Key Distribution Mechanisms for Wireless Sensor Networks: a Survey

SEYIT A. ÇAMTEPE and BÜLENT YENER
Rensselaer Polytechnic Institute

Advances in technology introduce new application areas for sensor networks. Foreseeable wide deployment of mission critical sensor networks creates concerns on security issues. Security of large scale densely deployed and infrastructure-less wireless networks of resource limited sensor nodes requires efficient key distribution and management mechanisms. We consider distributed and hierarchical wireless sensor networks where unicast, multicast and broadcast type of communications can take place. We evaluate deterministic, probabilistic and hybrid type of key pre-distribution and dynamic key generation algorithms for distributing pair-wise, group-wise and network-wise keys.

General Terms: Security,Theory

Additional Key Words and Phrases: Combinatorial key pre-distribution, distributed wireless sensor network, dynamic key generation, group-wise key, hierarchical wireless sensor network, key distribution, key matrix, key pre-distribution, master key, network-wise key, pair-wise key, pair-wise key pre-distribution, polynomial key share, random key pre-distribution

1. INTRODUCTION

Sensors are inexpensive, low-power devices which have limited resources [Akyildiz et al. 2002]. They are small in size, and have wireless communication capability within short distances. A sensor node typically contains a power unit, a sensing unit, a processing unit, a storage unit, and a wireless transmitter / receiver. A wireless sensor network (WSN) is composed of large number of sensor nodes with limited power, computation, storage and communication capabilities. Environments, where sensor nodes are deployed, can be controlled (such as home, office, warehouse, forest, etc.) or uncontrolled (such as hostile or disaster areas, toxic regions, etc.). If the environment is known and under control, deployment may be achieved manually to establish an infrastructure. However, manual deployments become infeasible or even impossible as the number of the nodes increases. If the environment is uncontrolled or the WSN is very large, deployment has to be performed by randomly scattering the sensor nodes to target area. It may be possible to provide denser sensor deployment at certain spots, but exact positions of the sensor nodes can not be controlled. Thus, network topology can not be known precisely prior to deployment. Although topology information can be obtained by using mobile sensor nodes and self-deployment protocols as proposed in [Wang et al. 2004] and [Zou and Chakrabarty 2003], this may not be possible for a large scale WSN.

Security in WSN has six challenges: (i) wireless nature of communication, (ii)

Authors' address: Rensselaer Polytechnic Institute, Computer Science Department, Lally 310, 110 8th Street, Troy, NY 12180-3590.
Technical Report TR-05-07 (March 23, 2005).

resource limitation on sensor nodes, (iii) very large and dense WSN, (iv) lack of fixed infrastructure, (v) unknown network topology prior to deployment, (vi) high risk of physical attacks to unattended sensors. Moreover, in some deployment scenarios sensor nodes need to operate under adversarial condition. Security solutions for such applications depend on existence of strong and efficient key distribution mechanisms. It is infeasible, or even impossible in uncontrolled environments, to visit large number of sensor nodes, and change their configuration. Moreover, use of a single shared key in whole WSN is not a good idea because an adversary can easily obtain the key. Thus, sensor nodes have to adapt their environments, and establish a secure network by: (i) using pre-distributed keys or keying materials, (ii) exchanging information with their immediate neighbors, or (iii) exchanging information with computationally robust nodes. Although there are ongoing works [Malan et al. 2004; Gaubatz et al. 2004; Huang et al. 2003] to customize public key cryptography and elliptic key cryptography for low-power devices, such approaches are still considered as costly due to high processing requirements. Key distribution and management problem in WSN is difficult one, and requires new approaches.

Motivation of this paper is to evaluate the key distribution solutions. Depending on application types, it is possible to discuss: (i) network architectures such as distributed or hierarchical, (ii) communication styles such as pair-wise (unicast), group-wise (multicast) or network-wise (broadcast), (iii) security requirements such as authentication, confidentiality or integrity, and (iv) keying requirements such as pre-distributed or dynamically generated pair-wise, group-wise or network-wise keys. In this paper, we provide a comparative survey, and taxonomy of solutions. It may not be always possible to give strict quantitative comparisons; however, there are certain metrics, as described in the next section, that can be used to evaluate the solutions. The structure of the paper is as follows: in Section 2 common terms and definitions are given, in Section 3 network models are defined, in Section 4 security vulnerabilities and requirements are discussed, in Sections 5 and 6 key distribution solutions are evaluated, and finally in Section 7 we provide summary and discussions.

2. TERMS, DEFINITIONS AND NOTATIONS

Terms used throughout this paper are as follows:

—*key*: symmetric key which is used to secure communication among two or more sensor nodes,

—*keying materials*: any kind of information and algorithms which are used to generate keys,

—*credentials*: keys, keying materials and algorithms,

—*key-chain*: list of keys or keying materials which are stored on a sensor node,

—*key-pool*: list of all keys or keying materials which are used in the WSN,

—*link-key*: key which is used to secure communication over a direct wireless link,

—*path-key*: key which is used to secure communication over multi-hop wireless links, through one or more sensor nodes,

—*pair-wise key*: key which is used to secure unicast communication between a pair of sensor nodes over single or multi-hop wireless link,

Abbreviations		**Notations**	
KDC	Key Distribution Center	N	WSN size
WSN	Wireless Sensor Network	KP	Key-Pool
HWSN	Hierarchical WSN	KC	Key-Chain
DWSN	Distributed WSN	K	Key
Hash	Hash	BS	Base Station
MAC	Message Authentication Code	S	Sensor node
PRF	Pseudo Random Function	RN	Random Nonce
ENC	Encryption	P	Polynomial
DAG	Directed Acyclic Graph		

Table I. Abbreviations and notations. Functions MAC and ENC accept a key and message to generate message authentication code and encrypted message respectively. Function PRF accepts a seed to generate a random number. Also, it is used to generate a key in which case part of the seed must be secret information.

- *group-wise key*: key which is used to secure multicast communication among a group of sensor nodes over single or multi-hop wireless link,
- *network-wise key*: key which is used to secure broadcast messages,
- *key reinforcement*: establishing a unique session key between two sensor nodes by using existing link- or path-key,
- *key graph*: a graph where nodes are sensor nodes, and there is an edge in between two nodes if the corresponding sensor nodes are within each others radio range, and if they share a key to secure their communication.

3. 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. Thus, data flow in such networks can be: (i) pair-wise (unicast) among

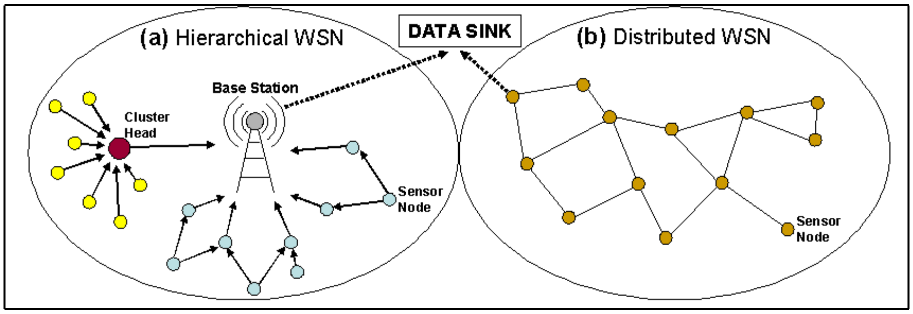


Fig. 1. Network Models: Hierarchical and Distributed Wireless Sensor Networks.

sensor nodes, (ii) group-wise (multicast) within a cluster of sensor nodes, and (iii) network-wise (broadcast) from base stations to sensor nodes.

A Distributed WSNs (DWSN) is shown in Figure 1(b); there is no fixed infrastructure, and network topology is not known prior to deployment. Sensor nodes are usually randomly scattered all over the target area. Once they are deployed, each sensor node scans its radio coverage area to figure out its neighbors. Data flow in DWSN is similar to data flow in HWSN with a difference that network-wise (broadcast) can be sent by every sensor nodes.

4. SECURITY VULNERABILITIES AND REQUIREMENTS

4.1 Security Vulnerabilities

Wireless nature of communication, lack of infrastructure and uncontrolled environment improve capabilities of adversaries in WSN. Stationary adversaries equipped with powerful computers and communication devices may access whole WSN from a remote location. They can gain mobility by using powerful laptops, batteries and antennas, and move around or within the WSN. Also, adversaries can plant their own sensor nodes, base stations or cluster heads in uncontrolled environments. They can replace, compromise or physically damage existing ones. Wireless communication helps adversaries to perform variety of passive, active and stealth type of attacks [Jakobsson et al. 2003]. In passive mode, adversaries silently listen to radio channels to capture data, security credentials, or to collect enough information to derive the credentials. In active attacks, adversaries may actively intercept key management systems, capture and read the contents of sensor nodes. They can use wireless devices with various capabilities to play man-in-the-middle or to hijack a session. They can insert, modify, replay or delete the traffic, jam a part of or whole network [Karlof and Wagner 2003].

Base stations are usually trust centers and store information such as security credentials, sensor readings and routing tables. Thus, compromise of one or more of them can render the entire network useless. Similarly, cluster heads, which are ordinary sensor nodes, are the places where the sensor readings are merged together. Also they are accepted as trusted components and sensor nodes rely on routing information from them.

Content of data flowing in a WSN can be classified into four categories: (i) sensor readings, (ii) mobile code, (iii) key management, and (iv) location information. In addition to active and passive attacks on key management traffic, adversaries may improve their capabilities by accessing mobile codes and location information. An adversary can insert a malicious mobile code which might spread to whole WSN, potentially compromising its security. It can use the location information to locate critical nodes, capture and read their security contents [Jakobsson et al. 2003].

4.2 Security Requirements

Wireless networks are more vulnerable to attacks then wired ones due to broadcast nature of transmission medium, resource limitation on sensor nodes and uncontrolled environments where they are left unattended. Security requirements in WSNs are similar to those of ad-hoc networks [Zhou and Haas 1999], [Stajano and Anderson 1999] due to similarities between MANET and WSN. Thus, WSNs also have following general security requirements:

—*Availability*: ensuring that service offered by whole WSN, by any part of it, or by a single sensor node must be available whenever required,

—*Authentication*: authenticating other nodes, cluster heads, and base stations before granting a limited resource, or revealing information,

—*Integrity*: ensuring that message or the entity under consideration is not altered,

—*Confidentiality*: providing privacy of the wireless communication channels to prevent eavesdropping,

—*Non-reputation*: preventing malicious nodes to hide their activities.

In addition to these general requirements, WSNs have following specific requirements:

—*Survivability*: ability to provide a minimum level of service in the presence of power loss, failures or attacks,

—*Degradation of security services*: ability to change security level as resource availability changes.

These security requirements can be provided by a key distribution mechanism with the requirements given below. These are also used as metrics throughout the paper to evaluate key distribution solutions.

—*Scalability*: ability to support larger networks. Key distribution mechanism must support large networks, and must be flexible against substantial increase in the size of the network even after deployment,

—*Efficiency*: storage, processing and communication limitations on sensor nodes must be considered,

  —*Storage complexity*: amount of memory required to store security credentials. ,

  —*Processing complexity*: amount of processor cycles required to establish a key,

  —*Communication complexity*: number of messages exchanged during a key generation process,

—*Key connectivity (probability of key-share)*: probability that two (or more) sensor nodes store the same key or keying material. Enough key connectivity must be provided for a WSN to perform its intended functionality,

—*Resilience*: resistance against node capture. Compromise of security credentials, which are stored on a sensor node or exchanged over radio links, should not reveal information

Problem	Approach	Mechanism	Keying style	Papers
Pair-wise	Probabilistic	Pre-distribution	Random key-chain	C, E, F, J K, N, S
			Pair-wise key	E
	Deterministic	Pre-distribution	Pair-wise key	G, M
			Combinatorial	P, Q
		Dynamic Key Generation	Master key	D, L
			Key matrix	A
			Polynomial	B, G
	Hybrid	Pre-distribution	Combinatorial	P, Q
		Dynamic Key Generation	Key matrix	H, M, R
			Polynomial	I, R
Group-wise	Deterministic	Dyn. Key Gen.	Polynomial	B, R

The papers are: A[Blom 1985], B[Blundo et al. 1992], C[Eschenauer and Gligor 2002], D[Lai et al. 2002], E[Chan et al. 2003], F[Pietro et al. 2003], G[Liu and Ning 2003c], H[Du et al. 2003], I[Liu and Ning 2003b], J[Zhu et al. 2003], K[Du et al. 2004], L[Dutertre et al. 2004], M[Lee and Stinson 2004b], N[Hwang et al. 2004], P[Camtepe and Yener 2004], Q[Lee and Stinson 2004a], R[Huang et al. 2004], S[Hwang and Kim 2004].

Table II. Classification of papers on pair-wise and group-wise key distribution problems in Distributed WSN.

about security of any other links in the WSN. Usually higher resilience means lower number of compromised links.

In general, resource usage, scalability, key connectivity and resilience are conflicting requirements; therefore, trade-offs among these requirements must be carefully observed.

5. KEY DISTRIBUTION IN DISTRIBUTED WSN

In DWSNs, sensor nodes use pre-distributed keys directly, or use keying materials to dynamically generate pair-wise and group-wise keys. Challenge is to find an efficient way of distributing keys and keying materials to sensor nodes prior to deployment. Solutions to key distribution problem in DWSN can use one of the three approaches: (i) probabilistic, (ii) deterministic, or (iii) hybrid. In probabilistic solutions, key-chains are randomly selected from a key-pool and distributed to sensor nodes. In deterministic solutions, deterministic processes are used to design the key-pool and the key-chains to provide better key connectivity. Finally, hybrid solutions use probabilistic approaches on deterministic solutions to improve scalability and resilience. Table II classifies the papers which provide solutions to pair-wise and group-wise key distribution problem in DWSN. Based on this classification, we describe the solutions in Sections 5.1 and 5.2.

5.1 Pair-wise Key Distribution Schemes

Pair-wise key distribution schemes are grouped according to proposed keying styles (i.e. pair-wise key, random key-chain, master key, . . .). Proposed schemes consist of three phases in general: (i) *key setup* prior to deployment, (ii) *shared-key discovery* after deployment, and (iii) *path-key establishment* if two sensor nodes do not share a key.

5.1.1 *Pair-wise key pre-distribution solutions.* The trivial solution in terms of resource usage is to deploy single master key to all sensors. Since, an adversary may capture a node and compromise the key very easily, it has very low resilience. The other extreme is to use distinct pair-wise keys for all possible pairs in the WSN. For a network of size *N*, each sensor S_i $(1 \leq i \leq N)$ stores a key-chain $KC_i = \{K_{i,j} | i \neq j \; and \; 1 \leq j \leq N\}$ of size $N-1$ out of $N(N-1)/2$ distinct keys. Node S_i stores a unique pair-wise key for each one of *N-1* sensor nodes in the WSN. However, not all *N-1* keys are required to be stored in nodes' key-chain to have a connected key graph. Although such an exhaustive solution creates unnecessary storage burden on a sensor node, this solution has very good key resilience.

Random pair-wise key scheme [Chan et al. 2003] addresses unnecessary storage problem, yet provides very good key resilience. It is based on Erdos and Renyi's work. Each sensor node stores a random set of Np pair-wise keys to achieve probability p that two nodes are connected. At *key setup* phase, each node identity is matched with Np other randomly selected node IDs with probability p. A pair-wise key is generated for each ID-pairs, and is stored in both nodes' key-chain along with the ID of other party. Each sensor uses $2Np$ units of memory to store its key-chain. At *shared-key discovery* phase, each node broadcasts its ID; therefore, each node sends one message, and receives one message from each node within its radio range. Neighboring nodes can tell if they share a common pair-wise key. This solution has very good key resilience. It is more scalable in the sense that efficient use of memory spaces helps support larger WSNs. However, it sacrifices key connectivity to decrease the storage usage.

Closest (location-based) pair-wise keys pre-distribution scheme [Liu and Ning 2003c] is an alternative to *Random pair-wise key scheme* [Chan et al. 2003]. It takes advantage of the location information to improve the key connectivity. Sensor nodes are deployed in a two dimensional area, and each sensor has an expected location that can be predicted. The idea is to have each sensor to share pair-wise keys with its c closest neighbors. In *key setup* phase, for each sensor node S_A, a unique key K_A and c closest neighbors $S_{B_1}, \ldots, S_{B_c}$ are selected. For each pair (S_A, S_{B_i}), a pair-wise key $K_{A,B_i} = PRF(K_{B_i} | ID_A)$ is generated. Node S_A stores all pair-wise keys, whereas node S_{B_i} only stores the key K_{B_i} and the PRF. Thus, each sensor uses $2c+1$ units of memory to store its key-chain. With this extension, deployments of new nodes are quite easy. A new node S_A can be preloaded with the pair-wise keys for c sensor nodes in its expected location. Solution decreases memory usage, and preserves a good key connectivity if deployment errors are low. A sensor uses its CPU to search for a pair-wise key, or to generate it with PRF function. Similar to *Random pair-wise key scheme* [Chan et al. 2003], this solution has very good key resilience, and it is scalable.

ID based one-way function scheme (IOS) [Lee and Stinson 2004b] assumes a connected r-regular graph G which has an edge decomposition into star-like subgraphs. Pair-wise keys are distributed according to these subgraphs. A sensor node S_A receives a secret key K_A and secret keys $Hash(K_B | ID_A)$ if S_A is in the star-like graph centered around node S_B. Node S_B can always generate the secret key $Hash(K_B | ID_A)$ by using its secret K_B and public $ID(A)$. In an r-regular graph

G, each sensor node can be center of one and leaf of $r/2$ star-like subgraphs. Thus, each sensor uses $r+1$ units of memory to store keys and key IDs. Solution has very good key resilience, and it permits any pair of nodes to share a key in one or at most two hops.

Multiple IOS [Lee and Stinson 2004b] is proposed to improve scalability of *ID based one-way function scheme (IOS).* Every node in graph G corresponds to ℓ nodes $S_A = S_{A_1}, \ldots, S_{A_\ell}$. Thus, sensor nodes S_{A_i} store a common key K_A and a secret $Hash(K_B|ID_{A_i})$. Every node S_{B_j} in the class of node S_B, can use common key K_B to generate the secret $Hash(K_B|ID_{A_i})$ for node S_{A_i}. *Multiple IOS* decreases memory usage by a factor of ℓ. It sacrifices resilience, because compromise of a class key means compromise of the links of ℓ sensor nodes.

5.1.2 *Master key based key pre-distribution solutions. Broadcast session key negotiation protocol (BROSK)* [Lai et al. 2002] is based on single master key which is pre-deployed to sensor nodes. A pair of sensor nodes (S_i, S_j) exchanges random nonce values. They use master key K_m to establish session key $K_{i,j} = PRF(K_m|RN_i|RN_j)$. Each sensor uses one unit of memory to store the master key. It is possible to derive all link keys once the master key is compromised; therefore the scheme has very low resilience.

Lightweight key management system [Dutertre et al. 2004] proposes a solution with slightly better resilience where more than one master key is employed. It assumes a WSN where groups of sensor nodes are deployed in successive generations of size θ. Each sensor node stores a group authentication key bk_1 and a key generation key bk_2. If two sensor nodes S_A and S_B are from the same generation, they authenticate each other by using the authentication key bk_1. They exchange random nonce values RN_A and RN_B, and establish the session key $K_{A,B} = PRF(bk_2|RN_A|RN_B)$. It is possible that nodes are from two different generations. A sensor node S_A, of an old generation i, stores a random nonce RN_A and a secret $S_{A,j}$ for each new generation j. Secret $S_{A,j}$ is used to authenticate sensor nodes from new generation j. Node S_B of new generation j can authenticate itself by generating the secret $S_{A,j} = PRF(gk_j|RN_A)$ given RN_A. Secret gk_j is only known to nodes of new generation j. Once authenticated, both parties use $S_{A,j}$ as the key generation key to generate the pair-wise key $K_{A,B}$. If there are g such generations, each sensor needs at most $4+2g$ units of memory to store the keys. Resilience of the scheme is still low because an adversary only needs to compromise the secrets bk_1, bk_2 and gk_j of generation j to compromise all the links of nodes in generation j. Furthermore, adversary may log the messages flowing in the network to process later when the required credentials are compromised completely.

5.1.3 *Random key-chain based key pre-distribution solutions.* Original solution is provided by *Basic probabilistic key pre-distribution scheme* [Eschenauer and Gligor 2002] which relies on probabilistic key sharing among the nodes of a random graph. In *key setup* phase, a large key-pool of KP keys and their identities are generated. For each sensor, k keys are randomly drawn from the key-pool KP without replacement. These k keys and their identities form the key-chain for a sensor node. Thus, probability of key share among two sensor nodes becomes $p = \frac{((KP-k)!)^2}{((KP-2k)!KP!)}$. In *shared-key discovery* phase, two neighbor nodes exchange and compare list of identities of keys in their key-chains. Basically, each sensor node broadcasts one message, and receives one message from each node within its radio range where messages carry key ID list of size k. *Cluster key grouping scheme* [Hwang et al. 2004] proposes to divide key-chains into C clusters where each cluster has a *start key ID*. Remaining key IDs within the cluster are implicitly known from the *start key ID*. Thus, only start key IDs for clusters are broadcasted during *shared-key discovery* phase which means messages carry key ID list of size c instead of k. Another solution is given by *Pair-wise key establishment protocol* [Zhu et al. 2003] which requires every sensor node to have a unique ID which is used as a seed to a PRF. Key IDs for the keys in the key-chain of node S_A are generated by $PRF(ID_A)$. Thus, broadcast messages carry only one key ID. Also, storage, which is required to buffer received broadcast message before processing, decreases substantially. But, a sensor node has to execute $PRF(ID)$ for each broadcast message received from a neighbor. *Transmission range adjustment scheme* [Hwang and Kim 2004] proposes sensor nodes to increase their transmission ranges during *shared-key discovery* phase. Nodes return to their original optimal transmission range once the keys are discovered. Idea is to decrease communication burden in *path-key establishment* phase, and to save energy while still providing a good key connectivity. It is possible to protect key identities broadcasted in *shared-key discovery* by using a method similar to *Merkle Puzzle* [Merkle 1978] which substantially increases processing and communication usage. After *shared-key discovery* phase, some node pairs may not be able to find a key in common. These pairs apply *path-key establishment* phase to communicate securely through other nodes. Scalability and resilience of the solutions can be improved by using larger key pools. But, larger key-pool means smaller probability of key share because key-chain size may not increase due to storage limitations. Probability that a link is compromised, when a sensor node is captured, is k/KP which is very high for small key-pools, and produces low resilience.

There are several key reinforcement proposals to strengthen security of the established link keys, and improve resilience. Objective is to securely generate a unique link- or path-key by using established keys, so that the key is not compromised when one or more sensor node is captured. One approach is to increase amount of key overlap required in *shared-key discovery* phase. *Q-composite random key pre-distribution scheme* [Chan et al. 2003] requires q common keys to establish a link key. Link key $K_{A,B}$ between a pair of sensor nodes S_A and S_B is set as hash of all common keys $K_{A,B} = Hash(K_1||K_2||K_3||\ldots||K_q)$. The scheme improves resilience because probability that a link is compromised, when a sensor node is captured, decreases from k/KP to $\binom{k}{q}/\binom{KP}{q}$. But, probability of key sharing also decreases because a pair of nodes has to share q keys instead of one. Another approach is to reinforce the established link key. In *Multi-path key reinforcement scheme* [Chan et al. 2003], node S_A generates j random key updates rk_i and sends them through j disjoint secure paths. S_B can generate reinforced link key $K^r_{A,B} = K_{A,B} \oplus rk_1 \oplus \ldots \oplus rk_j$ upon receiving all key updates. This approach requires nodes S_A and S_B to send and receive j more messages each of which carries a key update. Moreover, each node on the j disjoint path has to send and receive an extra message. Similar mechanism is proposed by *Pair-wise* *key establishment protocol* [Zhu et al. 2003] which uses *threshold secret sharing* for key reinforcement. S_A generates a secret key $K^r_{A,B}$, $j-1$ random shares sk_i, and $sk_j = K^r_{A,B} \oplus sk_1 \oplus \ldots \oplus sk_{j-1}$. S_A sends the shares through j disjoint secure paths. S_B can recover $K^r_{A,B}$ upon receiving all shares. In *Co-operative pair-wise key establishment protocol* [Pietro et al. 2003], S_A first chooses a set $C = \{c_1, c_2, \ldots, c_m\}$ of co-operative nodes. A co-operative node provides a hash $HMAC(K_{c_1,B}, ID_A)$. Reinforced key is then $K^r_{A,B} = K_{A,B} \oplus \left(\bigoplus_{c \in C} HMAC(K_{c,B}, ID_A)\right)$ where $K_{A,B}$ and $K_{c,B}$ are the established link keys. Node S_A shares set C with node S_B; therefore, S_B can generate the same key. This approach requires nodes S_A and S_B to send and receive c more messages. Moreover, cooperative nodes have to send and receive two extra messages. In addition to increased communication cost, each cooperative node has to execute $HMAC$ function twice for S_A and S_B. The key reinforcement solutions in general increase processing and communication complexity, but provide good resilience in the sense that a compromised key-chain does not directly affect security of any links in the WSN. But, it may be possible for an adversary to recover initial link keys. An adversary can then recover reinforced link keys from the recorded multi-path reinforcement messages when the link keys are compromised.

Sensor nodes, which are far away from each other, do not need to have common keys in their key-chains. Similar to *Closest pair-wise keys pre-distribution scheme* [Liu and Ning 2003c] (as we explained in Section 5.1.1), *Key pre-distribution by using deployment knowledge scheme* [Du et al. 2004] uses location information. It models a deployment knowledge and develops a key pre-distribution scheme based on the model. The scheme divides sensor nodes into $t \times n$ groups $G_{i,j}$ and deploys them at a resident point (x_i, y_j) for $1 \leq i \leq t$ and $1 \leq j \leq n$ where the points are arranged as two dimensional grids. Resident points of a node $m \in G_{i,j}$ follows the pdf $f^{i,j}_m(x, y | m \in G_{i,j}) = f(x - x_i, y - y_j)$ where $f(x, y)$ is a two dimensional Gaussian distribution. In *key setup* phase, key-pool KP is divided into $t \times n$ key-pools $KP_{i,j}$ of size $\omega_{i,j}$. The pool $KP_{i,j}$ is used as key-pool for the nodes in group $G_{i,j}$. Given $\omega_{i,j}$ and overlapping factors α and β, key-pool is divided into subsets as summarized in Figure 2 where (i) two horizontally and vertically neighboring key-pools have $\alpha \times \omega_{i,j}$ keys in common, (ii) two diagonally neighboring key-pools have $\beta \times \omega_{i,j}$ keys in common, and (iii) non-neighboring key-pools do not share a key. *Basic probabilistic key pre-distribution scheme* is applied within each group. Problem in this scheme is the difficulty to decide on parameters $\omega_{i,j}$, α and β to provide a good key connectivity.

5.1.4 *Combinatorial design based key pre-distribution solutions.* Key sharing probability among the sensor nodes can be increased by *designing* the key-chains. *Combinatorial design based pair-wise key pre-distribution scheme* [Camtepe and Yener 2004] is based on block design techniques in combinatorial design theory. It employs *symmetric* and *generalized quadrangles* design techniques. The scheme uses *finite projective plane* of order n (for prime power n) to generate a *symmetric design* (or symmetric BIBD) with parameters $(n^2+n+1, n+1, 1)$. Design supports n^2+n+1 nodes, and uses key-pool of size n^2+n+1. It generates n^2+n+1 key-chains of size $n+1$ where every pair of key-chains has exactly one key in common, and every key appears in exactly $n+1$ key-chains. After the deployment, every pair of nodes finds exactly one common key. Thus, probability of key sharing

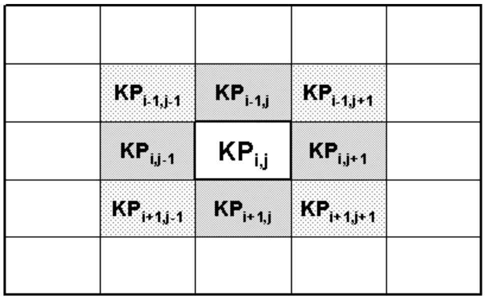


Fig. 2. Key pre-distribution with deployment knowledge where key-pool $KP_{i,j}$ has: (i) $\alpha \times \omega_{i,j}$ keys in common with key-pools $KP_{i-1,j}$, $KP_{i,j-1}$, $KP_{i,j+1}$ and $KP_{i+1,j}$, (ii) $\beta \times \omega_{i,j}$ keys in common with key-pools $KP_{i-1,j-1}$, $KP_{i-1,j+1}$, $KP_{i+1,j-1}$ and $KP_{i+1,j+1}$, and (iii) zero keys in common with others.

among a pair of sensor node is 1. Probability that a link is compromised, when a sensor node is captured, is $\approx 1/n$. Disadvantage of this solution is that, parameter n has to be a prime power; therefore, not all network sizes can be supported for a fixed key-chain size. More scalable solutions can be provided by using *generalized quadrangles* design with the property that not all pairs of neighboring nodes need to share a key directly. In GQ, a pair of key-chains may not have a key in common, but GQ guarantees that there are other key-chains which share exactly one key with both. Proposed GQ designs, $GQ(n,n)$, $GQ(n,n^2)$ and $GQ(n^2,n^3)$, support network sizes of orders $O(n^3)$, $O(n^5)$ and $O(n^4)$ in key-chain size, and provides key sharing probabilities of $\approx 1/n$, $\approx 1/n^2$ and $\approx 1/n^{1.5}$ respectively [Camtepe and Yener 2004]. Although GQ is more scalable than symmetric design, parameter n still needs to be a prime power. Combinatorial design techniques are used along with probabilistic approaches yielding *hybrid designs* to support arbitrary network sizes. *Hybrid design* first generates core symmetric or GQ design of size M for a given target network size of N where $M < N$ as summarized in Figure 3. *Complementary design* of the core design is generated for the remaining $N - M$ key-chains. *Complementary design* is complements of the core design key-chains $\overline{KC_i} = KP \backslash KC_i$ in the key-pool KP. *Hybrid design* then randomly selects key-chains KC'_i of size k among k-subsets of $\overline{KC_i}$. *Hybrid design* improves scalability and resilience, but sacrifices key sharing probability of the core symmetric or GQ design. Very similar approaches based on combinatorial design theory are proposed in [Lee and Stinson 2004a].

5.1.5 *Key matrix based dynamic key generation solutions.* All possible link keys in a network of size N can be represented as an $N \times N$ key matrix. It is possible to store small amount of information to each sensor node, so that every pair of nodes can calculate corresponding field of the matrix, and uses it as the link key. *Blom's scheme* [Blom 1985] uses a public $(\lambda + 1) \times N$ matrix G and a private $N \times (\lambda + 1)$ matrix D which is generated over GF(q) and where N is size of the network. Solution is λ-secure, meaning that keys are secure if no more than λ nodes are compromised. Matrix G must have $(\lambda + 1)$ linearly independent columns (i.e. Vandermonde matrix) to provide λ-secure property. Key matrix is then defined as a

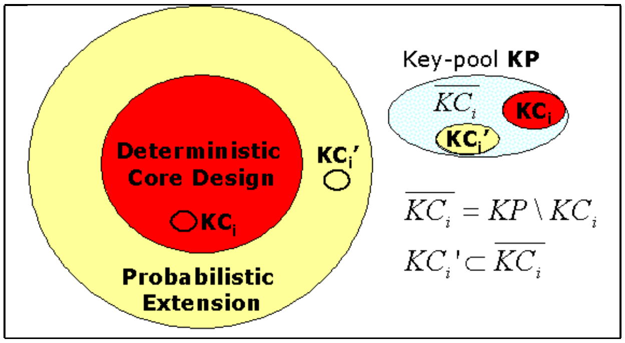

Deterministic Core Design
KC_i
KC_i'
Probabilistic Extension
Key-pool **KP**
$\overline{KC_i}$
KC_i
KC_i'
$\overline{KC_i} = KP \setminus KC_i$
$KC_i' \subset \overline{KC_i}$

Fig. 3. Hybrid design with a symmetric (or GQ) core of size M and a probabilistic extension of size $N - M$. Key-chains KC_i' of probabilistic extension are randomly selected among k-subsets of $\overline{KC_i}$ which are complements of core symmetric (or GQ) design key-chains KC_i.

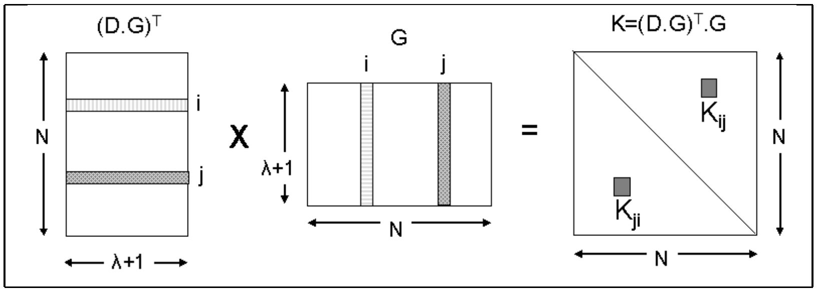


Fig. 4. Blom's scheme. Sensor node S_i stores $column_i$ from matrix G as public information, and row_i from matrix $(D.G)^T$ as private information. Nodes S_i and S_j exchange their public column vectors and generate $K_{ij} = row_i \times column_j$ and $K_{ji} = row_j \times column_i$ respectively where $K_{ij} = K_{ji}$

symmetric matrix $K = (D.G)^T.G$. Sensor node S_i stores $column_i$ of size $\lambda+1$ from matrix G as public information, and row_i of size $\lambda+1$ from matrix $(D.G)^T$ as private information. A pair of sensor nodes (S_i, S_j), first exchange their public information $column_i$ and $column_j$. The link key is then generated as $K_{ij} = row_i \times column_j$ and $K_{ji} = row_j \times column_i$ respectively as summarized in Figure 4. The scheme requires costly multiplication of two vectors of size $\lambda + 1$ where the elements are as large as the corresponding cryptographic key size. Each sensor node broadcasts one message, and receives one message from each node within its radio range where messages carry a vector of size $\lambda + 1$.

Multiple space key pre-distribution scheme [Du et al. 2003] improves the resilience of Blom's scheme [Blom 1985]. It uses a public matrix G and a set of ω private matrices D. These matrices form ω spaces (D_i, G) for $i = 1, \ldots, \omega$. For each sensor node, a set of τ spaces are randomly selected among these ω spaces. Required keying materials for each selected space are stored to the sensor node as in *Blom's scheme*; therefore, each sensor node stores $\tau + 1$ vectors of size $\lambda + 1$. In *shared key discovery* phase, a pair of nodes first agrees on a common space for which nodes has to exchange an extra message which includes τ space IDs. It is possible that a pair of nodes does not share a common space, in that case they have to apply *path-key establishment* phase to establish a key through intermediate nodes.

Scalability of *Blom's scheme* is improved in *Multiple space Blom's scheme (MBS)* [Lee and Stinson 2004b]. The scheme divides nodes into two sets U and V to form bipartite key connectivity graph. That means, not every pair of nodes has to share a key. Another difference from Blom's scheme is that private matrix D is not necessarily symmetric. Secret information $column_u^T D$ is assigned to each node $S_u \in U$ and $Dcolumn_v$ is assigned to each node $S_v \in V$. Nodes S_u and S_v also store public information $column_u$ and $column_v$ respectively. Nodes can exchange their public information to calculate secret key $column_u^T Dcolumn_v$. Larger networks are supported by *Deterministic multiple space Blom's scheme (DMBS)* [Lee and Stinson 2004b] where ℓ copies of strongly regular (regular of degree r) graph R are used. Each vertex of R can be considered as a class of ℓ nodes such as $S_u = S_{u_1}, \ldots, S_{u_\ell}$. An arbitrary direction is assigned to every edge in R, and every edge e has a random private matrix D_e which is not necessarily symmetric. Each sensor node S_{u_i} receives its public column vector $column_u$ of size $\lambda + 1$. For a directed edge $(S_{u_i}, S_{v_j}) \in R$, source node S_{u_i} receives secret information $column_u^T D_{uv}$ of size $\lambda + 1$, and destination node S_{v_j} receives secret information $D_{uv}column_v$ of size $\lambda + 1$. Thus, each node stores vectors of size $r(\lambda + 1)$. Nodes S_{u_i} and S_{v_j} can then generate the link key as $K_{u_i,v_j} = column_u^T D_{uv} column_v$. *DMBS* increases scalability with the cost of decreased resilience because capture of one sensor node compromises credentials of $\ell - 1$ other.

5.1.6 *Polynomial based dynamic key generation solutions. Polynomial based key pre-distribution scheme* [Blundo et al. 1992] distributes a polynomial share (a partially evaluated polynomial) to each sensor node by using which every pair of nodes can generate a link key. Symmetric polynomial $P(x, y)$ $(P(x, y) = P(y, x))$ of degree λ is used. The coefficients of the polynomial come from $GF(q)$ for sufficiently large prime q. Each sensor node stores a polynomial with $\lambda + 1$ coefficients which come from GF(q). Sensor node S_i receives its polynomial share of $f_i(y) = P(i, y)$. S_i (resp. S_j) can obtain link key $K_{i,j} = P(i, j)$ by evaluating its polynomial share $f_i(y)$ (resp. $f_j(y)$) at point j (resp. i). Every pair of sensor nodes can establish a key. The solution is λ-secure, meaning that coalition of less than $\lambda + 1$ sensor nodes knows nothing about pair-wise keys of others.

Polynomial pool-based key pre-distribution scheme [Liu and Ning 2003b] considers the fact that not all pairs of sensor nodes have to establish a key. It combines *Polynomial based key pre-distribution scheme* [Blundo et al. 1992] with the key-pool idea in [Eschenauer and Gligor 2002; Chan et al. 2003] to improve resilience and scalability. For *key setup* phase, a set F of λ-degree polynomials over finite field GF(q) is generated. Each sensor node S_i receives a subset F_i of the polynomial set F $(F_i \subseteq F)$. There are several ways to select polynomial subsets for sensor nodes. In one approach, along with the polynomial subset, each sensor stores list of sensor ID's with which it shares the polynomial. In another approach, a grid-based key pre-distribution scheme is employed. For a network of size N, $m \times m$ (for $m = \lceil \sqrt{N} \rceil$) grid with a set of $2 \times m$ column and row polynomials $\{f_i^c(x, y), f_i^r(x, y)\}$ $(i = 0, \ldots, m-1)$ are generated. Each row i in grid is associated with a polynomial

$f_i^r(x,y)$ and each column i with a polynomial $f_i^c(x,y)$. Each sensor is assigned to a coordinate (i,j) on the grid, and receives polynomials $\{f_i^c(x,y), f_j^r(x,y)\}$. A pair of sensor nodes only needs to check whether their column or row addresses overlap. In *shared-key discovery* phase, if two sensor nodes have the same polynomial, they can establish a key.

Location information can help provide better key connectivity. Similar to *Closest pair-wise keys pre-distribution scheme* [Liu and Ning 2003c] and *Key pre-distribution by using deployment knowledge scheme* [Du et al. 2004] (as we explained in Sections 5.1.1 and 5.1.3 respectively), *Location-based pair-wise keys scheme using bivariate polynomials* [Liu and Ning 2003c] uses location information where deployment area is divided into R rows and C columns, total of $R \times C$ cells. The scheme is based on *Polynomial based key pre-distribution scheme* [Blundo et al. 1992]. For each cell at c^{th} column and r^{th} row, a unique polynomial $f_{c,r}(x,y)$ is generated. Each sensor node stores polynomial share of its home cell and four immediate neighbor cells, total of five polynomials. Two sensor nodes simply exchange their cell coordinates to agree on a polynomial share. Similarly, *Grid-group deployment scheme* [Huang et al. 2004] divides deployment area into cells over which groups of sensor nodes are uniformly distributed. *Polynomial pool-based key pre-distribution* [Liu and Ning 2003b] and *Multiple space key pre-distribution* [Du et al. 2003](as we explained in Section 5.1.5) schemes are used to distribute pair-wise keys to a group of sensor nodes located within a cell. Also, every sensor node selects exactly one sensor from each neighboring cell, and shares a pairwise key with it.

5.2 Group-wise Key Distribution Schemes

Straightforward approach is to use existing pair-wise keys to establish group-wise keys. For example, *Lightweight key management system* [Dutertre et al. 2004] considers a WSN where group of sensor nodes are deployed in different phases. It proposes to distribute group-wise keys through the links which are secured with pair-wise keys. Yet another approach is to pre-distribute polynomial shares to sensor nodes by using which group members can generate a common group key. *Polynomial based key pre-distribution scheme* [Blundo et al. 1992] proposes two models. The first model is a non-interactive model where users compute a common key without any interaction. A random symmetric polynomial $P(x_1,\dots,x_t)$ in t variables of degree λ is selected initially where the coefficients come from $GF(q)$ for prime q which is large enough to accommodate the key length of the underlying cryptosystem. Each user S_i receives share $P_i(x_2,\dots,x_t) = P(i,x_2,\dots,x_t)$. Users $S_{j_1},\dots,S_{j_t}$ can generate the conference key $K_{j_1,\dots,j_t}$ by evaluating their polynomial shares. Each user S_{j_i} can evaluate $P_{j_i}(j_1,\dots,j_{i-1},j_{i+1},\dots,j_t)$ and obtain the conference key $K_{j_1,\dots,j_t}$ independently. In the second interactive model, interaction is allowed in key computation. Polynomial $P(x,y)$ of degree $(\lambda+t-2)$ is selected initially. Each user S_i receives share $P_i(y) = P(i,y)$. Users $S_{j_1},\dots,S_{j_t}$ can calculate the conference key $K_{j_1,\dots,j_t}$ as follows: (i) S_{j_t} selects a random key K, (ii) S_{j_t} calculates $K_{j_t,j_\ell} = P_{j_t}(j_\ell) = P(j_t,j_\ell)$ for each $\ell = 1,\dots,t-1$, (iii) S_{j_t} sends $\chi_\ell = K_{j_t,j_\ell} \oplus K$ to each S_{j_ℓ} for $(\ell = 1,\dots,t-1)$, and (iv) each S_{j_ℓ} generates $K_{j_\ell,j_t} = P_{j_\ell}(j_t)$, and derives the secret $K = \chi_\ell \oplus K_{j_\ell,j_t}$. Sensor node S_{j_t} performs $t-1$ polynomial evaluations, and sends $t-1$ messages which carry a single χ value to establish a group-wise key.

Problem	Keying style	Papers
Pair-wise	BS oriented	d, i, k, l, n
	Master key	g, n
Group-wise	Asymmetric keys	a, c, e
	Symmetric keys	n
Network-wise	Master key	f
	TESLA based	b, d, g, h, j, k, m, n

The papers are: a[Burmester and Desmedt 1994], b[Perrig et al. 2000], c[Steiner et al. 2000], d[Chen et al. 2000], e[Carman et al. 2002], f[Slijepcevic et al. 2002], g[Perrig et al. 2002], h[Staddon et al. 2002], i[Undercoffer et al. 2002], j[Liu and Ning 2003a; 2003d], k[Deng et al. 2003a; 2003b], l[Law et al. 2003], m[Bohge and Trappe 2003], n[Zhu et al. 2003].

Table III. Classification of solutions on pair-wise, group-wise and network-wise key distribution problems in Hierarchical WSN.

6. KEY DISTRIBUTION IN HIERARCHICAL WSN

In Hierarchical WSN, there are one or more computationally robust base stations which may act like a key distribution center. Initially, base stations may share a distinct pair-wise key with each sensor nodes. These keys can be used to secure establishment process of other keys. Table III classifies the papers which provide solutions to pair-wise, group-wise and network-wise key distribution problem in HWSN. Based on this classification, we describe the solutions in Sections 6.1, 6.2 and 6.3.

6.1 Pair-wise Key Distribution Schemes

In hierarchical WSNs, base station to sensor node, or sensor node to base station unicast communications require pair-wise keys. Solution for such environments is straightforward; base station can share a distinct pair-wise key with each sensor node. Very similar solutions are proposed in *Perimeter protection scenario* [Undercoffer et al. 2002], *Base station authentication protocols* [Chen et al. 2000; Deng et al. 2003a; 2003b], and *Localized encryption and authentication protocol (LEAP)* [Zhu et al. 2003]. Since the base station shares pair-wise keys with sensor nodes, it can intermediate establishment of a pair-wise key between any pair of sensor nodes. Similar approach is used in *ESA* [Law et al. 2003] where sensor nodes are separated into domains which are supervised by base stations. *SNEP* [Perrig et al. 2002] proposes each pair of communicating party S_A and S_B to share a master secret key $\chi_{A,B}$ and a PRF. S_A and S_B can then generate encryption keys $K_{A,B} = PRF(\chi_{A,B}, 1)$ and $K_{B,A} = PRF(\chi_{A,B}, 3)$, and MAC keys $K'_{A,B} = PRF(\chi_{A,B}, 2)$ and $K'_{B,A} = PRF(\chi_{A,B}, 4)$.

Localized encryption and authentication protocol (LEAP) [Zhu et al. 2003] proposes that each sensor node can establishes pair-wise keys with its immediate neighbor. In the *key setup* phase, nodes receive a general key K_I. A node S_u can use K_I and one-way hash function H to generate its master key $K_u = H_{K_I}(ID_u)$. In *shared key discovery* phase, node S_u broadcasts (ID_u, RN_u) and a neighbor S_v responds with $(ID_v, MAC_{K_v}(RN_u|ID_v))$. Node S_u can then generate the key $K_v = H_{K_I}(ID_v)$, and both nodes S_u and S_v can generate the session key $K_{u,v} = H_{K_v}(ID_u)$. Multi-hop pair-wise keys may be required to reach cluster heads. In that case, node S_u generates secret $K_{u,c}$, and finds m intermediate nodes. It divides the secret into shares $K_{u,c} = sk_1 \oplus sk_2 \oplus \ldots sk_m$, and sends each share through a separate intermediate node S_{v_i} $(1 \leq i \leq m)$. Basically, node S_u sends $ENC_{K_{u,v_i}}(sk_i), H_{sk_i}(0)$ to node S_{v_i}, and S_{v_i} sends $ENC_{K_{v_i,c}}(sk_i), H_{sk_i}(0)$ to cluster head S_c. Solution has high communication cost because S_u sends m messages through m intermediate nodes to increase resilience. However, security of the system depends on the general key K_I which can be compromised by capture of a sensor node. It is possible to compromise all the session keys generated by LEAP once K_I is compromised.

6.2 Group-wise Key Distribution Schemes

In hierarchical WSNs, sensor nodes require group-wise keys to secure multicast messages. One approach is to use secure but costly asymmetric cryptography. Burmester-Desmedt [Burmester and Desmedt 1994] and IKA2 [Steiner et al. 2000] use a Diffie-Hellman based group key transport protocol. These two algorithms are further improved by ID-STAR [Carman et al. 2002]. ID-STAR uses *Identity based cryptography* [Shamir 1984; Boneh and Franklin 2001] where sensor nodes' public keys can be derived from their identities. It is also possible to use existing pair-wise key structure to establish groups-wise keys. In an hierarchical network, where a base station share pair-wise keys with all the sensor nodes, base station can intermediate establishment of group-wise keys. *Localized encryption and authentication protocol (LEAP)* [Zhu et al. 2003] provides a mechanism to generate group-wise keys which follows LEAP pair-wise key establishment phase. Node S_u, who wants to establish a group key with all its neighbors $S_{v_1}, S_{v_2}, \ldots, S_{v_m}$, first generates a unique group key K_u^g. It then sends K_u^g to its neighbors S_{v_i} as $ENC_{K_{u,v_i}}(K_u^g)$. Security of the scheme depends on security of the pair-wise keys which in turn has very low resilience.

6.3 Network-wise Key Distribution Schemes

6.3.1 *Master key based solutions.* In hierarchical WSNs, base station to sensor node broadcast traffic is secured with network-wise keys. An insecure approach is to pre-distribute a single network-wise key to all sensor nodes. Another approach is proposed by *Multi-tiered security solution* [Slijepcevic et al. 2002] where data items are protected to a degree consistent with their value. It considers three types of data flowing in WSN: mobile code, locations of sensors nodes and application data. It is assumed that sensor nodes are initially loaded with a list of m master keys, a PRF and a seed. They use the PRF with the seed to obtain an index within the list of master keys. Selected key is named as *active master key*, and used to secure communication. *RC6* is used as encryption algorithm. Three security levels are defined. In level I, a strong encryption algorithm and active master key is used to secure mobile codes. In level II, sensors are divided into cells. A common location security key is generated within each cell, and used to secure location information. Finally in level III, MD5 hash of the active master key is used to secure application data. Problem with this scheme is that public credentials, such as master key list, PRF and seed, are subject to compromise.

6.3.2 *TESLA based solutions. Timed Efficient Stream Loss-tolerant Authentication (TESLA)* [Perrig et al. 2000] is a multicast stream authentication protocol. *TESLA* uses *delayed key disclosure mechanism* where the key used to authenticate i^{th} message is disclosed along with $(i+1)^{th}$ message. *SPINS* [Perrig et al. 2002] uses $\mu - TESLA$ which is an adoption of TESLA for HWSNs. *SPINS* employs base station as key distribution center. $\mu - TESLA$ provides authentication for data broadcasts, and requires that base station and sensor nodes be loosely time synchronized. Basically, base station (BS) randomly selects last key K_n of a chain, and applies one-way public function H to generate the rest of the chain K_0, K_1, ..., K_{n-1} as $K_i = H(K_{i+1})$. Given K_i, every sensor node can generate the sequence K_0, K_1, ..., K_{i-1}. However, given K_i, no one can generate K_{i+1}. At i^{th} time slot, BS sends authenticated message $MAC_{K_i}(Message)$. Sensor nodes store the message until BS discloses the verification key in $(i+1)^{th}$ time slot. Sensor nodes can verify disclosed verification key K_{i+1} by using the previous key K_i as $K_i = H(K_{i+1})$. In $\mu - TESLA$, nodes are required to store a message until the authentication key is disclosed. This operation may create storage problems, and encourages DoS types of attacks. An adversary may jam key disclosure messages to saturate storages of sensor nodes. $\mu - TESLA$ requires sensor nodes to bootstrap from the BS; that is, they receive the first key of the chain which is called *key chain commitment*. Bootstrapping procedure requires unicast communication, and can be secured with pair-wise keys. Also, $\mu - TESLA$ is used in [Chen et al. 2000; Deng et al. 2003a; 2003b] to authenticate message broadcasts from BS, in [Staddon et al. 2002] to authenticate route update broadcasts, and in *LEAP* [Zhu et al. 2003] to update pre-deployed network-wise keys in case of a node compromise. Another variant of TESLA is *TESLA Certificate* [Bohge and Trappe 2003] where a base station is used as certificate authority (CA). In this scheme, CA generates certificate $Cert(ID_A, t_{i+d}, ..., MAC_{K_i}(...))$ for sensor node S_A at time t_i. It discloses the TESLA key K_i at time t_{i+d} when the certificate expires.

Bootstrapping of key chain commitments in $\mu - TESLA$ causes high volume of packets flowing in WSN, and creates scalability problems. $\mu - TESLA$ *extensions* [Liu and Ning 2003a; 2003d] propose five extensions to address scalability issues. In *predetermined key chain commitment*, commitment is pre-distributed to sensors before the deployment. In this solution, key chain must cover lifetime of sensor nodes to prevent bootstrapping requirements. This can be achieved by using either long chains or large time intervals. A new coming node has to generate whole key chain from the beginning to authenticate recently disclosed key. Thus, long key chain means excessive processing for sensor nodes which are deployed at a later time. Large time interval means increased number of messages to store because sensor nodes have to store incoming messages until the authentication key is disclosed. *Two-level key chains scheme* tries to address these problems. There is a high-level key chain with long enough time interval to cover the life time of sensor nodes, and multiple low-level key-chains with short enough intervals as shown in figure 5(a). High-level key chain is used to distribute and authenticate randomly generated commitments of low-level key-chains. In this scheme, sensor nodes are initialized with the commitment of high-level chain, time intervals of high-level and low-level key chains and one way functions of high and low-level chains. However,

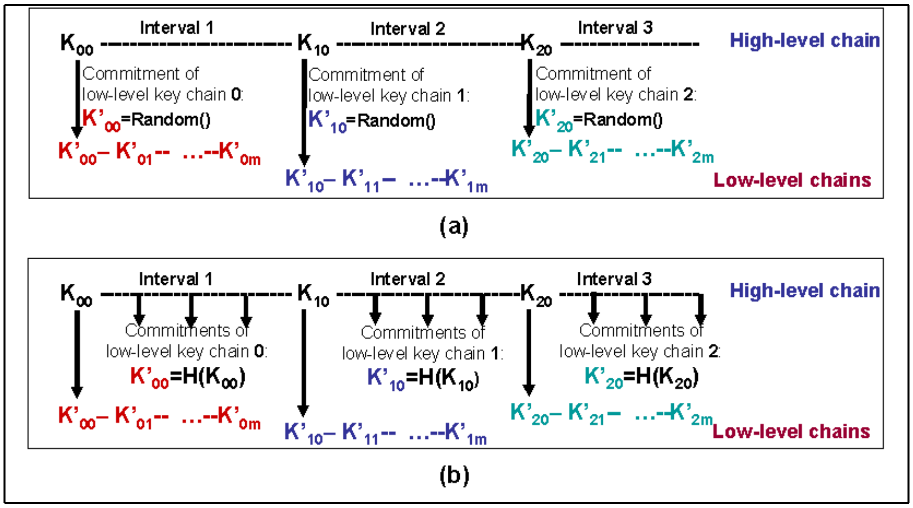


Fig. 5. $\mu - TESLA$ extensions: (a) two-level key chains scheme and (b) fault tolerant two-level key chains scheme. Downward arrows show broadcast of the low-level key commitments for each interval: (a) commitments are broadcasted at the beginning of the interval, (b) commitments are broadcasted periodically throughout the interval.

low-level keys are not chained together. Thus, loss of a low-level key disclosure can only be recovered with a key which is disclosed later within the same interval. Moreover, loss of a low-level key commitment may also mean loss of entire interval. An adversary may take advantage of this, and may jam disclosure of low-level key commitments. *Fault tolerant two-level key-chains scheme* is proposed to address these issues. In this scheme, the commitments of low-level key chains are not randomly generated, but obtained from high-level keys by using another one-way function as shown in figure 5(b). Low-level key commitments are periodically broadcasted; however, an adversary may still recover the commitment period, and can jam disclosure of low-level key commitments. *Fault tolerant two-level key-chains with random commitments scheme* uses a random process to broadcast the low-level commitments. Finally, *multi-level chains scheme* is proposed to provide smaller time intervals and shorter key chains.

7. SUMMARY AND DISCUSSIONS

Figure 6 provides taxonomy of papers on key distribution problems in DWSN and HWSN. In this figure, graphs are DAGs (directed acyclic graphs) where nodes represent papers. Directed edges show predecessor/successor relations among the papers. There is an edge from a paper to another one if latter provides improvement for the solutions proposed by former. Nodes (papers) are ordered over a horizontal time axis according to their publication dates. Vertical axis groups papers under three problems: (i) pair-wise, (ii) group-wise, and (iii) network-wise key distribution problems. Each problem is represented with a specific node, named as *origin node*, which has only outgoing edges. The style of an edge (dotted, dashed, solid) in between two nodes represents the problem in which an improvement is provided.

A paper may provide more than one solution to more than one problem; therefore, corresponding node may be reachable from more than one *origin node*, and there may be more than one edge with different styles in between two nodes.

Detailed evaluation for the edges in Figure 6 is given in Table IV. Solutions corresponding to nodes (papers) of directed edges are compared with each other by considering the six metrics defined in Section 4.2: (i) scalability ”S”, (ii) key connectivity ”K”, (iii) resilience ”R”, (iv) storage complexity ”M”, (v) processing complexity ”P” and (vi) communication complexity ”C”. Comparison results for each metric are presented as ”↑” (increase), ”↓” (decrease) and ”-” (no change). Solutions described in Sections 5 and 6 are summarized in Table V where metric values for each solution are listed.

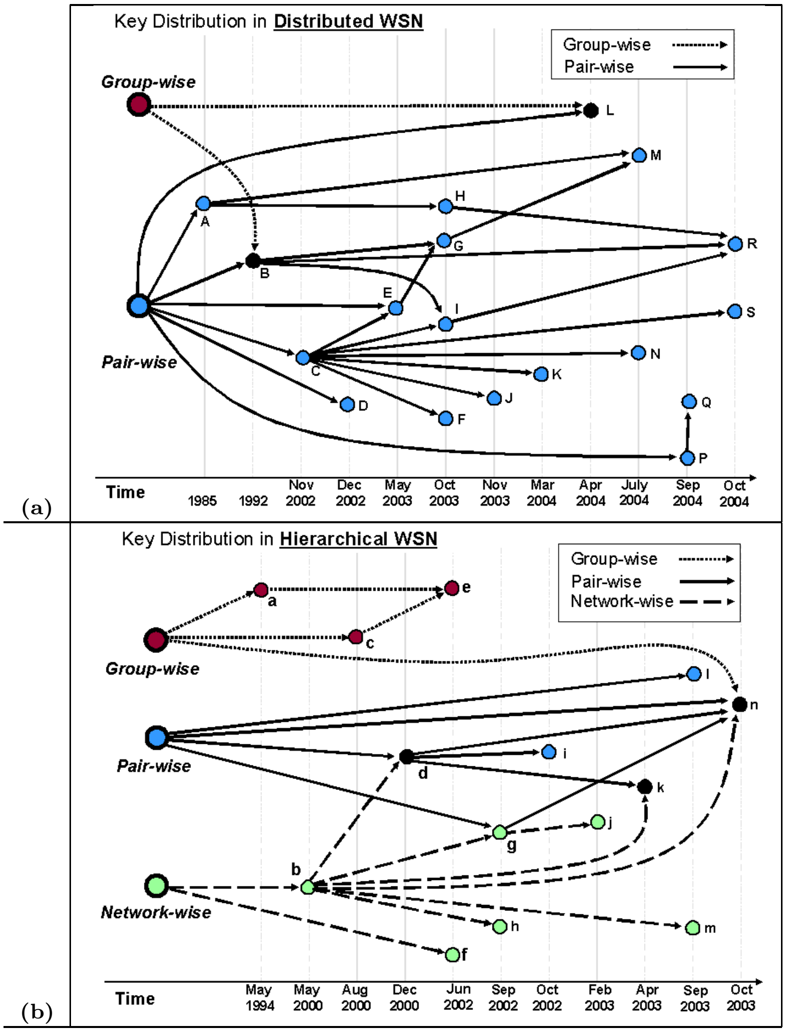
Scalability ”S” is ability to support larger networks. Larger networks can be supported if there is enough storage for the required security credentials which is related to storage complexity of the solution. In Table V, scalability of the similar (same keying problem and keying style) solutions are compared with each other. Basically, each solution is assigned a scalability rank where higher rank means higher scalability. There can be more than one solution sharing the same rank which means that corresponding solutions have roughly the same scalability.

Resilience ”R” of each solution is given as either one of the following ways: (i) probability that a link is compromised when an adversary captures a node, (ii) number of nodes whose security credentials are compromised when an adversary captures a node, or (iii) number of sensor nodes required to be captured to compromise whole WSN. Third one is represented as *n-secure* meaning that it is enough to capture *n+1* nodes to compromise whole WSN. As these values increase, network becomes less secure; therefore, resilience decreases.

Key connectivity ”K” considers probability that two (or more) sensor nodes store the same key or keying material to be able to establish pair-wise, group-wise or network-wise keys.

Efficiency of the solutions is measured with their storage, processing and communication complexities. Storage complexity ”M” is amount of memory units required to store security credentials. We consider key, key ID, node ID, node locations, etc. as one memory unit. Processing complexity ”P” is number of unit functions executed. Unit functions can be: (i) *Search* for one or more key in a key-chain, (ii) functions such as *PRF*, *Hash*, *MAC*, *XOR* and *ENC*, (ii) *VecMul(size)* which multiplies two vectors of given sizes, and (iii) *PolyEval* which evaluates a polynomial at a given point. Communication is the most power consuming operation performed by a sensor node. Communication complexity ”C” is measured as number and size of packets sent and received by a sensor node.

Based on the results shown in Tables IV and V we conclude that there are significant tradeoffs and, there is no one-size-fits-all solution for key distribution problems in WSNs.



The nodes are: A[Blom 1985], B[Blundo et al. 1992], C[Eschenauer and Gligor 2002], D[Lai et al. 2002], E[Chan et al. 2003], F[Pietro et al. 2003], G[Liu and Ning 2003c], H[Du et al. 2003], I[Liu and Ning 2003b], J[Zhu et al. 2003], K[Du et al. 2004], L[Dutertre et al. 2004], M[Lee and Stinson 2004b], N[Hwang et al. 2004], P[Camtepe and Yener 2004], Q[Lee and Stinson 2004a], R[Huang et al. 2004], S[Hwang and Kim 2004], a[Burmester and Desmedt 1994], b[Perrig et al. 2000], c[Steiner et al. 2000], d[Chen et al. 2000], e[Carman et al. 2002], f[Slijepcevic et al. 2002], g[Perrig et al. 2002], h[Staddon et al. 2002], i[Undercoffer et al. 2002], j[Liu and Ning 2003a; 2003d], k[Deng et al. 2003a; 2003b], l[Law et al. 2003], m[Bohge and Trappe 2003], n[Zhu et al. 2003].

Fig. 6. Taxonomy of the papers on key distribution problems in (a) DWSN and (b) HWSN. Graphs are DAGs (directed acyclic graphs) where nodes represent papers, and edges represent predecessor/successor relations (improvements) among solutions provided by the papers. There are three nodes which have only outgoing edges, and which represent the pair-wise, group-wise and network-wise key distribution problems. Style of an edge represents the problem on which destination node (paper) provides improvements.

Edges of figure 6(a)	**S**	**K**	**R**	**M**	**P**	**C**
A → H	↓	↓	↑	↑	-	↑
A → M [a]	↑	↓	↓	↑	-	-
B → G , I, R	↓	↓	-	↑	-	↑
C → E [b], F, J	-	-	↑	-	↑	↑
C → K	-	↑	↓	-	-	-
C → N	-	-	-	-	-	↓
E → G	↑	↑	-	↓	↑	-
G → M [c]	↑	-	↓	↓	-	-

Edges of figure 6(b)	**S**	**K**	**R**	**M**	**P**	**C**
g → j	↑	-	-	↓	↓	↑
g → n	-	-	↓	↑	↑	↑

[a] Deterministic multiple space Blom's scheme is considered

[b] Multi-path key reinforcement scheme is considered

[c] Multiple IOS scheme is considered

The papers are: A[Blom 1985], B[Blundo et al. 1992], C[Eschenauer and Gligor 2002], E[Chan et al. 2003], F[Pietro et al. 2003], G[Liu and Ning 2003c], H[Du et al. 2003], I[Liu and Ning 2003b], J[Zhu et al. 2003], K[Du et al. 2004], M[Lee and Stinson 2004b], N[Hwang et al. 2004], R[Huang et al. 2004], g[Perrig et al. 2002], j[Liu and Ning 2003a; 2003d], n[Zhu et al. 2003].

Table IV. Evaluation of edges in Figure 6. Solutions corresponding to nodes (papers) of directed edges are compared with each other by considering the six metrics defined in Section 4.2: (S)-scalability,(K)-key connectivity, (R)-resilience, (M)-storage complexity, (P)-processing complexity, (C)-communication complexity. A comparison result for a metric is given as ”↑” (increase), ”↓” (decrease) and ”-” (no change). Details of the solutions are given in Table V.

Table V: Evaluation of the solutions. Solutions are grouped, as in Sections 5 and 6, based on the keying problem and style. Citation of the paper which provides corresponding solution is listed in *ref* column along with the letter with which the paper is represented in Figure 6. Details of the solutions are provided for six metrics: (S)-scalability, (K)-key connectivity, (R)-resilience, (M)-storage complexity, (P)-processing complexity, and (C)-communication complexity. Numerical values in scalability column are the ranks of the solutions within each section where higher ranks mean higher scalability. Resilience column can take three different classes of values : (i) a number or an equation which represents probability that a link is compromised when an adversary captures a node, (ii) a number or an equation with keyword *nodes* which represents number of sensor nodes whose security credentials are compromised when an adversary captures a node, and (iii) a number or an equation with keyword *secure* which represents number of sensor nodes required to compromise security of whole WSN. Processing complexity is provided in terms of unit functions such as Search, Hash, MAC, PRF, HMAC, VecMul(size), PolyEval(count), etc. Communication complexity includes number and size of messages sent and received where axb, bxc means a number of messages of size c units are sent and b number of messages of size c units are received. Parameters used for each solution are described in detail in Sections 5 and 6. Summary of parameters are: (**d**) degree of a node, (**p**) probability that two nodes are connected due to Erdos and Renyi's work, (**c**) number of cooperative nodes, (**r**) regularity of a connected key distribution graph, (ℓ) number of nodes in a node class, (θ) number of nodes in a generation, (**g**) number of generations, (**j**) number of paths, (ω) number of spaces, (τ) number of spaces assigned to a node, (**m**) number of keys in master key list of a node, (**u**) number of commitment disclosure, (**v**) number of high level commitment disclosure, and (**w**) number of low level commitment disclosure. The papers are: A[Blom 1985], B[Blundo et al. 1992], C[Eschenauer and Gligor 2002], D[Lai et al. 2002], E[Chan et al. 2003], F[Pietro et al. 2003], G[Liu and Ning 2003c], H[Du et al. 2003], I[Liu and Ning 2003b], J[Zhu et al. 2003], K[Du et al. 2004], L[Dutertre et al. 2004], M[Lee and Stinson 2004b], N[Hwang et al. 2004], P[Camtepe and Yener 2004], Q[Lee and Stinson 2004a], R[Huang et al. 2004], S[Hwang and Kim 2004], a[Burmester and Desmedt 1994], b[Perrig et al. 2000], c[Steiner et al. 2000], d[Chen et al. 2000], e[Carman et al. 2002], f[Slijepcevic et al. 2002], g[Perrig et al. 2002], h[Staddon et al. 2002], i[Undercoffer et al. 2002], j[Liu and Ning 2003a; 2003d], k[Deng et al. 2003a; 2003b], l[Law et al. 2003], m[Bohge and Trappe 2003], n[Zhu et al. 2003].

Solution	Ref	(S)	(K)	(R)	(M)	(P)	(C)
Pair-wise key pre-distribution solutions in DWSN (Section 5.1.1)							
All pair-wise	-	1	1	0	2(N-1)	Search	1x1,dx1
Random pair-wise	E	2	Np/(N-1)	0	2Np	Search	1x1,dx1
Closest pair-wise	G	3	c/(N-1)	0	2c+1	Search or 1xPRF	1x1,dx1
IOS	M	3	r/(N-1)	0	r+1	Search or 1xHash	1x1,dx1
Multiple IOS	M	4	rℓ/(N-1)	ℓ nodes	r/ℓ+1	Search or 1xHash	1x1,dx1
Master key based key pre-distribution solutions in DWSN (Section 5.1.2)							
BROSK	D	1	1	1	1	1xPRF	1x1,dx1
Lightweight key management	L	1	1	θ nodes	4+2g	Search or 1xPRF	1x2,dx2
Random key-chain based key pre-distribution solutions in DWSN (Section 5.1.3)							

Continued on Next Page...

Table V – Continued

Solution	Ref	(S)	(K)	(R)	(M)	(P)	(C)
Basic probabilistic	C	2	$\frac{((KP-k)!)^2}{((KP-2k)!KP!)}$	k/KP	2k	Search	1xk,dxk
Cluster key grouping	N	2	$\frac{((KP-k)!)^2}{((KP-2k)!KP!)}$	k/KP	2k	Search	1xC,dxC
Pair-wise key establishment	J	3	$\frac{((KP-k)!)^2}{((KP-2k)!KP!)}$	k/KP	k	Search+1xPRF	1x1,dx1
Q-composite random	E	1	see E	$\binom{k}{q}/\binom{KP}{q}$	2k	Search	1xk,dxk
Multi-path key reinforcement	E	2	$\frac{((KP-k)!)^2}{((KP-2k)!KP!)}$	0	2k	j XOR+Search	1xk+jx1,dxk+jx1
Pair-wise with threshold	J	2	$\frac{((KP-k)!)^2}{((KP-2k)!KP!)}$	0	2k	j XOR+Search	1xk+jx1,dxk+jx1
Co-operative pair-wise	F	2	$\frac{((KP-k)!)^2}{((KP-2k)!KP!)}$	0	2k	c XOR+Search	1xk+cx1,dxk+cx1
Using deployment knowledge	K	2	see K	k/KP	2k	Search	1xk,dxk
Combinatorial design based key pre-distribution solutions in DWSN (Section 5.1.4)							
Combinatorial - Symmetric	P	1	1	1/n	2(n+1)	Search	1xn,dxn
Combinatorial - GQ(n, n^2)	P	2	$1/n^2$	$1/n^3$	2(n+1)	Search	1xn,dxn
Combinatorial - Hybrid	P	3	see P	$1/n^3$	2(n+1)	Search	1xn,dxn
Key matrix based dynamic key generation solutions in DWSN (Section 5.1.5)							
Blom's scheme	A	2	1	λ-secure	2(λ+1)	VecMul(λ+1)	1x(λ+1),dx(λ+1)
Multiple space	H	1	$\frac{((\omega-\tau)!)^2}{((\omega-2\tau)!\omega!)}$	λ-secure	2τ(λ+1)	VecMul(λ+1)	1xτ+1x(λ+1), dxτ+dx(λ+1)
MBS	M	3	r/(N-1)	λ-secure	2(λ+1)	VecMul(λ+1)	1x(λ+1),dx(λ+1)
DMBS	M	4	rℓ/(N-1)	ℓ nodes	(r/ℓ+1)(λ+1)	VecMul(λ+1)	1x(λ+1),dx(λ+1)
Polynomial based dynamic key generation solutions in DWSN (Section 5.1.6)							
Polynomial based	B	3	1	λ-secure	λ+1	PolyEval(1)	1x1,dx1
Polynomial pool	I	2	1	λ-secure	2(λ+1)	PolyEval(1)	1x2,dx2
Location-based pair-wise	G	1	1	λ-secure	5(λ+1)	PolyEval(1)	1x2,dx2
Grid-group deployment	R	2	1	λ-secure	2(λ+1)	PolyEval(1)	1x2,dx2
Group-wise key distribution solution in DWSN (Section 5.2)							
Polynomial - non-interactive	B	1	1	λ-secure	λ+1	PolyEval(1)	1x1,(t-1)x1
Polynomial - interactive	B	1	1	λ-secure	λ+1	1xXOR+PolyEval(t-1)	tx1,(t-1)x1
Pair-wise key distribution solution in HWSN(Section 6.1)							
SNEP	g	1	1	0	1	1xPRF	0,0

Continued on Next Page. . .

Table V – Continued

Solution	Ref	(S)	(K)	(R)	(M)	(P)	(C)
LEAP pair-wise	n	1	1	1	2	1xMAC	2x2,2x1
Group-wise key distribution solution in HWSN(Section 6.2)							
LEAP group-wise	n	1	1	1	1	mxENC	0,mx1
Key matrix based network-wise key distribution solution in HWSN(Section 6.3.1)							
Multitiered	f	1	1	1	m	1xPRF+1XHash	0,0
TESLA based network-wise key distribution solution in HWSN(Section 6.3.2)							
micro-TESLA	g	1	1	0	high	1xMAC+1XHash	0,ux1
TESLA Certificate	m	1	1	0	high	2xMAC	0,3ux1
μ-TESLA extensions	j	2	1	0	low	1xMAC+1XHash	0,vwx1

REFERENCES

Akyildiz, I., Su, W., Sankarasubramaniam, Y., and Cayirci, E. 2002. Wireless sensor networks: a survey. *Computer Networks*.

Blom, R. 1985. An optimal class of symmetric key generation systems. In *Eurocrypt 84*.

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

Bohge, M. and Trappe, W. 2003. An authentication framework for hierarchical ad hoc sensor networks. In *ACM workshop on Wireless Security*.

Boneh, D. and Franklin, M. 2001. Identity-based encryption from the weil pairing. In *CRYPTO 2001*.

Burmester, M. and Desmedt, Y. 1994. A secure and efficient conference key distribution system. In *Eurocrypt 94*.

Camtepe, S. and Yener, B. 2004. Combinatorial design of key distribution mechanisms for wireless sensor networks. In *9th European Symposium on Research Computer Security*.

Carman, D., Matt, B., and Cirincione, G. 2002. Energy-efficient and low-latency key management for sensor networks. In *23rd Army Science Conference*.

Chan, H., Perrig, A., and Song, D. 2003. Random key predistribution schemes for sensor networks. In *IEEE Symposium on Research in Security and Privacy*.

Chen, M., Cui, W., Wen, V., and Woo, A. 2000. Security and deployment issues in a sensor network. Ninja Project: A Scalable Internet Services Architecture, Berkeley.

Deng, J., Han, R., and Mishra, S. 2003a. Enhancing base station security in wireless sensor networks. Tech. Rep. CU-CS-951-03, Department of Computer Science, University of Colorado. April.

Deng, J., Han, R., and Mishra, S. 2003b. A performance evaluation of intrusion-tolerant routing in wireless sensor networks. In *2nd International Workshop on Information Processing in Sensor Networks (IPSN '03)*.

Du, W., Deng, J., Han, Y., Chen, S., and Varshney, P. 2004. A key management scheme for wireless sensor networks using deployment knowledge. In *IEEE Infocom'04*.

Du, W., Deng, J., Han, Y., and Varshney, P. 2003. A pairwise key pre-distribution scheme for wireless sensor networks. In *Proceedings of the 10th ACM conference on Computer and Communications Security CCS'03*.

Dutertre, B., Cheung, S., and Levy, J. 2004. Lightweight key management in wireless sensor networks by leveraging initial trust. Tech. Rep. SRI-SDL-04-02, System Design Laboratory. April.

Eschenauer, L. and Gligor, V. D. 2002. A key-management scheme for distributed sensor networks. In *9th ACM conference on Computer and Communications Security*.

Gaubatz, G., Kaps, J. P., and Sunar, B. 2004. Public key cryptography in sensor networks. In *First European Workshop on Security in Ad-Hoc and Sensor Networks (ESAS 2004)*.

Huang, D., Mehta, M., Medhi, D., and Harn, L. 2004. Location-aware key management scheme for wireless sensor networks. In *2nd ACM workshop on Security of Ad Hoc and Sensor Networks*.

Huang, Q., Cukier, J., Kobayashi, H., Liu, B., and Zhang, J. 2003. Fast authenticated key establishment protocols for self-organizing sensor networks. In *2nd ACM international conference on Wireless Sensor Networks and Applications*.

Hwang, D., Lai, B., and Verbauwhede, I. 2004. Energy-memory-security tradeoffs in distributed sensor networks. In *3rd International Conference on Ad-Hoc Networks and Wireless (ADHOC-NOW 2004)*.

Hwang, J. and Kim, Y. 2004. Revisiting random key pre-distribution for sensor networks. In *ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN 04)*.

Jakobsson, M., Wetzel, S., and Yener, B. 2003. Stealth attacks on ad-hoc wireless networks. In *Vehicular Technology Conference*.

Karlof, C. and Wagner, D. 2003. Secure routing in wireless sensor networks: Attacks and countermeasures. In *First IEEE International Workshop on Sensor Network Protocols and Applications*.

Lai, B., Kim, S., and Verbauwhede, I. 2002. Scalable session key construction protocol for wireless sensor networks. In *IEEE Workshop on Large Scale RealTime and Embedded Systems (LARTES)*.

Law, Y., Corin, R., Etalle, S., and Hartel, P. 2003. A formally verified decentralized key management for wireless sensor networks. In *Personal Wireless Communications*.

Lee, J. and Stinson, D. 2004a. A combinatorial approach to key pre-distributed sensor networks. http:// www. cacr. math. uwaterloo. ca/ ∼dstinson/ pubs.html.

Lee, J. and Stinson, D. 2004b. Deterministic key pre-distribution schemes for distributed sensor networks. http:// www. cacr. math. uwaterloo. ca/ ∼dstinson/ pubs.html.

Liu, D. and Ning, P. 2003a. Efficient distribution of key chain commitments for broadcast authentication in distributed sensor networks. In *10th Annual Network and Distributed System Security Symposium*.

Liu, D. and Ning, P. 2003b. Establishing pairwise keys in distributed sensor networks. In *10th ACM conference on Computer and communications security CCS'03*.

Liu, D. and Ning, P. 2003c. Location-based pairwise key establishment for static sensor networks. In *1st ACM Workshop on Security of Ad Hoc and Sensor Networks*.

Liu, D. and Ning, P. 2003d. Multi-level u-tesla: A broadcast authentication system for distributed sensor networks. Tech. Rep. TR-2003-08, Department of Computer Science, North Carolina State University.

Malan, D., Welsh, M., and Smith, M. 2004. A public-key infrastructure for key distribution in tinyos based on elliptic curve cryptography. In *First IEEE International Conference on Sensor and Ad Hoc Communications and Networks (SECON04)*.

Merkle, R. 1978. Secure communication over insecure channels. In *Communications of the ACM*.

Perrig, A., Canetti, R., Tygar, J., and Song, D. X. 2000. Efficient authentication and signing of multicast streams over lossy channels. In *IEEE Symposium on Security and Privacy*.

Perrig, A., Szewczyk, R., Wen, V., Culler, D., and Tygar, J. 2002. Spins: Security protocols for sensor networks. *Wireless Networks*.

Pietro, R., Mancini, L., and Mei, A. 2003. Random key assignment secure wireless sensor networks. In *1st ACM workshop on Security of Ad Hoc and Sensor Networks*.

Shamir, A. 1984. Identity-based cryptosystems and signature schemes. In *CRYPTO 1984*.

Slijepcevic, S., Potkonjak, M., Tsiatsis, V., Zimbeck, S., and Srivastava, M. 2002. On communication security in wireless ad-hoc sensor network. In *Eleventh IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'02)*.

Staddon, J., Balfanz, D., and Durfee, G. 2002. Efficient tracing of failed nodes in sensor networks. In *1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)*.

Stajano, F. and Anderson, R. 1999. The resurrecting duckling: security issues for ad-hoc wireless networks. In *AT&T software symposium*.

Steiner, M., Tsudik, G., and M.Waidner. 2000. Key agreement in dynamic peer groups. *IEEE Transactions on Parallel and Distributed Systems*.

Undercoffer, J., Avancha, S., Joshi, A., and Pinkston, J. 2002. Security for sensor networks. In *CADIP Research Symposium*.

Wang, G., Cao, G., , and Porta, T. 2004. Movement-assisted sensor deployment. In *INFOCOM 2004*.

Zhou, L. and Haas, Z. 1999. Securing ad hoc networks. *IEEE Network Magazine*.

Zhu, S., Setia, S., and Jajodia, S. 2003. Leap: Efficient security mechanisms for large-scale distributed sensor networks. In *10th ACM Conference on Computer and Communications Security (CCS '03)*.

ZHU, S., XU, S., SETIA, S., AND JAJODIA, S. 2003. Establishing pairwise keys for secure communication in ad hoc networks: a probabilistic approach. In *11th IEEE International Conference on Network Protocols (ICNP'03)*.

ZOU, Y. AND CHAKRABARTY, K. 2003. Sensor deployment and target localization based on virtual forces. In *INFOCOM 2003*.