ISSN 2278-3091
Volume 4, No.2, March - April 2015
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/ijatcse/static/pdf/file/ijatcse03422015.pdf

A survey in Wireless Sensor Networks based on Key Management Schemes

Sivakumar P
Assistant Professor
Department of Information Technology,
PSNA College of Engg. & Tech., Dindigul
sivapsnait@gmail.com

Saravanan S
Assistant Professor
Department of Information Technology,
PSNA College of Engg. & Tech., Dindigu
ssaravananme@gmail.com

Anandaraj M
Associate Professor
Department of Information Technology,
PSNA College of Engg. & Tech., Dindigul
anandaraj@psnacet.edu.in

***Abstract*— Wireless Sensor Networks (WSNs) consist of a set of small devices, called the sensor nodes, with sensing and wireless communication capabilities. They are used in many applications such as military, ecological, and health-related areas. WSNs may include certain constraints like low computation capability, limited energy resources, small memory and poor resilience to physical capture. Also, sensor nodes are mostly deployed in potentially adverse or even in hostile environment. These constraints and issues make security in WSNs a challenge. So, efficient key distribution and management schemes are must. To address the trade-off between above listed constraints and security, many key establishment techniques have been established. In this paper, the need for cryptographic schemes in key management and a comparison study between symmetric key and public key cryptographic schemes has been presented. Also, a survey on various key management schemes and a brief comparison among them is depicted. It is noticed that no key distribution technique is ideal to all scenarios where WSNs are used and the techniques employed should depend upon the requirements of the target applications and resources of each individual sensor network.**

Keywords- WSN, cryptography, key management, key predistribution, resilience

1. INTRODUCTION

Advances in wireless communication and electronics have resulted in the development of tiny sensor nodes which includes sensing, data processing, and communication components and hence it is possible to deploy Wireless Sensor Networks (WSNs) which represent a significant improvement over traditional wired sensor networks. WSNs are widely used in military applications, environmental applications, health applications, home applications, commercial applications and so on. These applications require communication in WSN which must be highly secure.

To inhibit the adversary from intercepting the transmitted information in the wireless communication channel and to prevent the false information distribution in the network, authentication and confidentiality must be used to achieve network security, which require key management.

The key establishment technique for a secure application must incorporate certain requirements like authenticity, confidentiality, integrity, scalability, flexibility, fault-tolerance, energy efficiency and self-healing.

The efficiency of the key management scheme is not only based on the ability to provide secrecy for the messages transferred, but also, it must include the following to deny the adversaries:

- *Resistance to replication*
  An adversary may try to compromise a few nodes in the network and then replicate those compromised nodes back into the network. A good key establishment technique must resist node replication to guard against such attacks.
- *Revocation of compromised nodes*
  If few nodes are compromised in the sensor network, the key establishment technique should provide an efficient way to revoke the compromised nodes.
- *Resilience*
  If a node within a sensor network is captured, the key establishment technique should ensure that the secret information about other nodes is not revealed.

The remaining part of the paper is organized as follows: Section II gives a view about cryptographic schemes, Section III gives the survey on various key management schemes, Section IV presents the analysis by comparing the discussed key schemes and finally Section V gives the conclusion.

2. CRYPTOGRAPHIC SCHEMES

Cryptographic schemes can be widely classified into symmetric and asymmetric cryptographic functions

A. Public Key Cryptography in WSNs

Asymmetric key cryptography is also known as public key cryptography. Due to resource limitations in the sensors, it is believed that public key cryptosystems are not suitable for WSNs. But even, recent studies have shown that public key cryptography can be applied to sensor networks by using the apt algorithms and low-power techniques. RSA and ECC algorithms are mostly investigated.

Comparison between RSA and ECC

The attraction of ECC is that it appears to offer equal security as that of RSA for a far smaller key size, thereby reducing processing and communication overhead. For example [2], RSA with 1024 bit keys (RSA-1024) provides a currently accepted level of security for many applications and

is equivalent in strength to ECC with 160 bit keys (ECC-160) [5].

The RSA private key operation, which is too slow, limits its use in a sensor node. ECC has no such issues since both the public key operation and private key operation use the same point multiplication operations.

There will be a central point with each sensor containing a certificate signed by the central point's private key using a RSA or ECC signature [2]. During the handshake process, the two parties will verify each other's certificate and establish the session key to be used in the communication.

In comparison with RSA cryptography at the same security level, ECDSA signatures are significantly cheaper than RSA signatures and ECDSA verifications are within reasonable range of RSA verifications. Also, the relative performance advantage of ECC over RSA increases as the key size increases, which is analyzed in terms of the execution time and energy cost.

B. Symmetric Key Cryptography in WSNs

Most research studies focus on symmetric key cryptography in wireless sensor networks because of the constraints in public key cryptography such as power consumption and complex computation. In [10], five popular encryption schemes, RC4 [6], RC5 [7], IDEA [6], SHA-1 [8], and MD5 [6,9], were evaluated on six different microprocessors ranging in word size from 8 bit (Atmel AVR) to 16 bit (Mitsubishi M16C) to 32 bit widths (StrongARM, XScale).

For each algorithm and platform, the code memory size and execution time were measured. The experiments indicated uniform cryptographic cost for each encryption and architecture class. Moreover, it is stated that hashing algorithms like MD5 and SHA-11 incurred higher overhead than encryption algorithms likeRC4, RC5, and IDEA.

In [11], Law *et al*. evaluated two different symmetric key algorithms namely, RC5 and TEA [12]. They further evaluated six block ciphers, including RC5 and RC6 [13], AES [10], MISTY1 [14]. Code, CPU cycles and data memory were the benchmark parameters for evaluation. The evaluation results showed that AES is suitable for high-security and energy-efficiency requirements while MISTY1 is suitable for good storage and energy efficiency.

The evaluation results in [15] concluded that AES is secure and it involves less number of rounds when compared to RC5 and RC6 which requires 18 and 20 rounds respectively.

Comparison between symmetric and public key schemes

In [16], Karlof *et al.* mentioned that the average execution time (in ms) of Skipjack and RC5 (symmetric key schemes) take only 0.38 and 0.26 respectively whereas the execution times of public key schemes are higher as mentioned in table I.

TABLE 1. PUBLIC KEY CRYPTOGRAPHY: AVERAGE OF ECC AND RSA EXECUTION TIMES [17]

Algorithm	Operation time (s)
ECC secp160r1	0.81s
ECC secp224r1	2.19s
RSA-1024 public-key $e = 2^{16} + 1$	0.43s
RSA-1024 private key w. CRT[1]	10.99s
RSA-2048 public-key $e = 2^{16} + 1$	1.94s
RSA-2048 private-key w. CRT[1]	83.26s
1 Chinese Remainder Theory	

Now, it is obvious that symmetric key cryptography is faster and consumes less energy when compared to public key cryptography.

3. KEY MANAGEMENT SCHEMES

Key management is an important mechanism to ensure the security in WSNs. The goal is to generate keys between sensor nodes which are required to exchange data. As shown above, public key cryptography has certain limitations in WSNs. This is the reason why most proposed key management schemes are based on symmetric key cryptography. Various such key management schemes are discussed in this session.

1) Centralized Key Management Schemes

In a centralized key management scheme, key distribution center (KDC) is the only entity to control key generation, distribution, revocation and rekeying. The LKHW scheme is based on the Logical Key Hierarchy (LKH) [18] which comes under centralized scheme. In this scheme, the base station will be treated as a KDC and all keys are logically distributed in a tree rooted at the base station [2].

This scheme has small memory requirement and perfectly controlled node replication. It is resilient to node capture and possible to revoke key pairs. However, as there is only one managing entity, it suffers from single point of failure and the base station becomes the target of attacks. The security of the entire network will be breached if there is a problem with the controller.

Furthermore, when the KDC is not working, keys cannot be generated, regenerated, and distributed which make the network to become vulnerable. Also, the scalability is being affected as it will be too hard for a single entity to manage a large network.

2) Distributed Key Management Schemes

Here, different controllers are involved in key management activities. This minimizes the risk of failure and provides better scalability. Distributed schemes are the most proposed key management schemes. These schemes can be classified into deterministic and probabilistic approaches.

A. Deterministic Approaches

LEAP and BROSK, the popular deterministic approaches are discussed below.

1) LEAP

In [19], Zhu *et al.* have proposed a key management protocol called Localized Encryption and Authentication Protocol (LEAP). In this scheme, four types of keys namely individual key, group key, pairwise key and cluster keys are established for each sensor node.

a) Establishing Individual Node Keys: Individual key is generated using a pseudo-random function f, node id and a master key K^m which is known only to the controller and is pre-loaded into each node prior to its deployment.

b) Establishing Pairwise Shared Keys: Each node *u* generates a master key $K_u = f\ K_i\ (u)$, where K_i is the initial key which is loaded during predistribution phase. Then, during the neighbor discovery phase, a HELLO message will be broadcasted by node u and it expects an acknowledgment from neighboring nodes. It can be represented as follows:

$$u \rightarrow^{*}: u$$
$$v \rightarrow u : v, MAC(K_v, u|v)$$

Now node *u* can compute its pairwise key, $f\ K_{uv} = f\ K_v\ (u)$, with v. Node v can also compute $f\ K_{uv}$ in the same way since it knows u, fK_v. Then, K_{uv} will be their pairwise key.

c) Establishing Cluster Keys: Suppose node u wants to establish a cluster key with all its immediate neighbours $v_1, v_2, \ldots v_m$. Node u first generates a random key K_u^y, then encrypts this key with the pairwise key shared with each neighbor, and finally transmits the encrypted key to each neighbor v_i where $1 \leq i \leq m$. LEAP uses unicast for key exchange.

2) BROSK

BROSK [20] is a fully ad hoc key negotiation protocol. Each node can negotiate a session key with its neighbors by broadcasting the key negotiation message, say, M1: $ID_A|N_A|\ MAC_K\ (ID_A|N_A)$ where K is the master key shared among all nodes and ID_A is the identity of node A. Once a node receives this, it can construct the shared session key by generating the MAC of two nonces. For example, if node B receives the broadcast message from node A. Node A also receives the broadcast message from node B, say, M2: $ID_B|N_B|\ MAC_K\ (ID_B|N_B)$. Then their shared session key will be K_{AB}: $MAC_A\ (N_A|N_B)$**.**

B. Probabilistic Approaches

In probabilistic approaches, before sensors are deployed, each sensor is preloaded with some keys. After deployment, sensor nodes will undergo a discovery phase during which shared keys can be established. But there is only some probability that two sensor nodes can set up a shared key to communicate securely. In this section, many such predistribution schemes are discussed.

1) Basic Scheme

In [21], Eschenauer and Gligor introduced a random key predistribution scheme for sensor networks, popularly known as "Basic Scheme". This scheme consists of three phases: key predistribution, shared key discovery, and path key establishment.

a) Key Predistribution Phase: In this phase, each sensor is preloaded with a key ring in its memory. From a large pool of *P* keys, *k* keys will be drawn randomly to form the key ring. A trusted node will be selected as the controller node where the key identifiers of each key in the key ring and the node identifiers are stored. Further, it is assumed that each sensor node shares a pairwise key with the controller node.

b) Shared Key Discovery Phase: In this phase, each sensor tries to discover its neighboring nodes with which it can share keys. To accomplish this two methods are suggested in [21]. The first method is that each node must broadcast a list of key identifiers that are associated with the keys in their key ring in plain text. This allows its neighboring nodes to check whether they share a key. This method seems to be simple. However there is a chance to identify the key-sharing patterns by the adversary. The second method is the challenge–response technique which helps to hide key-sharing patterns from the adversary. For every key K_i in the key ring, each node should broadcast a list of α which is the challenge text encrypted with K_i i.e. (α) where i=1,….,*k*. The recipient who can reveal the challenge text by the decryption of (α) with the proper key can establish a shared key with the broadcasting node.

c) Path-key Establishment Phase: This phase establishes a link between two nodes when they cannot set up a common key. This can be achieved by having an intermediate node which shares a common key with both the sender and the recipient nodes.

The messages from the controller node are signed by the pairwise key shared with the sensor nodes. This ensures that an adversary cannot forge a controller node. In spite of ideal resilience, this scheme is not scalable, and is not memory efficient, particularly in the case of large networks. In addition, after node deployment, if a new node wants to join the network, none of the previously deployed sensors will have a common key with the new node.

2) q-Composite keying scheme

In [22], Chan *et al.* suggested that by increasing the number of key overlap in the key ring, resilience against node capture can be increased. According to the *q* composite keying scheme, atleast q common keys must be shared to set up a key in order to establish a secure link between the communicating nodes. Also, a multipath key reinforcement was presented for a new phase called key update phase to enhance the basic scheme. Suppose if there are *j* disjoint paths between *A* and *B*, then *A* generates *j* random values $v_1, v_2, \ldots, v_j$. These random values are routed along each disjoint path to *B*.

When *B* receives all *j* keys, both A and B can compute a new link key as follows:

$$k' = k \oplus v1 \oplus v2 \oplus \ldots \oplus vj$$

If an adversary needs to reconstruct the key used for communication, then he/she must eavesdrop on all possible *j* paths. But this phase includes more communication overhead to find the disjoint paths.

3) Random-Pairwise key scheme

In [22], Chan *et al* proposed a random-pairwise keys scheme which states that storing all n-1 keys is no more necessary and only np keys are needed to be stored in the key ring where n denotes the number of nodes and p denotes the probability of establishing secure communication among two nodes. Also maximum allowable network size is given by with n = k/p where k represents the number of keys in a node's key ring. A node which needs to communicate with other sensor nodes starts broadcasting its identity; if any other node shares a pairwise key with the broadcasting node, then a cryptographic handshake is performed between the nodes which can ensure a secure link.

This broadcasting is also possible beyond the communication range of a node. In this case, the node identity is rebroadcasted by the intermediate nodes to a certain number of hops. But this range extension process should be done carefully as the adversary may perform a denial of service attack by dropping the malicious nodes into the network which create random node identities that may flood the network and thus slows down the entire process. By limiting the number of hops, this type of denial of service attack can be avoided.

Advantages of this scheme are: it provides good resilience to node capture prevents node replication. Disadvantages include poor scalability and this scheme will not support large sized networks.

4) Polynomial-based key predistribution protocol

In [23], Blundo *et al.* proposed a polynomial-based key predistribution scheme which is meant for group key predistribution. A bivariate *t*-degree polynomial $f(x, y) =$ is generated randomly by the key setup server over a finite field *Fq* where *q* is a prime number such that it obeys the property of $f(x, y) = f(y, x)$. The setup server computes a polynomial share $f(i, y)$ for each sensor i. For any two sensor nodes *i* and *j*, node *i* can compute the common key $f(i, j)$ by evaluating $f(i, y)$ at point *j*, and node *j* can compute the same key $f(j, i) = f(i, j)$ by evaluating $f(j, y)$ at point *i*. In this approach, each sensor node needs to store a *t*-degree polynomial $f(i, x)$, which occupies $(t + 1) \log q$ storage space. This scheme provides perfect security and *t*-collusion resistant. However, the disadvantage of this scheme includes the storage cost for a polynomial share which is exponential in terms of the group size.

5) Blom's scheme

In [24], Du *et al.* proposed another pairwise key predistribution scheme which uses Blom's method [25] to find a pairwise secret key between a pair of nodes. It is also stated that perfect security in the network is possible still no more than λ nodes are compromised, which is called the λ -secure property. In this method, the base station first constructs a $(\lambda+1) \times N$ matrix *G* during the predeployment phase, over a finite field $GF(q)$ where *N* is the network size and matrix *G* will be public. Then the base station computes a random $(\lambda + 1) \times (\lambda +1)$ symmetric matrix *D* over $GF(q)$, and creates a $N \times (\lambda +1)$ matrix $A = (D \times G)^T$ where $(D \times G)^T$ is the transpose of $D \times G$. Matrix *D* must be kept secret.

Here, $K_{ij} = K_{ji}$ by symmetry property. To carry out the above computation, in the predistribution phase, for any sensor $k = 1$to N:

- Store the k^{th} row of matrix *A* at node *k*
- Store the k^{th} column of matrix *G* at node *k*

Therefore, when node *i* and node *j* need to discover the pairwise key between them, they first exchange their columns of *G*, and then compute K_{ij} and K_{ji} using their private rows of *A* respectively.

6) Grid-based key distribution scheme

In [26], Polynomial Pool-Based Scheme using a grid-based key assignment is presented. In this grid-based approach an m × m grid is constructed with a set of 2m polynomials, where the value of m is the square root of N, where N is the number of sensors in the network.

Each row i in the grid is associated with a polynomial (x,y) and each column j of the grid is associated with a polynomial share (x,y). During the first phase of key establishment, the key setup server generates 2m t-degree bivariate polynomials over a finite field Fq. Then each node is assigned to an intersection in the grid. This intersection point is where the sensors will be deployed in the network. If the intersection is (i,j), then the node identity will be (i,j). Each node is equipped with its identity and the row and column polynomial shares of that grid intersection.

During the polynomial share discovery phase, to establish a pairwise key between node i and j, node i checks for common rows or columns with j, i.e., $c_i = c_j$ or $r_i = r_j$. If no match is found, node i finds an intermediate node through which it can setup a pairwise key with node j. It is to be noted that, grid provides many paths between two nodes and hence node i can find alternate paths to node j even if the adversary compromises few intermediate nodes. This scheme offers low computation and communication overhead. But this scheme suffers from storage overhead as each node must store the identities of compromised nodes in addition to the polynomial shares to avoid attack.

7) Key management scheme using deployment knowledge

In [27], Du et al. presented a key management scheme using deployment knowledge which is based on the Basic Scheme. This scheme is modeled using non-uniform probability density functions (pdfs) where the other schemes

discussed so far is based on uniform pdfs. Regarding non-uniform pdfs, the positions where the sensor nodes are to be deployed is assumed based on certain patterns. For example, to acquire an approximate knowledge about the positions of the sensors, by making an assumption that the sensors are dropped from an aircraft. Based on the deployment knowledge key distribution scheme will be developed. There are two terms used in deployment model: 1) Deployment point 2) Resident point. The deployment point and the resident point are two terms discussed in the deployment model. The point at which the sensor node is actually deployed is said to be deployment point and the point at which the sensor actually resides after deployment is said to be resident point. In Du et al. [27], the group deployment model is designed using Gaussian distribution function as follows:

• N sensor nodes that are deployed in a place are divided into t × n equal size groups. Each group $G_{i,j}$ for i = 1,. . . , t and j = 1,. . . ,n is from the deployment point with index (i,j) and (x_i,y_j) is the deployment point for $G_{i,j}$.

• The Deployment Model follows a grid-based approach with all deployment points arranged in a grid.

• The pdf of the resident points for node K in group $G_{i,j}$ is $(x,y|K \epsilon G_{i,j})=f(x-x_i , y-y_i)$

If two groups are deployed closer they may share common keys. As the deployment distance between the groups increases, the amount of key overlap decreases. Each group will have a sub key pool when combined will yield key pool S. Same as the basic scheme, predistrbution of keys in the deployment model includes three phases: key predistribution, shared key discovery, and path key establishment. *During key predistribution*, the key pool will be divided into subkey pools. Two subkey pools are said to be neighbors if their groups have nearby resident points. After dividing the key pool, nodes in each group are loaded with the keys from their corresponding subkey pool. The rest two schemes are same as that of the basic scheme. This scheme helps to increase resilience against node capture and reduces communication overhead. But it includes computational complexity.

4. ANALYSIS

Table 2 gives the comparative study of the various key management schemes considering the scalability, computational overhead, communication overhead, storage load and resilience to node capture as the parameters. Considering the keys, LKHW, LEAP and BROSK uses master keys. Cluster key is used only by LEAP and BROSK. Pairwise scheme is established in all the discussed schemes. Path key is established in all probabilistic key schemes.

5. CONCLUSION

Efficient key distribution and management schemes are necessary to face the security challenges in WSN. Various key management schemes have been discussed so far and it is analyzed that no key distribution technique is ideal to all scenarios and the technique employed depends upon the requirements of the target applications and resources of each individual sensor network.

TABLE 2. COMPARISON AMONG VARIOUS KEY MANAGEMENT SCHEMES

Approaches	Scalability	Computational overhead	Communication overhead	Storage load	Resilience to node capture
LKHW[18]	Limited	Low	Low	Low	Poor
LEAP[19]	Good	Low	Low	Low	Poor
BROSK[20]	Good	Low	Low	Low	Poor
Basic scheme[21]	Limited	Medium	Medium	High	Medium
q-Composite[22]	Good	High	High	High	Good
Random pairwise [22]	Limited	Low	Low	Low	Good
Polynomial based[23]	Good	Medium	Medium	High	Good
Blom's scheme[25]	Good	Medium	Medium	High	Good
Grid based[26]	Good	Low	Low	High	Good
Using deployment model[27]	Good	Medium	Low	Medium	Good

REFERENCES

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci,“ A survey on sensor networks”, IEEE Commun, vol.40, pp. 102-114, 2002.

[2] Yong Wang, Garhan Attebury, and Byrav Ramamurthy, “A survey of security issues in wireless sensor networks”, IEEE Commun,vol.8, 2nd quarter 2006.

[3] N. Gura, A. Patel, A. Wander, H. Eberle, S.C. Shantz, “Comparing elliptic curve cryptography and RSA on 8-bit CPUs”, in Proceedings of the 6th International Workshop on Cryptographic Hardware and Embedded Systems, Boston, August 2004.

[4] D. Hankerson, A. Menezes, S. Vanstone, Guide to Elliptic Curve Cryptography, New York: Springer-Verlag, 2004.

[5] Simon Blake-Wilson, Minghua Qu,” Elliptic Curve Cryptography”, SECG Std. SEC1, available at www.secg.org/collateral/sec1.pdf, 2000.

[6] A. J. Menezes, S. A. Vanstone, and P. C. V. Oorschot, Handbook of Applied Cryptography, Boca Raton, FL: CRC Press, 1996.

[7] R. L. Rivest, “The RC5 Encryption Algorithm,” Fast Software Encryption, Springer, pp. 86–96, 1995.

[8] D. Eastlake III and P. Jones, “US Secure Hash Algorithm 1 (SHA1),” RFC 3174 (Informational), Sept. 2001.

[9] P. Ganesan et al., “Analyzing and Modeling Encryption Over head for Sensor Network Nodes,” WSNA '03: Proc. 2nd ACM Int'l. Conf. Wireless Sensor Networks and Applications, pp. 151–59, 2003.

[10] J. Daemen and V. Rijmen, “AES Proposal: Rijndael,” Proc. 1st AES Conf., Aug. 1998.

[11] Y. W. Law et al., “Assessing Security- Critical Energy-Efficient Sensor Networks,” Proc. 18th IFIP TC11 Int'l. Conf. Info. Security, Security, and Privacy in the Age of Uncertainty (SEC), pp. 459–63, May 2003.

[12] D. J. Wheeler and R. M. Needham, “TEA, A Tiny Encryption Algorithm,” Proc. Fast Software Encryption: 2nd Int'l. Wksp., in Lecture Notes in Computer Science (series), B. Preneel (Ed.), vol. 1008, 1994.

[13] R. L. Rivest et al., “The RC6 Block Cipher,” submitted to NIST as a candidate for the AES.

[14] M. Matsui, “New Block Encryption Algorithm Misty,” Proc. 4th Int'l. Wksp. Fast Software Encryption, in LNCS, E. Biham (Ed.), vol. 1267, London: Springer-Verlag, pp. 54–68, 1997.

[15] Y. W. Law, J. M. Doumen, P. H. Hartel, “Benchmarking Block Ciphers for Wireless Sensor Networks (Extended Abstract),” 1st IEEE Int'l. Conf. Mobile Ad-hoc and Sensor Systems, IEEE Computer Society Press, Oct. 2004.

[16] C. Karlof, N. Sastry, and D. Wagner, “Tinysec: A Link-Layer Security Architecture for Wireless Sensor Networks,” SenSys '04: Proc. 2nd Int'l. Conf. Embedded Networked Sensor Systems, New York: ACM Press, pp. 162–75, 2004.

[17] N. Gura et al., “Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs,” CHES '04: Proc. Wksp. Cryptographic Hardware and Embedded Systems, Aug. 2004.

[18] R. D. Pietro et al., “LKHW: A Directed Diffusion-Based Secure Multicast Scheme for Wireless Sensor Networks,” ICPPW ’03: Proc. 32nd Int’l. Conf. Parallel Processing Wksps., IEEE Computer Society Press, pp. 397–406, 2003.

[19] S. Zhu, S. Setia, and S. Jajodia, “LEAP: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks,” CCS ’03: Proc. 10th ACM Conf. Comp. and Commun. Security, New York: ACM Press, pp. 62–72, 2003.

[20] B. Lai, S. Kim, and I. Verbauwhede, “Scalable Session Key Construction Protocols for Wireless Sensor Networks,” IEEE Wksp. Large Scale Real Time and Embedded Systems, 2002.

[21] L. Eschenauer and V. D. Gligor, “A Key-Management Scheme for Distributed Sensor Networks,” CCS ’02: Proc. 9th ACM Conf. Comp. and Commun. Security, New York: ACM Press, pp. 41–47, 2002.

[22] H. Chan, A. Perrig, D. Song, “Random Key Predistribution Schemes for Sensor Networks,” Proc. IEEE Symp. Security and Privacy, May 2003.

[23] C. Blundo et al., “Perfectly-Secure Key Distribution for Dynamic Conferences,” CRYPTO ’92: Proc. 12th Annual Int’l.Cryptology Conf. Advances in Cryptology, London: Springer-Verlag, pp. 471–86, 1993.

[24] W. Du et al., “A Pairwise Key Predistribution Scheme for Wireless Sensor Networks,” CCS ’03: Proc. 10th ACM Conf. Comp. and Communications Security, New York: ACM Press, pp. 42–51, 2003.

[25] R. Blom, “An Optimal Class of Symmetric Key Generation Systems,” Proc. EUROCRYPT ‘84 Wksp., Advances in Cryptology, New York: Springer-Verlag, pp. 335–38, 1985.

[26] D. Liu and P. Ning, “Establishing Pairwise Keys Distributed Sensor Networks,” CCS ’03: Proc. 10th ACM Conf. Comp. and Commun. Security, New York: ACM Press, pp. 52–61, 2003.

[27] W. Du et al., “A Key Management Scheme for Wireless Sensor Networks using Deployment Knowledge,” Proc. IEEE INFOCOM, Hong Kong, pp. 586–97, 2004.