A Key Management Scheme for Cluster Based Wireless Sensor Networks

Reza Azarderakhsh, Arash Reyhani-Masoleh, and Zine-Eddine Abid
Department of Electrical and Computer Engineering
The University of Western Ontario, London, Ontario, Canada
E-mail: razarder@uwo.ca, {areyhani, zabid}@eng.uwo.ca

Abstract

Key Management is a major challenge to achieve security in wireless sensor networks. In most of the schemes presented for key management in wireless sensor networks, it is assumed that the sensor nodes have the same capability. The recent research has shown that the survivability of the network can be improved if sensor nodes are grouped in clusters in which a powerful cluster head assigned. However, to gain advantages of clustering in order to find an efficient key management scheme needs more research. In this paper, we investigate the key management in cluster-based wireless sensor networks using both private and public key cryptography. Our goal is to introduce a platform in which public key cryptography is used to establish a secure link between sensor nodes and gateways. Instead of preloading a large number of keys into the sensor nodes, each node requests a session key from the gateway to establish a secure link with its neighbors after clustering phase. The security analysis and performance evaluation show that the proposed scheme has significant saving in storage space, transmission overhead, and perfect resilience against node capture.

1. Introduction

Wireless Sensor Networks (WSNs) have recently attracted much attention because of their wide range of application, such as military, environmental monitoring, and heath care industry. Unlike wired and Mobile Ad hoc Networks, wireless sensor networks are infrastructure-less and can operate in any environment as compared to the traditional networks. Wireless sensor networks mainly consist of large number of tiny and simple nodes that are randomly deployed in operating areas unattended [1]. Some commercially available sensor nodes, such as Berkeley MICAz mote, include limited computational capability (8MHZ and 8-bit), with few memory (128KB programing memory) [15]. Security in WSNs has been receiving much attention in the literature. In most cases, the symmetric-key based key schemes have been presented [2, 3, 4, 7]. As shown in these schemes, the use of public key cryptography is not suitable for WSNs because it exceeds the computational and memory storage of the device. Note that none of the proposed schemes would provide the flexibility offered by the public-key-based solutions. Instead, symmetric-key based key management schemes are used [16].

In hierarchical WSNs, sensor nodes are clustered and a gateway or cluster head is allocated for each cluster. Gateway nodes are more powerful in computational capability, memory storage, life time, and communication range as compared to other nodes. In this paper, we propose a framework for key management in cluster based WSNs using a hybrid technique of public key and symmetric key cryptography. A symmetric key is assigned dynamically to sensor nodes to establish a secure link with their neighbors. A public key is pre-loaded to the sensor nodes and gateways for communicating with each other. Because gateway nodes are powerful, using Elliptic Curve Cryptography (ECC) [9] as a lightweight public key cryptography would not provide overhead in the network. Unlike the homogeneous WSNs where its key management scheme needs to pre-load high number of keys for individual nodes, it is not necessary for a sensor node in the clustered WSNs to keep all other nodes' key in its memory. As a result, the overall network communications and storage overhead will be reduced if the proposed scheme is used.

2. Previous Work

The first simple scheme [7] proposed for key management in WSNs assumes that the resource constraint nature of nodes in the homogeneous networks prevents them to use public key cryptography. Therefore, the key and key materials should be pre-loaded to each sensor node prior to deployments [7]. Eschenauer and Gligor (hereafter it is referred to as EG) first presented a basic random key pre-distribution scheme where each node is preloaded with some random keys from a large key pool [7]. To improve

978-0-7695-3492-3/08 $25.00 © 2008 IEEE
DOI 10.1109/EUC.2008.176

IEEE computer society

the network resiliency against node capture, Chan et al. [2] proposed a scheme called $q-\mathrm{composite}$. Many other improvements are proposed based on the mentioned schemes [16]. All key management schemes have mainly considered homogeneous and balanced deployment of nodes in WSNs. However, such networks have poor performance and scalability [16]. Many schemes designed for homogeneous sensor networks suffer from high computation and communication overheads. At present, there are only few key management schemes for heterogeneous WSNs. For instance, Du et al. [6] and Lu et al. [11] independently proposed a key management scheme for heterogeneous WSNs based on the random key pre-distribution and polynomial key predistribution scheme. The scheme presented in [6] pre-loads a large number of keys in each powerful sensor nodes (denoted as H-sensors) and very small number of keys in weak nodes (denoted as L-sensor) nodes based on probabilistic techniques. Lu et al. in [11] presents a framework based on polynomial key pre-distribution schemes for key management in peer-to-peer WSNs with heterogeneous sensor nodes.

Two main techniques for public key cryptography are RSA and ECC [16]. Traditionally until 2004, these schemes have been thought to be expensive, heavy weight and slow for WSNs. Recently, several researchers have improved public key cryptography in WSNs [14]. Implementing public key cryptography in WSNs is challenging. This is because sensor nodes have limited resources and spending too much processor's time and power on additional cryptographic computations is costly. By using ECC, it is shown in [14] that Public key Cryptography (PKC) is indeed feasible in WSNs. For a given security level, ECC demands considerably less resources than the conventional PKC [12]. ECC has been the top choice among various PKC options due to its fast computation, small key size, and compact signatures. PKC is also a reality in sensor nodes. There exist some ECC libraries for software implementations of sensor networks: TinyECC proposed by Liu and Ning [10] and WMECC proposed by Wang and Li [15]. Also it is possible to use Elliptic Curve Digital Signature Algorithm (ECDSA) without time synchronization for signing and verifying a message in the most common sensor nodes used nowadays, i.e.,MICAz [15]. Thus, it is desirable to explore the use of PKC on resource constrained sensor platforms [13].

3. Network Model

The system architecture for clustered WSNs is shown in Figure 1. The network includes the Base Station (BS), gateways, and sensor nodes. Gateways are less-energy-constrained and tamper resistant as compared to other nodes. Sensor nodes can communicate with each other if they are within a certain range. Communication between nodes is over a single channel and TDMA protocol can be used to provide MAC layer communication. Base station is assumed to be secure and trusted by all the nodes in the network. Moreover, it is assumed that sensor and gateway nodes are stationary and all nodes are assumed to be aware of their position information through the technique that is used in [8], [5].

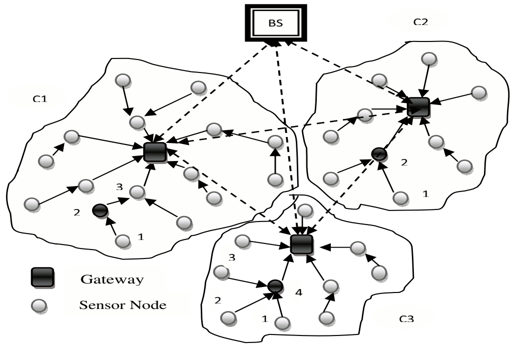

Figure 1. Clustered WSNs with multi-gateway and a Base Station (BS)

Network setup is performed in two phases; Network Bootstrapping and Clustering. In the bootstrapping stage, gateways discover the nodes that are located in their communication range. Gateways broadcast a message indicating the start of clustering. Each gateway starts the clustering at a different instance of time in order to avoid collisions. Sensor nodes also broadcast a message with their maximum transmission power indicating their location and energy reserved in this message. Each discovered node in this phase is included in a per-gateway range set. In the clustering phase, gateways calculate the cost of communications with each node in the range set. This information is then exchanged between all gateways. Upon receiving the information from all other gateways, each gateway starts clustering the sensor nodes, based on the communication cost and the current load on its cluster. When the clustering is over, all the sensors are informed about the ID of the cluster they belong to. Since gateways share the common information during clustering, each sensor node is picked by only one gateway [8]. For further information regarding the clustering scheme, refer to [8, 5].

3.1. Gateway Based Routing

In multihop clustered WSNs, a hierarchical structure is formed. An intra-cluster routing scheme is used to determine how to route packets from sensor nodes to the gateway in a multihop manner. Each sensor node sends a message, including its ID, its neighboring nodes, and its location information to the gateway. Gateways construct Least-Cost-Path (LCP) routing or Minimum Spanning Tree (MST) to reach sensor nodes. As the gateways are powerful and have sufficient memory resources, one broadcast is enough to cover all the sensor nodes in each cluster. However, in case of large number of nodes and huge clusters, gateways can broadcast in multiple times. On the other hand, broadcast from gateways must be authenticated with sensor nodes to prevent attacks and fake messages from adversaries. In this case, broadcast authentication can be provided with, for example, the ECDSA digital signature scheme [10], without requiring time synchronization. An alternate routing algorithm is needed to ensure reliable communication among sensor nodes in emergency conditions. Sensor nodes may find different parent nodes to reach the gateway so that they can reserve them for future routing algorithm in case of any changes.

3.2. Revisiting EG and $q-\mathrm{composite}$ Schemes

The basic scheme proposed in [7] can be revisited for the clustered WSN. Given the same generated key pool size of P, we store a key ring of size g in the gateways, and a key ring of size m, $g \gg m$ in each sensor node. The probability of a sensor node and its gateway to have at least one common key is [16]:

$$P[Match] = 1 - \frac{(P-m)!(P-g)!}{P!(P-g-m)!} \;. \qquad (1)$$

Recall that in the original $q-\mathrm{composite}$ scheme [2], each key ring is randomly drawn from a key pool. In such scheme, two nodes can establish a direct communication link if and only if they share q or more common keys and then the link is encrypted with the key obtained by hashing of common keys [2]. Thus an increase in q leads to a significant increase in the number of keys stored by all the sensors. In the clustered WSNs, it is possible to reduce the burden of the $q-\mathrm{composite}$ scheme in sensor nodes while retaining its security advantages. Note that the probability of a node and a gateway holding exactly i common key equals [6]:

$$p(i) = \frac{\binom{P}{i}\binom{P-i}{(g-i)+(m-i)}\binom{(g-i)+(m-i)}{m-i}}{\binom{P}{g}\binom{P}{m}} \;. \qquad (2)$$

Therefore, the probability that two nodes share at least q keys is: $P_c = 1 - \sum_{i=0}^{q-1} p(i)$. For a given minimum connectivity probability p, the largest key pool size can be computed such that $P_c \geq p$.

4. Proposed Key Management for Clustered WSNs

Typical packet size in WSNs are 30 bytes [1]. As compared to other PKC schemes, ECC has small message expansion for encryption and a little more power consumption. On most WSNs, transmitting a single bit costs as much power as executing 1000 instructions [15]. Small message size and low overhead are the main features of ECC. Therefore, instead of using symmetric key based on probabilistic approaches, we propose to use a public key management scheme based on ECC and Diffie-Hellman key exchange scheme.

Table 1. Notations for the proposed scheme

Notation	Definition
N	Number of nodes in the network
G	Number of clusters
$P^u_{n_i}$,$P^r_{n_i}$	Public & private key of node n_i, $0 \leq i \leq N-1$
$P^u_{G_j}$, $P^r_{G_j}$	Public & private key of gateway G_j, $0 \leq j \leq G-1$
P^u_{BS}	Public key for base station
$n_i^{'}$	Expected neighbors of node n_i
$n_i^{''}$	Neighbors of node n_i involved in routing
P_G	Total number of keys in each gateway
P_n	Total number of keys in each sensor
id_{G_j}	Unique id of gateway G_j
$K_{i,i'}$	Symmetric key between node i and node $i^{'}$ in a cluster

To generate the required pair of public and private keys for sensor nodes and gateways, a server based on ECC is used [15]. The proposed key management scheme is given in Table 2 which is explained below:

Key Pre-loading phase: The following two steps are assigned by server before deployment: **Step 1:** Each gateway G_j, $0 \leq j \leq G-1$ is assigned a unique id_{G_j} and is pre-loaded with the public key of all sensor nodes, its own public and private keys, i.e., $(P^u_{G_j}, P^r_{G_j})$, and the public key of base station (P^u_{BS}).

Step 2: The sensor node n_i, is assigned a unique id_{n_i} with its private and public keys $(P^r_{n_i}, P^u_{n_i})$ and public key of all the gateways $(P^u_{G_j},\ 0 \leq j \leq G-1)$ in the network.

Broadcast authentication: Since all nodes know the public key of their gateway G_j, broadcast authentication of gateways by its sensor nodes using a low cost ECDSA signature verification can be done. This digital signature can be verified by sensor nodes knowing the public keys of gateway nodes. **Step 1:** The gateway G_j broadcasts message B_j obtained by encrypting the concatenation of its public key $P^u_{G_j}$ and id_{G_j} using it's private key $P^r_{G_j}$, i.e., $B_j = E_{P^r_{G_j}}(id_{G_j} \parallel P^u_{G_j})$. **Step 2:** Each node n_i in the cluster j authenticates the received message B_j by decrypting it using the public key of gateway G_j, i.e., n_i verifies: $D_{P^u_{G_j}}(B_j)$.

Key distribution: Having all the other nodes public key in the network is not beneficial due to storage overhead in nodes. **Step 1:** Therefore, a sensor node has to send a request all the way to the gateway node in its entire cluster, which includes id_{n_i}, location of n_i, and list of its neighbors ($n_i^{'}$). A sensor node may send the session-key request message with the LCP routing algorithm to the gateway. **Step 2:** Upon receiving the session key request by the gateway, the gateway will send the ECC encrypted pairwise key ($K_{i,i'}^{'}$) between the node i and its neighbor node i', by using the public key of n_i, i.e., $K_{i,i'}^{'} = E_{P_{n_i}^u}(K_{i,i'})$. As the sensor nodes need to establish a communication link related to its routing algorithm, it needs the session key between itself and at least $n_i^{''}$ other neighbor nodes ($n_i^{'} > n_i^{''} \geq 2$), where $n_i^{''}$ is the number of neighbors of node *i* involved in its routing algorithm. As an example, in Figure 1, node 2 in C1 and C2 needs at least two session keys of neighbor nodes, while node 4 in C3, needs at least the session keys of its three neighbor nodes.

Now, two new schemes can be utilized: (i) A sensor node asks for the public keys of its neighbors from the gateway and the gateway may send the other neighbor's public key to a sensor node in a plain-text. Note that, the idea of asking for public key's of other neighbors is not feasible because it will increase the communication overhead. (ii) The sensor node n_i sends symmetric-key request to the gateway, upon receiving the request, gateway generates a random key $K_{i,i'}$, which is for symmetric cryptography algorithm, e.g., SKIPJACK. Then, the gateway will encrypt the session-key shared between node n_i and its requested neighbors with ECC, using public key ($P_{n_i}^u$) and then unicast it to the node n_i. **Step 3:** Each sensor node decrypts the message received from the gateway with its own private key ($P_{n_i}^r$) and gets the symmetric key shared with the other neighbor nodes, then it can establish secure communication with all (needed) neighbor nodes. After a certain time all the sensor nodes in each cluster will obtain a session key which is shared with their neighbors.

Table 2. Key management algorithm

Before Deployment: Key Pre-loading
Step 1. $Server \rightarrow G_j$**:**
$id_{G_j} \parallel (P_{n_i}^u) \parallel (P_{G_j}^u, P_{G_j}^r) \parallel (P_{BS}^u)$
Step 2. $Server \rightarrow n_i$
$id_{n_i} \parallel (P_{n_i}^u, P_{n_i}^r) \parallel (P_G^u)$
Broadcast Authentication
Step 1. $G_j \rightarrow n_i$**(gateway** G_j **broadcasts to all nodes in its cluster)**
$B_j = E_{P_{G_j}^r}(id_{G_j} \parallel P_{G_j}^u)$
Step 2. $n_i\ verifies$**:** $D_{P_{G_j}^u}(B_j)$
After Deployment: Key distribution
Step 1. $n_i \rightarrow G_j$ **(All nodes in the cluster** J **broadcasts to gateway** G_j**)**
$id_{n_i} \parallel nonce \parallel\ Location\ of\ n_i \parallel List\ (n_i')$
Step 2. $G_j \rightarrow n_i$ **(unicast):** $K_{i,i'}^{'} = E_{P_{n_i}^u}(K_{i,i'})$
Step 3. $n_i\ calculates$**:** $D_{P_{n_i}^r}(K_{i,i'}^{'}) = K_{i,i'}$

5. Performance Evaluation

In this section, we evaluate the performance of our proposed scheme. Experiments are performed on our own simulator written in C++. In our simulations, we use $N = 1000$ nodes deployed in a $A = 1000 \times 1000\ m^2$ with $G = 10$ clusters and gateways. Transmission radius is set up to $d = 100m$ for each node and the packet size is assigned 40 bytes for data packets.

5.1. Storage Saving

Assume that the network contains N nodes and the number of gateways are G, the optimal number of gateways can be achieved by calculating average energy consumption and load balanced clustering [8] as explained above. Each gateway, say G_j, should be pre-loaded $P_G = N + 4$, $\{(P_{n_i}^u, 0 \leq i \leq N-1), (P_{G_j}^u, P_{G_j}^r), P_{nn},\ P_{BS}^u\}$. The sensor node n_i in the network should be pre-loaded with $(P_{n_i}^r, P_{n_i}^u)$ and public key of all the gateways in the network ($P_{G_j}^u, 0 \leq j \leq G-1$), which is embedded to the nodes before deployment. Therefore, the number of total keys that should be pre-loaded in each sensor node is $2 + G$.

After the bootstrapping and clustering, each sensor node receives a session key shared between its neighbors. Therefore the number of total keys in each sensor node will be $P_n = 2 + G + n_i^{''}$, with $2 \leq n_i^{''}\ <\ n_i^{'}$. The approximate number of expected neighbors of each nodes can be calculated knowing the network density and transmission range of sensor nodes [1]. Compared to the basic key management scheme, the number of overall keys stored in the network can be significantly reduced in the proposed scheme. As an example: Assume the number of nodes is $N = 1000$, and with optimal clustering, we can achieve a minimum number of gateways of $G = 10$ [8]. Thus we have 10 clusters in the network with the number of sensors being dependent on load balanced clustering. The total number of keys preloaded in each gateway would be 1004.

Assuming that the transmission range of each node is $d = 100m$. Then, the average number of neighbors for each sensor node will be [1]:

$$n_i' = \frac{N\pi d^2}{A} \approx 31. \qquad (3)$$

Therefore, the number of total keys in each sensor node will be $P_n = 43$, to achieve complete connected network. As shown in (1) for a basic scheme where the P is the key pool size, we can say that there is almost a 95% chance, that two nodes will have a single shared key with $P = 5,000$, and a chain of 120 keys. While 328 keys are needed to have five-nine (0.99999) connectivity. In case of clustered sensor nodes with the EG scheme (EGC), it needs $g = 750$ keys for a gateway and $m = 50$ for each node to have the five-nine connectivity. However, for our proposed scheme with clustered sensor nodes, it requires roughly one third of storage needed for the clustered EG basic scheme.

5.2. Transmission Overhead

In the proposed scheme, each sensor node needs at least two other neighbors to establish a link to reach the gateway. All previous work, including this paper has assumed that all the nodes within a neighborhood are within transmission range of each other. We will evaluate the impact of having smaller radius than neighborhood radius. As it is shown above, the total number of neighbors in a typical network is about 31 nodes. We now vary the transmission range of a node, d between 10% and 70% of the neighborhood radius. We set the number of session key to each sensor node to have five-nine connectivity in a neighborhood of 31 nodes which all are within transmission range. The basic EG scheme, requires 50 keys/node [7]. Since each individual node does not need to connect to the gateway directly, it is possible to have a significant saving in the transmission and in the energy consumption of the network.
From [8], let α_t and α_r be energy/bit consumed by the transmitter and receiver nodes, respectively. Then the consumed energy in transmitter (E_{Tx}) and receiver (E_{Rx}) to send r bits will be:

$$E_{Tx} = (\alpha_t + \alpha_{\mathrm{amp}}.d^2) \times r \tag{4}$$

and,

$$E_{Rx} = \alpha_r \times r,$$

Where α_{amp} is energy dissipated in the transmitter, and the path loss estimated as $1/d^2$. The simulation results are shown in Figure 2. It indicates that with a small transmission radius, the number of the nodes connected to their neighbors are small. For the EGC scheme, which is the clustered form of EG scheme, the probability of a gateway being within range of a node is small. Since the sensor nodes have small number of keys in this case, there is a very low probability that they can securely connect to the network. As the transmission radius grows from 0.1 to 0.3 (horizontal axis) in Figure 2, our scheme achieves about 25 nodes (vertical axis) and the EGC and basic the EG scheme reach 22 and 17 neighbor nodes, respectively. The fact remains that in our scheme, reducing the transmission radius (to 0.5) achieves 27 neighbor nodes.

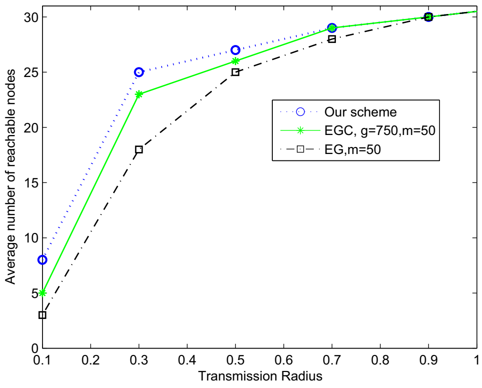

Figure 2. The effect of changing the transmission radius, d of randomly positioned nodes

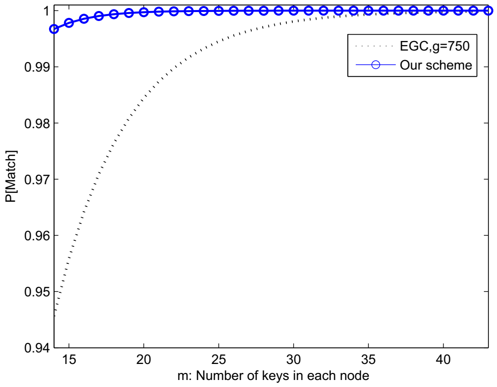

Figure 3. The probability of connectivity

5.3. Security Analysis

We assume that the gateways in the proposed scheme are tamper proof. Each node is pre-loaded with a unique private key and a unique public key. After bootstrapping phase, and during the clustering phase, each node will get a session key. As the session key is pairwise, sensor nodes are able to verify the identities of the nodes to which they are communicating. An adversary is unable to impersonate the identity of any node except by capturing it, so it can support the node-to-node authentication. Obviously capturing node n_i, will jeopardize just $(P^r_{n_i}, P^u_{n_i})$ and the pubic key of all gateways and reveals no information about the links that is not directly involved in communications with compromised node n_i.

As the number of compromised nodes increases both the basic EG scheme and the$q - \mathrm{composite}$ scheme expose more fraction of network to adversary. Since more keys are pre-loaded in sensor nodes of the basic EG scheme, the adversary may utilize more information about other links in network. In our proposed scheme each sensor node preloaded with a unique private key before deployment, and after deployment each node includes different number of symmetric shared keys. Therefore, capturing a node does not effect to the security of the rest nodes, and the best resiliency against node capture can be obtained. So the fraction of the network that can be compromised is always zero.

The results illustrated in Figure 3 explain the need to understand the neighborhood size in terms of transmission range and connectivity probability. For achieving the target connectivity, adequate number of keys are needed to deploy in each sensor node. With decreasing transmission radius to 0.3 in order to reach a 22 nodes in the EGC scheme as shown in Figure 2, it will obtain 0.99 connectivity (see Figure 3). While our scheme obtains five-nine connectivity with having 25 nodes (i.e. 25 session keys should be requested by a node n_i to obtain 0.99999 connectivity) in its neighborhood. As a results, significant energy saving can be obtained by reducing the transmission range while having the same secure connectivity.

6. Conclusions and Future Work

In this paper, we have presented a key management scheme for clustered WSNs which uses both public and symmetric key cryptography. We have used public key cryptography for giving a session key to a sensor node which has already requested for a session key through its potential neighbors. Our proposed scheme not only does reduce the transmission range and hence power consumption but also reduces the risk of single or multiple node capture. Significant storage saving can also be obtained as compared to the previous schemes. Key revocation for the proposed scheme will be considered in our future research.

Acknowledgments

This research was supported by Natural Sciences and Engineering Council of Canada through a research grant to A. Reyhani-Masoleh.

References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422, 2002.

[2] H. Chan, A. Perrig, and D. X. Song. Random key predistribution schemes for sensor networks. In *Proceedings of IEEE Symposium on Security and Privacy*, pages 197–213, Washington, DC, USA, 2003. IEEE Computer Society.

[3] R. Di Pietro, L. Mancini, A. Mei, and Panconesi. Connectivity properties of secure WSN. In *Proceedings of 2nd ACM workshop on Security of ad hoc and sensor networks*, pages 53–58. ACM Press, 2004.

[4] W. Du, J. Deng, Y. Han, P. Varshney, J. Katz, and A. Khalili. A pairwise key predistribution scheme for wireless sensor networks. *ACM Transactions on Information and System Security (TISSEC)*, 8(2):228–258, 2005.

[5] X. Du and Y. Xiao. Energy efficient chessboard clustering and routing in heterogeneous sensor networks. *Journal of Wireless and Mobile Computing*, 1(2):121–130, 2006.

[6] X. Du, Y. Xiao, and Guizani. An effective key management scheme for heterogeneous sensor networks. *Ad Hoc Networks*, 5(1):24–34, 2007.

[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, CCS '02*, pages 41–47, New York, NY, USA, 2002. ACM.

[8] G. Gupta and M. Younis. Load-balanced clustering of wireless sensor networks. In *Proceedings of IEEE International Conference on Communications, ICC'03*, 2003.

[9] N. Koblitz. Elliptic curve cryptosystems. *Mathematics of Computation*, 48(177):203–209, 1987.

[10] A. Liu, P. Kampanakis, and P. Ning. Tinyecc: Elliptic curve cryptography for sensor networks (version 0.3), 2007, available at http://discovery.csc.ncsu.edu/software/TinyECC.

[11] K. Lu, Y. Qian, M. Guizani, and H. Chen. A framework for a distributed key management scheme in heterogeneous wireless sensor networks. *IEEE Transactions on Wireless Communications*, 7(2):639–647, 2008.

[12] D. Malan, M. Welsh, and M. Smith. A public-key infrastructure for key distribution in tinyos based on elliptic curve cryptography. In *Proceedings of First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, IEEE SECON'04*, pages 71–80, 2004.

[13] B. Panja and S. K. Madria. An energy and communication efficient group key in sensor networks using elliptic curve polynomial. In E. Kranakis and J. Opatrny, editors, *ADHOC-NOW*, volume 4686 of *Lecture Notes in Computer Science*, pages 153–171. Springer, 2007.

[14] P. Szczechowiak, L. Oliveira, M. Scott, M. Collier, and R. Dahab. Nanoecc: Testing the limits of elliptic curve cryptography in sensor networks. In *Proceedings of European conference on Wireless Sensor Networks (EWSN08)*. Springer, 2008.

[15] H. Wang and Q. Li. Efficient implementation of public key cryptosystems on mote sensors. In *Proceedings of International Conference on Information and Communication Security ICICS'06*, volume 4307, pages 519–528. Springer, 2006.

[16] Y. Xiao, V. Rayi, B. Sun, F. Hu, and M. Galloway. A survey of key management schemes in wireless sensor networks. *Computer Communications*, 30(11-12):2314–2341, 2007.