Volume 3, Issue 9, September 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Notable Trends in Key Distribution Schemes for Wireless Sensor Networks

Ramu Kuchipudi
MJCET, IT Department, Hyderabd-34
India

Dr.Ahmed Abdul Moiz Qyser
Head,MJCET,CSE Department,Hyderabad-34
India

Abstract*— Key management has remained a challenging issue in wireless sensor networks (WSNs) due to the constraints of sensor node resources. Various key management schemes that trade off security and operational requirements have been proposed in recent years. Wireless sensor networks (WSN) with low cost, low energy consumption and high utilization are becoming practically feasible through recent advances in wireless communication and microelectronics. The security concerns of the sensor nodes becomes a challenging issue since the nodes are often placed in hostile or adverse environment. Key Management is a critical security service for communication in WSNs. The key management system should be substantially secure, robust and efficient for a secure communication protocol. Many key establishment techniques have come up to address the tradeoffs between limited memory and security but choosing an effective scheme is debatable. In this paper, we provide a survey of various key management schemes in WSNs. choosing a key management scheme depends upon the target applications requirements and the resource of the sensor network.

***Keywords*— key management, sensor networks, probabilistic key sharing**

I. INTRODUCTION

A wireless sensor network (WSN) is a network formed by a large number of sensor nodes, each equipped with sensor(s) to detect physical phenomena such as heat, light, motion, or sound. Using different sensors, WSNs can be implemented to support many applications including security, entertainment, automation, industrial monitoring, public utilities, and asset management. However, many WSN devices have severe resource constraints in terms of energy, computation, and memory, caused by a need to limit the cost of the large number of devices required for many applications and by deployment scenarios that prevent easy access to the devices. Such resource limitations lead to many open issues — including WSN security — which have been studied actively by researchers. Many applications require WSNs to exchange sensitive information or contain feedback processes that have high reliability requirements, and they require a high level of security to succeed. Yet, strong security is difficult to achieve with resource-limited sensor nodes, and many well-known approaches become infeasible. In this article, we explore the security issues for key management for WSNs. First, we examine the needs and requirements for key management. Then, we explore several promising key management protocols, and conclude with a discussion of future trends that may affect their development.

Wireless sensor networks (WSN) consist of a large collection of sensor nodes with each node equipped with sensors, processors and radio transceiver. Large number of sensor nodes can be deployed in a variety of situations capable of performing both military and civilian tasks owing to their low cost. Key Management is a security aspect that gets a great deal of attention in Wireless sensor networks. Key Management establishes the keys that are required for providing confidentiality, integrity and authentication requirements. Key Management establishes secure connection between nodes at network formation stage, ensures that messages are encrypted and communicating nodes are authenticated. Asymmetric cryptography is not suitable for most sensor networks because of increased energy consumption and large code computation and storage requirements. Hence several alternative approaches have come up for performing key management in wireless sensor networks.

II. NETWORK MODELS

Communication in WSNs usually occurs in ad hoc manner, and shows similarities to wireless ad hoc networks. Likewise, WSNs are dynamic in the sense that radio range and network connectivity changes by time. Sensor nodes dies and new sensor nodes may be added to the network. However, WSNs are more constrained, denser, and may suffer (or take advantage) of redundant information. WSN architectures are organized in hierarchical and distributed structures as shown in Figure 1.A Hierarchical WSNs (HWSN) is shown in Figure 1(a); there is a hierarchy among the nodes based on their capabilities: base stations, cluster heads and sensor nodes.

Base stations are many orders of magnitude more powerful than sensor nodes and cluster heads. A base station is typically a gateway to another network, a powerful data processing / storage center, or an access point for human interface. Base stations collect sensor readings, perform costly operations on behalf of sensor nodes and manage the network. In some applications, base stations are assumed to be trusted and temper resistant. Thus, they are used as key distribution centers. Sensor nodes are deployed around one or more hop neighborhood of the base stations. They form a dense network where a cluster of sensors lying in a specific area may provide similar or close readings. Nodes with better resources, named as cluster heads, may be used collect and merge local traffic and send it to base stations. Transmission

power of a base station is usually enough to reach all sensor nodes, but sensor nodes depend on the ad hoc communication to reach base stations.

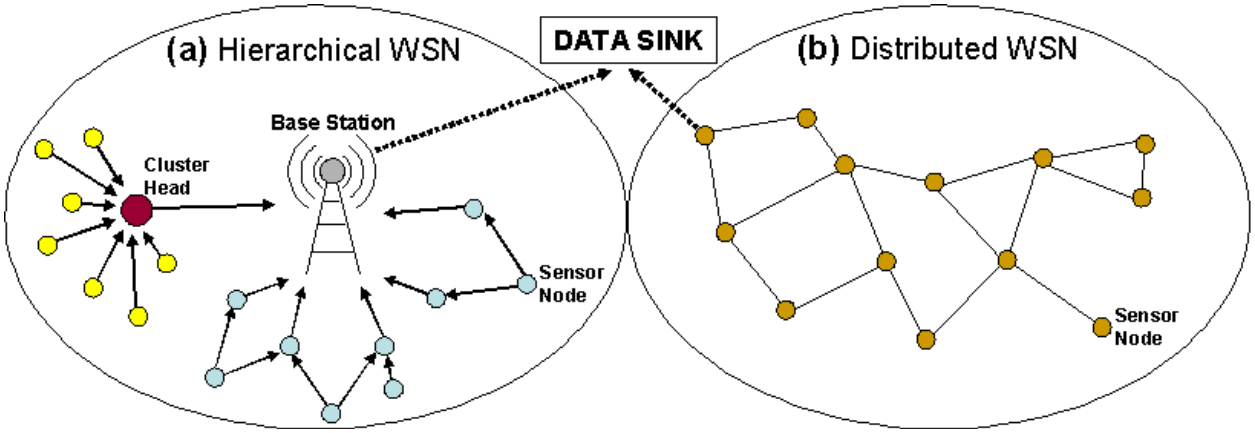

Fig. 1. Network Models: Hierarchical and Distributed Wireless Sensor Networks.

III. SECURITY AND OPERATIONAL REQUIREMENTS FOR KEY MANAGEMENT

Key management requirements can be divided into security requirements that form a subset of the overall WSN security requirements and operational requirements that act as constraints in the design and realization of key management. For key management, the most critical requirements are robustness and self-organization. Although confidentiality and integrity are important, the ability to distribute secret, shared keys satisfies both requirements, and all key management schemes are able to accomplish this. Likewise, data freshness is typically attained by including a nonce (a cryptographic time stamp) in each packet to verify that the data is new. That approach hinges on the integrity of each data packet to ensure the nonce has not been modified, which easily can be accomplished after a secret, shared key is established. On the other hand, self-organization —the ability to independently self-organize and self-heal in the presence of dynamic changes in a WSN — is a requirement that is more difficult to satisfy. Normally, without the considerations of security, WSNs are designed to satisfy this property such that nodes can freely form connections around a failed node or re-establish the network automatically after it has been disturbed.

However, when a key management scheme distributes certain communication keys to a few nodes, this requirement can be violated as other nodes are unable to form connections dynamically with these specific nodes for lack of proper keys. With robustness, the problem lies with the compromise of one or more nodes. Because WSN nodes are frequently deployed in unsupervised and remote locations, physical tampering is a real threat, and the WSN must be able to withstand the compromise of some nodes. If the network uses only one key, then the compromise of one node compromises the entire network. If the network uses multiple keys, it is interesting to determine how many compromised nodes it takes to compromise the security of the entire network. We look at the distribution schemes and discuss robustness in more detail later. On the other hand, several operational requirements exist for WSNs: accessibility, flexibility, and scalability. These requirements act as constraints for security design because one must ensure that they are not violated in the design and realization of a security scheme. Accessibility, the need for data to be accessible by many nodes, arises as WSNs must efficiently utilize the limited energy, computation, and memory.

IV. EXISTING KEY MANAGEMENT SCHEMES

To realize a practical, robust keying model, researchers have proposed various key management protocols that address the problems in each of the three basic schemes discussed previously. In this section, five different key management protocols are presented and reviewed in chronological order.

A. Straight forward approaches:

The three simplest keying models that are used to compare the different relationships between the WSN security and operational requirements are network keying, pairwise keying, and group keying. A Single mission key is used. Pairwise private key sharing between every two nodes is impractical. It requires pre-distribution and storage of n-1 keys in each node which is n(n-1)/2 per WSN. Most of the keys would be unusable since direct communication is possible only in the nodes neighborhood addition and deletion of the node and re-keying are complex.

B. Basic Probabilistic Approach:

The scheme includes selective distribution and revocation of keys to sensor nodes as well as node re-keying without substantial computation and communication capabilities. It re-lies on probabilistic key sharing among the nodes of a ran-dom graph and uses simple protocols for shared-key discovery and path-key establishment, and for key revocation,re-keying, and incremental addition of nodes. Eschenauer and Gligor [3] presented one of the first key management schemes for WSNs that is elegant, simple, and provides an effective tradeoff between robustness and scalability.

This scheme works as follows:

• Generate a large pool of keys (e.g., 10,000 keys).

• Randomly take k keys out of the pool to establish a key ring, where k << N, where N is the total number of nodes. Each node receives its own unique key ring, consisting of a subset of keys.
• When two nodes must communicate, they search for a common key within the key ring by broadcasting the identities (IDs) of the keys they have. If such a key does not exist,they attempt to communicate through a common third party, who is able to establish communications with both nodes. This phase is called path-key discovery.

As one can see, as long as the total number of keys stored in a node is less than N – 1, the scheme uses less storage than a pure pairwise scheme. This scheme also is scalable because the number of keys in the pool and the size of the key ring are both adjustable. Therefore, a more mission critical application can use a larger pool of keys and adjust the key ring size appropriately to be more secure. However, this scheme has some drawbacks. Compared to the newer schemes, Eschenauer's random key distribution is just a key distribution scheme. It lacks the authentication process and does not clearly define any process for revoking or refreshing keys. In addition, the dynamic handshaking process for each connection prevents any form of passive data aggregation; thus, one event detected by two neighboring nodes will result in two separate signals. There is no support for clustering or collaborative operations. If a home lighting automation application uses Eschenauer's keying scheme, turning off all the lights on one floor would entail sending a message to each light, which is rather inefficient. Lastly, since not every node is guaranteed to have a common key with all of its neighbors, there is a chance that some nodes will be unreachable [3]. Overall, Eschenauer's scheme failed to satisfy security requirement authentication and operational requirement accessibility.

C. Q-Composite Random Key Pre-distribution scheme:

This scheme [11] does not need to establish pair-wise key between every pair of nodes in a sensor network for a secure key management scheme for the wireless sensor networks. Communicating nodes should share at least Q number of keys. Thus in case of a key compromise, the nodes can communicate with the other keys. The value Q should be so selected such that the network maintains a certain desired level of connectivity. The size of the random key pool is reduced but this gives an advantage to the adversary. Only a few nodes need to be compromised to compromise the entire network.

D. Pairwise Key Predistribution Scheme:

In 2003, Du et al. proposed a key management scheme [4] based on the pairwise keying model. This model extends Eschenauer and Blom's work [5] by using the same paradigm as Eschenauer and Gligor [3] but instead of individual keys, it uses the concept of Blom's key matrix, which is an array of keys. In Du's scheme, there are k key matrices in each node, and the key matrices are distributed randomly. Blom's model is based on the idea of a symmetric matrix multiplication, where row i column j is equivalent to row j column i. Thus, when node i calculates key ij and node j calculates key ji, the keys are identical, leading to a commonly shared secret. Blom's scheme distributes the information required for this calculation in terms of a public matrix and a private matrix. In Du's pairwise key management scheme, instead of using only one private matrix, the sink node generates i private matrices, and each node stores a subset of these matrices in the same manner as Eschenauer's key ring. When two nodes must communicate, they start by broadcasting the node IDs, the indices of key matrices they carry, and the seed of the column of the public matrix. If they share a common key matrix, then they can compute the pairwise secret key using Blom's scheme. If they do not share a common key matrix, they will go into a path-key discovery phase to find a common third party to route the data.

The benefit of Du's scheme is that it offers an even stronger robustness against node compromise at a reasonable scalability cost. The authors claimed that an adversary must compromise five times as many nodes compared with Eschenauer's scheme to compromise the entire network. Their analysis of scalability shows that the energy cost remains reasonable and on par with the energy cost of using the advanced encryption standard for a WSN consisting of 264 nodes, which is 48 times higher than the maximum number of nodes defined in IEEE 802.15.4. The main disadvantage of this scheme is its complexity, which makes it hard to implement and increases overhead costs. Also, cluster operations are not supported because it is a pairwise keying scheme, and neither key revocation nor key refreshing are considered. The operational requirement, accessibility, is also difficult to satisfy because nodes will not be able to passively monitor communications. Lastly, compared to other, simpler schemes, Du's scheme likely uses more energy due to its computational complexity and on-demand key computation. To summarize,Du's scheme can satisfy most requirements but in the operational requirements area, it fails to satisfy accessibility and may not be competitive with simpler schemes in terms of scalability due to its higher overhead costs.

E. Localized Encryption and Authentication Protocol:

Zhu, Setia, and Jajordia introduced the localized encryption and authentication protocol (LEAP)[6], which employs a hybrid approach. This is a jack-of-all-trades protocol offering network-wide,cluster/group, and pairwise keying capabilities. To accomplish this, LEAP uses four types of keys: individual, group, cluster, and pairwise shared keys. The individual key is unique for each sensor node to communicate with the sink node.The group key is a network-wide key for communication from the sink node to all sensor nodes.An authentication mechanism known as μTimed Efficient Streaming Loss-tolerant Authentication Protocol (μTESLA) [7] is used for the broadcast authentication of the sink node, which ensures that packets sent with the group key are from the sink node only. The cluster key is used for collaborations within a cluster. An authentication mechanism known as a one-way hash-key chain that employs a non-reversible mathematical operation is used to ensure that the source of the packet can be authenticated without precluding passive data aggregation. Lastly, the pairwise shared key is used for secure communications between neighboring nodes.LEAP

uses a pre-distribution key to help establish the four types of keys. The individual key is first established using a function of a seed and the ID of the node. Then, in the pairwise shared key phase, a neighbor discovery process is initiated, and nodes broadcast their IDs.

The receiving node uses a function, seeded with an initial key, to calculate the shared key between it and all of its neighbors. Afterwards, the initial key and any intermediate keys that were generated are erased. Thirdly, the cluster key is distributed by the cluster head using pairwise communication secured with the pairwise shared key. Lastly, for distributing the network-wide group key, the sink node broadcasts it in a multihop,cluster-by-cluster manner starting with the closest cluster. LEAP has many advantages that satisfy the requirements of WSNs. First, it has μTESLA and one-way key chain authentication, as well as key revocation and key refreshing. The accessibility requirement also can be easily satisfied by encrypting data that requires aggregation with the cluster key. The fine granularity it supports enables data to be encrypted at the correct level (i.e., key level) to ensure reasonable security is achieved without prohibiting data fusion or aggregation. The scalability of LEAP can be analyzed in terms of computational cost and storage cost. The computational cost of LEAP is inversely proportional to the number of nodes and directly proportional to the number of neighbors (i.e., node density) [6] because the higher the density of the network, the more connections are formed per cluster. The storage cost also is quite reasonable as pairwise keying is used only for one-hop neighbors. It is apparent that LEAP satisfies both the security and operational requirements very well. The only drawback with LEAP is that it assumes the sink node is never compromised.

F .Location-Aware Combinatorial Key Management Scheme :

The Scalable, Hierarchical, Efficient, Locationaware,and Light-weight (SHELL) protocol [8] is a complicated cluster-based key management scheme published recently. It is influenced by LEAP with its use of multiple types of keys but introduces a new distributed key management entity. Each cluster has its own distributed key management entity residing in a non-clusterhead node. Thus, the operational responsibility and key management responsibility are separated,leading to a better resiliency against node capture.

The main benefit of SHELL is that it has a high robustness against node capture. Although some nodes have unique functions, the capture of that particular node does not reveal enough keys to compromise the entire network nor to disrupt the operation of the network. For instance, the capture of a key-generating gateway node, a key management entity, does not compromise the network because it does not contain the key between the cluster head and the cluster nodes. Likewise, due to the distributed feature, there are at least two key-generating gateway nodes and the disruption of one does not hinder the operation of the network.

In addition, SHELL accounts for the processes for node addition, replacement, and refreshment of The main benefit of SHELL is that it has a high robustness against node capture. Although some nodes have unique functions,the capture of that particular node does not reveal enough keys to compromise the entire network nor to disrupt the operation of the network.keys. It also supports cluster (group) communications and does not preclude data fusion or aggregation within the clusters. However,SHELL has some drawbacks. Its structure and operation are highly complex, involving heterogeneous node operations and multiple (at least seven) types of keys. The specific network entities include the gateway node, key-generating gateway node, inter-gateway node, command node, and sensor node. The energy consumption and cryptographic overheads, although scalable,have not been compared with other schemes.Due to its increased complexity, the energy usage and cryptographic overhead are likely higher than other schemes. Finally, the implementation of such a complex protocol also may be difficult with the current programming limitations of a WSN. To conclude, SHELL focuses on satisfying the robustness and security requirements while trading off the availability requirement since higher complexity leads to higher energy usage and lowers the mean time between failures. In this case, failures would likely occur due to the depletion of battery energy of individual nodes.

G. Energy and Communication Efficient Group Key Management Protocol:

Panja et al. [9] recently introduced a hierarchical group keying scheme using the Tree-based Group Diffie-Hellman (TGDH) protocol. The main feature of this scheme is that each key is made up of many partial keys. By breaking up the keys into smaller components, it makes rekeying an efficient and simple task by revoking,changing, or adding one or more partial key(s).The TGDH keying scheme works on a hierarchical WSN that has one level of general sensor nodes and multiple levels of cluster heads; that is, there can be a head of clusters responsible for multiple cluster heads below it in a tree-like manner. The data collection process starts with a group of sensor nodes collecting data from a region of interest and sending it to the nearest cluster head. The cluster head then aggregates the data to reduce the size and overhead and sends it to its parent. The parent, if it has multiple children, repeats the process of data aggregation and forwards the data to its parent until the sink node is reached.To establish the keys in this hierarchical treebased WSN, two separate schemes are used:intra-cluster and inter-cluster keying. The intracluster keying process starts with the leaf nodes sending each of their partial keys to the parent.Then the parent calculates its own partial key and combines the partial keys together to form the cluster key. The parent node then broadcasts this cluster key to its leaf nodes.

All communications are encrypted with a pre-distributed key to provide confidentiality during this early stage.Afterwards, the inter-cluster keying is initiated. This process is very similar to that of intra-cluster keying except that intermediate keys of the parents (children of the next higher level) are used instead of the partial keys.The advantage of this scheme is that compared to SHELL, it is simple and elegant and hence, easy to implement. Panja et al. also simulated the performance of their scheme in comparison with Security Protocols for Sensor Networks (SPINS) [7], a keying protocol that establishes secure one-to-one communication in a WSN. The results are promising with fast and

scalable key delivery time and energy usage.Also, by using small partial keys, the storage and computational costs are reduced. This is especially significant for the leaf nodes that frequently have the least amount of resources at their disposal. The drawback of Panja's scheme is that although key revocation and key refreshing processes are addressed, node addition and replacement are not considered explicitly. In addition,its security robustness against the compromise of the initial pre-distributed key has not been analyzed.Further, the security strength of the low complexity keys at the leaf nodes, for example,against brute-force attacks, has not been proven.To conclude, Panja's scheme trades off robustness to better satisfy the self-organization, accessibility, flexibility, and scalability requirements. The use of a tree-based hierarchical structure also ensures that this *protocol* is very scalable.

H. .Polynomial based key pre-distribution scheme:

Blundo et al. [12] distributes a polynomial share (a partially evaluated polynomial) to each sensor node using which every pair of nodes can generate a link key. Symmetric polynomial P(x, y) (P(x, y) = P(y, x)) of degree, d is used. The coefficients of the polynomial come from GF(q) for sufficiently large prime q. Each sensor node stores a polynomial with d+ 1 coefficients which come from GF(q). Sensor node Si receives its polynomial share of fi(y) = P(i, y). Si (resp. Sj) can obtain link key Ki,j = P(i, j) by evaluating its polynomial share fi(y) (resp. fj(y)) at point j (resp. i). Every pair of sensor nodes can establish a key.

V. TRENDS IN KEY MANAGEMENT

Although there are many papers published on various keying schemes, whether or not they will be implemented or used in practice depends on market demands. A proposed scheme likely will be adopted in practice if it targets open standards. Recently, both the IEEE 802.15 task group and the ZigBee Alliance have released newly revised standards.

Despite the fact that the new IEEE 802.15.4b still does not specify a key management scheme, it clarifies ambiguities and enhances many features compared with the original standard. In this section, we discuss the security related changes in 802.15.4b and the features relevant to key management in the newly enhanced Zig-Bee standard 2006.The 802.15 task group 4b was chartered to provide specific enhancements and clarifications to the original 802.15.4-2003 standard. The revised standard was published in September 2006 as 802.15.4-2006. In terms of the security changes, the standard introduces a new counter with cipher block chaining mode* (CCM*) cipher suite mode, which incorporates the confidentiality- only and authentication-only modes that were provided, non-securely, by cipher block chaining (CBC) and counTeR (CTR) cipher suite modes in the original standard.Another new feature of the 4b standard that is relevant to key management is secure group keying. Many applications can benefit from secure broadcasting or multicasting abilities. Unfortunately, the secure broadcasting mechanism in 802.15.4-2003 is actually insecure because it does not provide data freshness and is vulnerable to replay attacks.

The replay attack vulnerability can be exploited against applications that require secure broadcast, such as home automation. Imagine the devastating effect of a replayed lights-off broadcast right before a home invasion. Hence, replay resistant broadcasting has been introduced in 4b using frame counters on a per-device basis, and the feature is made more flexible with the introduction of group keying. Many other small clarifications were proposed, but only the most relevant ones were presented here.The ZigBee Enhanced standard was released in September 2006 and offers improvements and new features. The most relevant feature is the support of group devices and targeted broadcasts.Like 802.15.4b, the ZigBee Alliance also has treated group device support with a high priority. In addition, the new targeted broadcast feature that can reach a specific subset of devices (routers, end nodes, sleeping nodes, currently awake nodes) is important to key management scheme selection because it adds a new functional requirement. It is worthwhile to take these new features into account when considering the various key management schemes.Based on the new trends in 802.15.4b and the ZigBee Enhanced standard, one can easily assume that a purely random or pairwise keying scheme like Eschenauer [3] and Du [4] would not be commercially viable. The keying schemes that offer group or multicast abilities are much more compatible with industry trends. Thus, the development of a key management scheme that incorporates the flexible network, pair, and cluster abilities found in LEAP, along with the adjustable robustness of Eschenauer or Du may be desirable. The main issue with SHELL is its high complexity; further developments can target streamlining of its processes and reduction of overheads. Panja's scheme also has some advantages for extremely large networks, and one might want to enhance Panja's scheme with the multiple-level keying of LEAP, for added flexibility,or with Eschenauer's or Du's scheme, for increased robustness. In any case, the trade offs presented earlier remain true, and the two new industry standards provide a good guideline to the acceptable essential services that a viable key management solution must provide.

ACKNOWLEDGEMENTS

The work was partly supported by the R & D Cell of Muffakham Jah College of Engineering & Technology, Hyderabad, India. The authors would like to thank to all the people from Industry and Academia for their active support.

VI. CONCLUSION

Many researchers have worked on Key management for Wireless Sensor Networks (WSNs) which is a very critical issue from the security point of view. In this paper, we have presented an overview of some the schemes presented in various papers. The choice of deciding on a particular key management scheme should be based the requirements of that particular application. There are immense research opportunities in the field of key management in wireless sensor network. Further study on the security aspects of key management in WSNs will make the wireless sensor networks immensely useful in various aspects of life.

REFERENCES

[1] D. W. Carman, P. S. Kruus, and B. J. Matt, "Constraints and Approaches for Distributed Sensor Security," NAI Labs tech. rep. 00-010, 2000.

[2] N. Sastry and D. Wagner, "Security Considerations for IEEE 802.15.4 Networks," Proc. *2004 ACM Wksp. Wireless Sec*., 2004, pp. 32–42.

[3] L. Eschenauer and V. D. Gligor, "A Key-Management Scheme for Distributed Sensor Networks," *Proc. 9th ACM Conf. Comp. and Commun. Sec.*, 2002, pp. 41–47.

[4] W. Du et al., "A Pairwise Key Predistribution Scheme for Wireless Sensor Networks," *Proc. 10th ACM Conf.Comp. Commun. Sec*., 2003, pp. 42–51.

[5] R. Blom, "An Optimal Class of Symmetric Key Generation Systems*," Proc. EUROCRYPT '84 Wksp. Advances in Cryptology: Theory and App. of Cryptographic Techniques*,1985, pp. 335–38.

[6] S. Zhu, S. Setia, and S. Jajodia, "LEAP: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks,"*Proc. 10th ACM Conf. Comp. and Commun.Sec.*, 2003, pp. 62–72.

[7] A. Perrig et al., "SPINS: Security Protocols for Sensor Networks," *Wireless Network*, vol. 8, 2002, pp. 521–34.

[8] M. F. Younis, K. Ghumman, and M. Eltoweissy, "Location-Aware Combinatorial Key Management Scheme for Clustered Sensor Networks," *IEEE Trans. Parallel and Distrib. Sys.*, vol. 17, 2006, pp. 865–82.

[9] B. Panja, S. K. Madria, and B. Bhargava, "Energy and Communication Efficient Group Key Management Protocol for Hierarchical Sensor Networks," SUTC '06:Proc. IEEE Int'l. Conf. Sensor Networks, Ubiquitous, and Trustworthy Comp., 2006, pp. 384–93.

[10] S. A. Camtepe and B. Yener, "Key Distribution Mechanisms for Wireless Sensor Networks: A Survey," Tech.rep. TR-05-07, Dept. of Comp. Sci., Rensselaer Polytechnic Inst., 2005.

[11] Chan, H., Perrig, A., and Song, D. 2003. Random Key Predistribution Schemes for Sensor Networks. In Proceedings of the 2003 IEEE Symposium on Security and Privacy (May 11 - 14, 2003). SP. IEEE Computer Society, Washington, DC, 197-213.

[12] Blundo, C., Santis, A., Herzberg, A., Kutten, S., Vaccaro, U., and Yung, M. 1992. Perfectly-secure key distribution for dynamic conferences. In Crypto 92

