Adaptive Periodic Threshold-sensitive Energy Efficient Sensor Network) [10] protocol which aims at capturing periodic data aggregations and respond to time-critical events. In APTEEN protocol, once the CHs are decided, the CH broadcasts four parameters: attributes, thresholds, TDMA schedule and count time. By using these parameters, the APTEEN protocol can provide access to periodic data as well as be informed of events of certain significance. Also, the THCHP (Two-level Hierarchical Clustering based Hybrid routing Protocol) [11] is expanded from the APTEEN protocol. The THCHP can be used for applications that require periodic data monitoring as well as warnings about critical events. The use of a two-level clustering hierarchy enables fixing the number of level 1 clusters K and optimizing the number of level 2 clusters so that the average sensor node energy dissipation is minimized.

Another modified version of the TEEN protocol is H-TEEN (Hierarchical Threshold sensitive Energy Efficient Network) [12] protocol where sensors self-organize into clusters and build a tree of transmissions, propagating data only to their parent in this tree. In H-TEEN protocol, it uses a constant number of hops to propagate the messages, having its cluster heads transmitting directly to the next level of the hierarchy. The H-TEEN protocol achieves high success rates in small area networks.

Kavitha and Viswanatha [13] have proposed Hybrid Reliable Routing (HRR) technique in wireless sensor networks. The HRR is intended to offer a hierarchical transmission environment by organizing randomly deployed sensor nodes into clusters efficiently. The remaining nodes can acquire the energy availability factor of the neighboring CHs. After that, they join that cluster which has more energy than other CHs, for that reason, ensuring service for a longer time. Once the CHs are identified, they generate a Dominating Set (DS). The members nodes of DS find least energy consumed multihop route to the sink. Meanwhile, graph theory can be used to generate the sensor clusters and help in identifying the CH.

As a basic routing protocol of hierarchical clustered multihop routing protocol, the TEEN protocol is succeeded by various routing protocols as we mentioned. However, all mentioned routing protocols are commonly focused on the fast data transmission with less energy consumption. Thus, they can be attacked from various security vulnerabilities to make the network useless. Therefore, the security for routing protocols including the TEEN protocol is necessary to provide secure data transmission.

3. Hybrid Key Scheme for the TEEN Protocol

In this section, we propose a hybrid key scheme adapted to the TEEN protocol.

Figure 1 introduces an example of hierarchical clustering routing protocols. Each cluster has a CH which aggregates data from cluster members. CH sends aggregated data to the BS or an upper level CH. These CHs, in turn, form a cluster with higher level CH as their CH. So some CHs can role as a second level CH. This action is repeated to form a hierarchy of clusters with the uppermost level cluster nodes for reporting directly to the BS. CHs at higher level in the hierarchical clustering need to send data over correspondingly larger distances.

As can be seen in Figure 1, there are three types of communication in the TEEN protocol: node-to-CH, CH-to-BS, and CH-to-CH communication. In node-to-CH communication case, a member node of the cluster tries to send the sensed data to the CH when it is satisfied the threshold values. On this occasion, a similar data can be transmitted by its neighbor nodes with high probability. In other words, if one similar data was damaged by an attacker, it can be restored by data of neighbor nodes. On the other hand, a CH, which has aggregated data of the cluster, sends whole cluster data to the BS or upper level CH. At this time, if a transmitted data was attacked by an attacker, a CH loses all information which sensed from its cluster or lower level cluster.

According to previous different types of communication, it needs different security method that depends on data integration to transmit the data securely. In node-to-CH communication case, even though sensed data which satisfies the thresholds can transmit, interested data can be detected by various nodes located around the event. Hence, a symmetric key scheme, which has simple algorithm and less calculation time, is adaptable for node-to-CH communication. On the other hand, in CH-to-BS and CH-to-CH communication cases, an aggregated data contains whole information of the cluster. If data modification or loss happens during communication, it is difficult to restore an information of the cluster. To prevent data loss or modification, a complex security algorithm is necessary, though a symmetric key scheme can give less calculation time. Therefore, a public key scheme, which has complex algorithm, is suitable for CH-to-BS and CH-to-CH communication. To sum up, it is appropriate to use different security method for different types of communication. That is, a hybrid key scheme is used for the TEEN routing protocol.

To apply the hybrid key scheme to the TEEN protocol, we append three procedures. One is addition of a symmetric key scheme inside the cluster, and the rest are addition of a public key scheme outside the cluster. Additionally, the last communication step is modified to use the hybrid key scheme during communication. To make secure keys for the hybrid key scheme, step of generating random numbers and exchanging those numbers are considered compared to the TEEN protocol. The next steps show the procedures of the hybrid key scheme for the TEEN protocol. Table 1 displays the notations used in this protocol description.

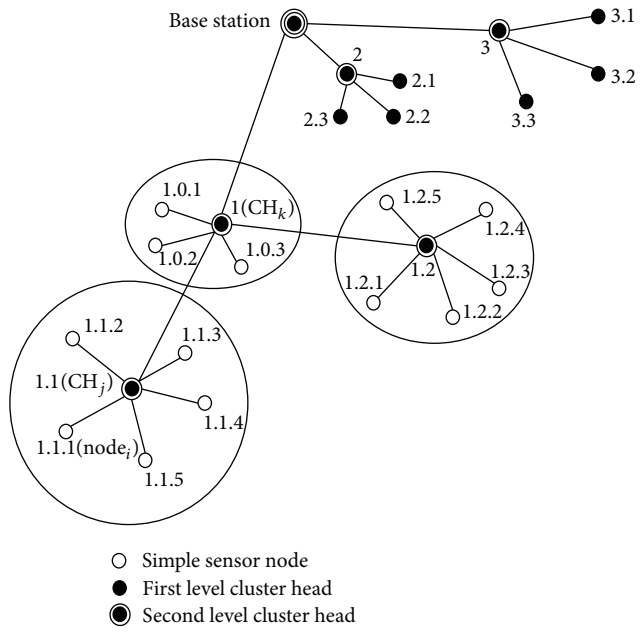

Figure 1: Hierarchical clustering routing protocol.

Table 1: Notations used in the protocol description.

Notation	Description
S	Generated random number before deployment
r_i	Random number generated at node i
$H(S)$	S is the data supplied to a hash function H
ID_j	Identification of node j
$A \parallel B$	Data A concatenated with data B
$E_K(A)$	Encrypt data A by using the key K
$D_K(C)$	Decrypt cipher C by using the key K
p_k	Public key used at node k (CH_k)
q_k	Private key corresponds to p_k at node k (CH_k)

Procedure 1. Figure 2 shows a process of symmetric key creation between node_i and CH_j. First of all, insert a random number S into every node before node deployment. When a process of node deployment is finished, every node generates their own random number such as r_i for node_i. After generating a random number, node_i and CH_j create or exchange the following messages to make a shared key:

(1) node_i and CH_j: $H(S) = k$,

(2) $\mathrm{node}_i \rightarrow \mathrm{CH}_j$: $E_k(\mathrm{ID}_i, r_i) \parallel \mathrm{ID}_i$, $H[E_k(\mathrm{ID}_i, r_i) \parallel \mathrm{ID}_i]$,

(3) $\mathrm{node}_i \leftarrow \mathrm{CH}_j$: $E_k(\mathrm{ID}_j, r_j) \parallel \mathrm{ID}_j$, $H[E_k(\mathrm{ID}_j, r_j) \parallel \mathrm{ID}_j]$,

(4) node_i and CH_j: $S^{r_i \times r_j}$.

In Message 1, each node calculates a hashed value k using a hash function and an inserted random number S. After CH selection and cluster setup steps, node_i sends the join signal to be a member of CH_j. At this time, node_i also sends

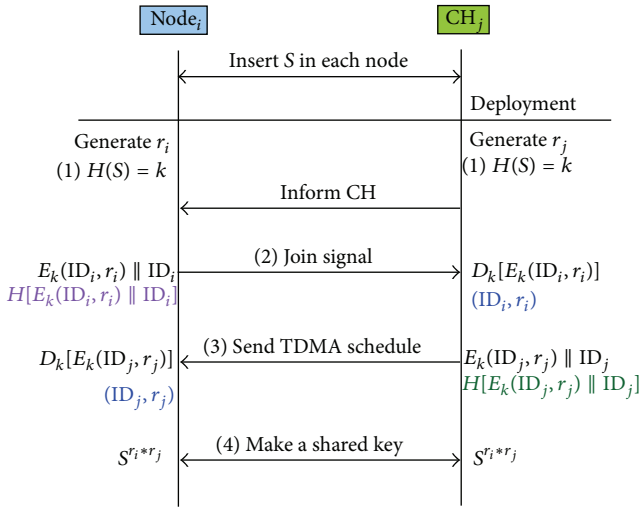

Figure 2: Process of symmetric key creation.

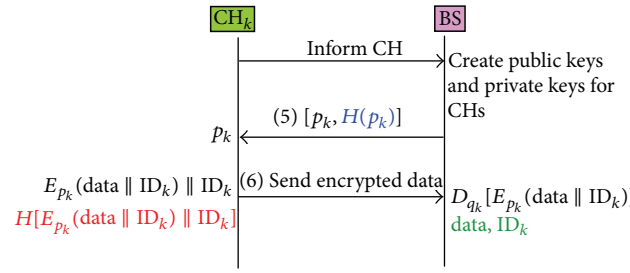

Figure 3: Process of public key creation.

$E_k(\mathrm{ID}_i, r_i) \parallel \mathrm{ID}_i$ and $H[E_k(\mathrm{ID}_i, r_i) \parallel \mathrm{ID}_i]$, hashed value of $E_k(\mathrm{ID}_i, r_i) \parallel \mathrm{ID}_i$, for message authentication such as Message 2. Concatenated identification of node$_i$(ID$_i$) is used for confirming the sender. Then a CH$_j$ calculates a hashed value using delivered $E_k(\mathrm{ID}_i, r_i) \parallel \mathrm{ID}_i$. If a calculated hash value is same as the received hash value, we can use the message received from node$_i$. Else we just discard the whole messages. Through a message authentication and decryption, CH$_j$ can get an ID$_i$ and a random number r_i. In Message 3, the CH$_j$ sends a TDMA schedule to node$_i$ as a member of CH$_j$ for removing the collision. Also, CH$_j$ sends $E_k(\mathrm{ID}_j, r_j) \parallel \mathrm{ID}_j$ and $H[E_k(\mathrm{ID}_j, r_j) \parallel \mathrm{ID}_j]$. If a received message is valid to node$_i$, it can get a r_j and an ID$_j$ after decryption. By using Messages 1, 2, and 3, they can exchange their own generated random numbers r_i and r_j, so, node$_i$ and CH$_j$ can create a symmetric key, $S^{r_i \times r_j}$, only for node$_i$-to-CH$_j$ communication. In this shared key, exchanged r_i and r_j are used for exponent part, and a S is used for base part to create the symmetric key.

Procedure 2. As can be seen, in the TEEN protocol, the role of the BS is just receiving data from the CHs. But, one additional duty is added to BS in the hybrid key scheme.

Figure 3 shows a process of public key creation between CH$_k$ and BS. They can make public key for CH$_j$ using the following messages:

(5) CH$_k$ ← BS: p_k, $H(p_k)$,

(6) CH$_k$ → BS: $E_{p_k}(\mathrm{data}, \mathrm{ID}_k) \parallel \mathrm{ID}_k$, $H[E_{p_k}(\mathrm{data}, \mathrm{ID}_k) \parallel \mathrm{ID}_k]$.

When the CHs inform their selection to the BS directly, the BS makes public and private key pairs. After creation the pairs, generated public key p_k for the CH$_k$ and its hashed value $H(p_k)$ are delivered to CH$_k$ as Message 5. CH$_k$ calculates a hashed value of p_k. If a calculated value and a received value are identical, p_k can be used as a public key for CH$_k$. When CH$_k$ sends aggregated data of a cluster to the BS, it uses a p_k for the data encryption. After encryption, CH$_k$ transmits encrypted data, $E_{p_k}(\mathrm{data}, \mathrm{ID}_k) \parallel \mathrm{ID}_k$, and a hashed value, $H[E_{p_k}(\mathrm{data}, \mathrm{ID}_k) \parallel \mathrm{ID}_k]$ as Message 6. The BS calculates hash value of encrypted data to compare received hash value with calculated hash value. If two values are identical, the BS decrypts an encrypted data by using a private key q_k corresponding to a public key p_k: $D_{q_k}(E_{p_k}(\mathrm{data}))$.

Procedure 3. Figure 4 shows a process of public key creation among CH$_j$, CH$_j$, and the BS. They can make public keys for CHs using the following messages:

(7) CH$_k$ ← BS: p_k, $H(p_k)$,

(8) CH$_k$ → BS: $E_{p_k}(\mathrm{data}, \mathrm{ID}_k) \parallel \mathrm{ID}_k$, $H[E_{p_k}(\mathrm{data}, \mathrm{ID}_k) \parallel \mathrm{ID}_k]$,

(9) CH$_k$ ← BS: p_k, $H(p_k)$,

(10) CH$_k$ → BS: $E_{p_k}(\mathrm{data}, \mathrm{ID}_k) \parallel \mathrm{ID}_k$, $H[E_{p_k}(\mathrm{data}, \mathrm{ID}_k) \parallel \mathrm{ID}_k]$.

When the CH$_j$ informs its selection to the BS through CH$_k$, CH$_k$ sends concatenated ID list, (CH$_j$ ∥ CH$_k$), to the BS. During the creation of public and private key pairs from the BS, it creates key pairs (p_j, q_j) for CH$_j$ and (p_k, q_k) for CH$_k$, respectively. After creation, LIST1 contains (CH$_k$, p_k) ∥ (CH$_j$, p_j). This LIST1 and a hashed value of LIST1, H(LIST1), delivered to CH$_k$ as Message 7. CH$_k$ calculates a hashed value of LIST1. If a calculated value and a received value are same, CH$_k$ extracts its public key p_k from the LIST1 for communication between CH$_k$ and the BS. On the other hand, CH$_k$ also sends LIST2, (CH$_j$, p_j), and a hashed value of LIST2, H(LIST2), to lower level CH$_j$ as Message 8. CH$_j$ tries to calculate a hashed value of LIST2. If the calculated value is identical as the received value from CH$_k$, CH$_j$ can use a public key p_j for communication between CH$_j$ and CH$_k$. When CH$_j$ transmits aggregated data to the BS through CH$_k$, an encrypted data, $E_{p_j}(\mathrm{data}, \mathrm{ID}_j) \parallel \mathrm{ID}_j$, and a hashed value, $H[E_{p_j}(\mathrm{data}, \mathrm{ID}_j) \parallel \mathrm{ID}_j]$, are transmitted to CH$_k$ as Message 9. After calculation and comparison between received and calculated values, CH$_k$ appends its ID, ID$_k$, end of encrypted data as $E_{p_j}(\mathrm{data}, \mathrm{ID}_j) \parallel \mathrm{ID}_j \parallel \mathrm{ID}_k$ and makes hashed value of modified data like as $H[E_{p_j}(\mathrm{data}, \mathrm{ID}_j) \parallel \mathrm{ID}_j \parallel \mathrm{ID}_k]$. When the BS receives these data from CH$_k$ as Message 10, the BS calculates hash value of encrypted data to compare a received hash value with a calculated hash value. If two values are identical, the BS decrypts an encrypted data by using a private key

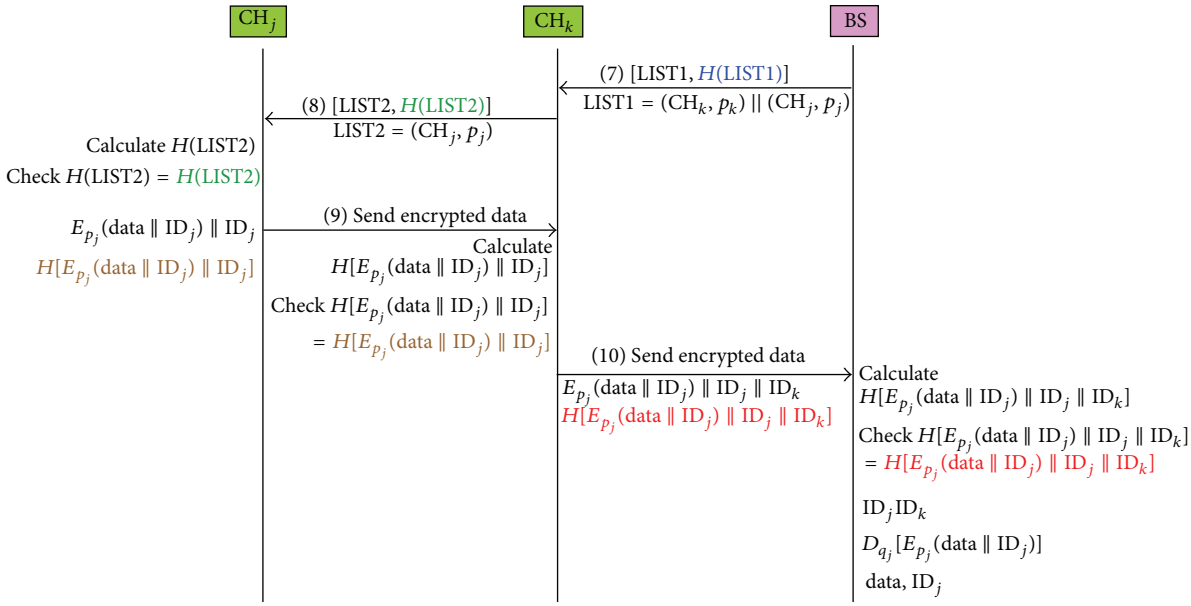

Figure 4: Process of public key creation.

q_j corresponding to a public key p_j, because ID$_j$ is attached previously than ID$_k$. In this procedure, when CH$_k$ tries to send its own aggregated data to the BS, it uses simply same step as Procedure 2.

After Procedures 1, 2, and 3, a hybrid key scheme, which uses a symmetric key scheme for intracluster communication and a public key scheme for intercluster communication, is added to the TEEN protocol for secure data transmission.

4. Performance Evaluation

In this section, we explain the simulation results performed on NS-2 [14] about an energy consumption and a probability of transmission. The purpose of the simulation is to evaluate the improvement about a probability of transmission over that of the symmetric key scheme, while energy consumption closes to symmetric key scheme's energy consumption.

For the comparison, we simulated the TEEN protocol on NS-2. The scenario of our simulation is as follows: 100 nodes distributed randomly in a 100 m × 100 m area. Each sensor has an initial energy of 2 J, and a sensing area is 10 m. The time duration of each round is 20 seconds. On the other hand, AES-128 and XTR-128 are used for symmetric and public key schemes, respectively, because energy consumption of AES is smallest among different symmetric key schemes: 3DES, RC5, Blowfish, and so forth. Besides, energy cost of XTR is also smallest among different public key schemes: DH, ECDH, ECDSA, and so forth [8]. Moreover, we use the SHA-1 for message authentication. The whole simulation parameters are listed in Table 2.

4.1. Network Lifetime. Figure 5 shows the comparison of network lifetimes for the TEEN protocol and the TEEN protocol with symmetric, public, and hybrid key schemes.

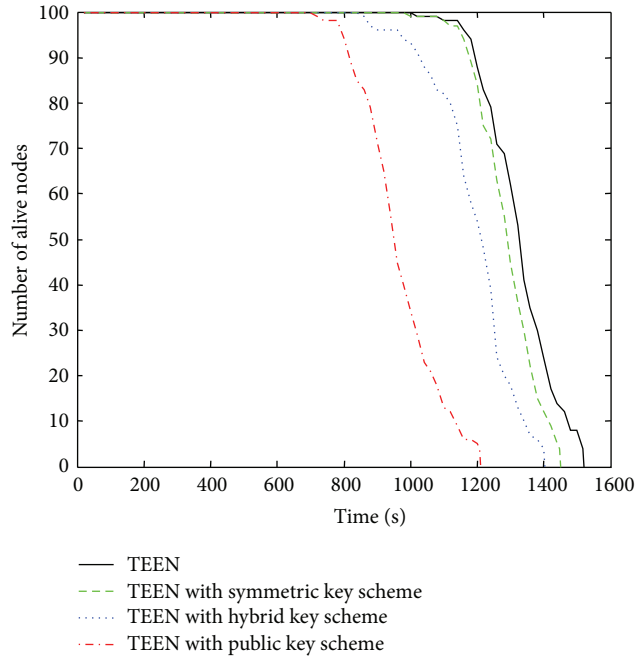

Figure 5: Number of alive nodes as a function of time, for the TEEN protocol without security and with symmetric, public, and hybrid key schemes.

As can be seen, when we use only a symmetric key scheme, network lifetime decreases about 5% compared with that of the TEEN protocol; on the other hand, network lifetime decreases about 21% when we use only the public key scheme. However, using the hybrid key scheme, network lifetime decreases about 8% compared with the TEEN protocol and about 4% compared with the TEEN protocol with symmetric

Table 2: Simulation parameters.

Monitoring area	100 m × 100 m
The number of nodes	100
Sensing range	10 m
Initial energy of each node	2 J
Symmetric key scheme	AES-128
Public key scheme	XTR-128
Message authentication	SHA-1
Symmetric key creation	0.00008 mJ
Encryption using symmetric key	0.0013 mJ
Encryption using public key	0.0021 mJ
TX power of encrypted data using symmetric key	0.0256 mJ
RX power of decrypted data using symmetric key	0.01792 mJ
TX power of encrypted data using public key	0.31 mJ
Energy consumption for authentication	5.9 μJ

key scheme. Those results show that the hybrid key scheme's energy dissipation is close to that of the symmetric key scheme's energy consumption.

4.2. Probability of Successful Transmission. Figures 6 and 7 show the probability of successful transmission in symmetric, public, and hybrid key schemes under the TEEN protocol. In these figures, r means the probability of the basic successful transmission for each node against attackers. When r is 0.99, as Figure 6, most of the transmission trials are successful, and three key schemes give a probability more than 0.97 on average. In addition, the gap of transmission probability among the three key schemes is small. On the other hand, when r drops to 0.9, as in Figure 7, the probability gap among three schemes is increased as the amount of communication trial is increased. The gap between a hybrid and the symmetric key schemes is wider than the gap between a hybrid and the public key schemes, especially. It means if r is decreasing, it is difficult to maintain security for systems by using only the symmetric key scheme. In this case, the average of probability is 0.899, 0.866, and 0.799 for the public, a hybrid, and the symmetric key schemes, respectively. As r drops from 0.99 to 0.9, a hybrid key scheme also decrease the probability. However, it is not far from probability of the public key scheme. Finally, Figures 6 and 7 show the fluctuating point with the network lifetime, because it generates different cluster topology every round.

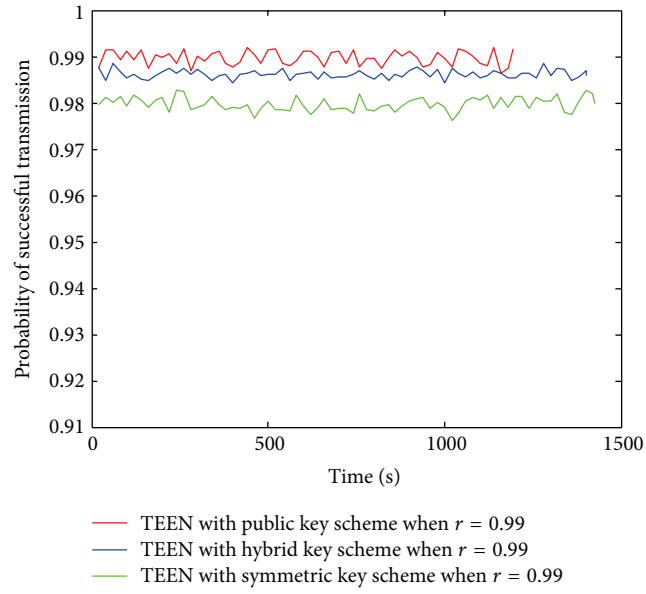

Figure 6: Probability of successful transmission as a function of time when r is 0.99.

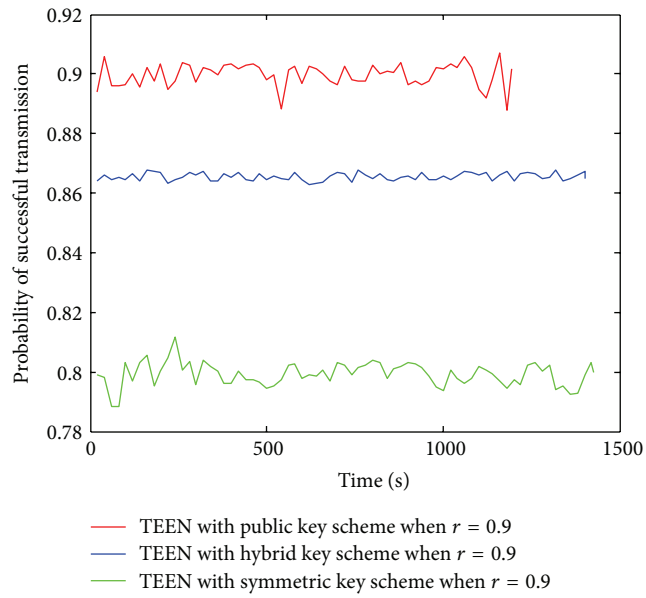

Figure 7: Probability of successful transmission as a function of time when r is 0.9.

5. Security Analysis

In this section, we analyze the security of a hybrid key scheme for the TEEN protocol. At the first part of this section, we analyze the security under several attacks which can happen for the TEEN protocol. The second part explains the security strength based on the key size.

5.1. Security Analysis. Hybrid key scheme is able to protect against typical attacks on wireless sensor networks. Various security vulnerabilities on WSNs are discussed in several literatures [5, 8]. Most attacks against WSNs routing protocols can be divided into one of the following categories: Sybil, manipulating routing information, and Hello flooding attacks. Hence, we discuss how a hybrid key can protect from various security vulnerabilities as follows.

5.1.1. Protecting from the Sybil Attack. The Sybil attack is an attack which a node sends multiple identities to other nodes. Authentication is used to verify the identity of the sender of a communication. So a node is hard to pretend to be another node. That is, when a node i transmits a sensed data to node j,

it can send a MAC (Message Authentication Code) computed using the shared key $S^{r_i \times r_j}$ between nodes i and j. Since the shared key is only known by two nodes due to their random numbers r_i and r_j, nobody can pretend to be node i or j. Therefore, a hybrid key scheme is able to protect from the Sybil attack.

5.1.2. Protecting from the Manipulating Routing Information Attack. In this protocol, the routing information is distributed by each CH for each cluster. When a CH_j sends a TDMA schedule to cluster members, it appends ID_j to each message and creates the hashed value for the message. Only cluster members can check the validation of the message due to the hashed value, thus an adversary is difficult to send inexact routing information.

5.1.3. Protecting from the Hello Flood Attack. In TEEN protocol, non-CH nodes decide the cluster to which they want to belong based on signal strength. It means that a powerful advertisement can make the malicious attacker be the CH. However, identification of each node is used to verify identities of neighbor. Thus a hybrid key scheme is able to prevent from the Hello flood attack.

5.2. Security Strength Analysis. In this part, we explain the security strength of hybrid key scheme by calculating the amount of time an attacker may need in order to break the key scheme. In general, MIPS (Million Instructions Per Second) is widely accepted as a unit to approximate computation that can be performed [15].

The security strength analysis is based on the key size of the key schemes. The shorter the key size is, the more vulnerable the encrypted data becomes to exhaustive key brute-force attack. In this hybrid key scheme, as changing the role of CHs at every 20 seconds, the symmetric key which used communication for CH-to-node is also updated at every 20 seconds for each cluster. So an attacker should finish the calculation within 20 seconds to break encrypted data. In fact, it would take about 5,300,000 years using a PC with 3 GHz to break an 80-bit symmetric cipher [16]. Assume that an attacker uses the Cray Jaguar which is wellknown as the fastest operational supercomputer with a sustained processing rate of 1.759 PFLOPS (Peta FLoating point Operations Per Second) in November 2009 [17]. To break a 128-bit encrypted data, it may take over than 2.4×10^{13} years. Using a 128-bit key with AES indicates that the encrypted data can be protected.

6. Conclusion

In WSNs, each sensor node has limited resources in many hostile and tactical scenarios and important commercial applications. So TEEN routing protocol is proposed for time critical applications and energy consumption. However, most routing protocols including TEEN protocol for WSNs do not include security at the designing stage. Hence, attackers can easily attack by exploiting vulnerabilities. Also, because a wireless channel is open to everyone, it can provide easy way for attackers to break into WSNs. Therefore, WSNs demand security to protect from attackers in the design of WSN protocols.

As security is becoming a major concern for WSNs protocol because of the wide security-critical applications of WSNs, several countermeasures have been proposed, such as authentication, identitying verification, and bidirectional link verification [5]. Alternatively, key establishment is the first step to establish a security, since all encryption-decryption and authentication methods use keys. Generally, two types of key schemes are used in cryptography: symmetric and public key schemes. The former is computationally inexpensive and needs small time for calculation, though it is difficult to manage the symmetric keys. On the other hand, even if the latter is much more expensive, it gives easier key management and resilient to node compromise than the former.

In this paper, we propose a hybrid key scheme that uses both symmetric and public key schemes in order to take the advantage of the rapid calculation times of the symmetric key scheme and flexible key management of the public key scheme, adapted to the TEEN protocol for WSNs to provide secure data transmission. Two key schemes are used depending on the types of communication for the TEEN protocol. Concretely, the symmetric key scheme is applied to intracluster communication, and the public key scheme is used for intercluster communication. Also, to provide the assurance of the identities among communication nodes, we make hashed value generated from the hash function. In this way, our scheme decreases about 8% compared with network lifetime of the TEEN protocol and about 4% compared with that of the TEEN protocol with symmetric key scheme. On the other hand, when r drops from 0.99 to 0.9, the probability of a hybrid key scheme is decreased; however, a hybrid key scheme provides better probability of successful transmission than that of the symmetric key scheme.

Acknowledgments

This work was partially supported by Leading Foreign Research Institute Recruitment Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science, and Technology (MEST) (K20902001632-10E0100-06010) and by the World-Class University Program funded by the Ministry of Education, Science, and Technology (MEST) through the National Research Foundation of Korea (R31-10026).

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.

[2] M. Yu, K. K. Leung, and A. Malvankar, "A dynamic clustering and energy efficient routing technique for sensor networks," *IEEE Transactions on Wireless Communications*, vol. 6, no. 8, pp. 3069–3079, 2007.

[3] M. Arati and P. A. Dharma, "TEEN: a routing protocol for enhanced efficiency in wireless sensor networks," in *Proceedings*

of the 15th International Parallel & Distributed Processing Symposium, pp. 2009–2015, San Francisco, Calif, USA, April 2001.

[4] E. Shi and A. Perrig, "Designing secure sensor networks," *IEEE Wireless Communications Magazine*, vol. 11, no. 6, pp. 38–43, 2004.

[5] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," in *IEEE Workshop on Sensor Network Protocols and Applications*, pp. 113–127, May 2003.

[6] W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz, and A. Khalili, "A pairwise key predistribution scheme for wireless sensor networks," *ACM Transactions on Information and System Security*, vol. 8, no. 2, pp. 228–258, 2005.

[7] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pp. 41–47, November 2002.

[8] D. W. Carman, P. S. Kruus, and B. J. Matt, "Constrainsts and approaches for distributed sensor network security," Tech. Rep., NAI Laboratories, 2000.

[9] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Annual Hawaii International Conference on System Siences (HICSS '00)*, January 2000.

[10] M. Arati and P. A. Dharma, "APTEEN: a hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks," in *Proceedings of the 16th International Parallel and Distributed Processing Symposium*, pp. 195–202, 2002.

[11] S. D. Muruganathan and A. O. Fapojuwo, "A hybrid routing protocol for wireless sensor networks based on a two-level clustering hierarchy with enhanced energy efficiency," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '08)*, pp. 2051–2056, April 2008.

[12] S. Nikoletseas, I. Chatzigiannakis, H. Euthimiou, A. Kinalis, A. Antoniou, and G. Mylonas, "Energy efficient protocols for sensing multiple events in smart dust networks," in *Proceedings of the 37th Annual Simulation Symposium, ANSS-37 2004*, pp. 15–24, April 2004.

[13] C. Kavitha and K. V. Viswanatha, "A hybrid reliable routing technique (HRR) for wirless sensor network," *International Journal of Computer Science and Network Security*, vol. 9, no. 3, pp. 35–39, 2009.

[14] The Network Simulator—ns-2, http://www.isi.edu/nsnam/ns/.

[15] P. Prasithsangaree and P. Krishnamurthy, "Analysis of tradeoffs between security strength and energy savings in security protocols for WLANs," in *Proceedings of the Vehicular Technology Conference*, vol. 7, pp. 5219–5223, September 2004.

[16] A. K. Lenstra and E. R. Verheul, "Selecting cryptographic key sizes," *Journal of Cryptology*, vol. 14, no. 4, pp. 255–293, 2001.

[17] TOP500 Supercomputing sites, http://www.top500.org/.