3RD QUARTER 2008, VOLUME 10, NO. 3

IEEE COMMUNICATIONS SURVEYS

www.comsoc.org/pubs/surveys

SECURING WIRELESS SENSOR NETWORKS: A SURVEY

YUN ZHOU AND YUGUANG FANG, UNIVERSITY OF FLORIDA
YANCHAO ZHANG, NEW JERSEY INSTITUTE OF TECHNOLOGY

ABSTRACT

The significant advances of hardware manufacturing technology and the development of efficient software algorithms make technically and economically feasible a network composed of numerous, small, low-cost sensors using wireless communications, that is, a wireless sensor network. WSNs have attracted intensive interest from both academia and industry due to their wide application in civil and military scenarios. In hostile scenarios, it is very important to protect WSNs from malicious attacks. Due to various resource limitations and the salient features of a wireless sensor network, the security design for such networks is significantly challenging. In this article, we present a comprehensive survey of WSN security issues that were investigated by researchers in recent years and that shed light on future directions for WSN security.

The significant advances of hardware manufacturing technology and efficient software algorithms make a network composed of numerous, small, low-cost sensors, using wireless communication — a wireless sensor network (WSN) [1–3] — a promising network infrastructure for many applications such as environmental monitoring, medical care, and home appliance management. This is particularly true for battlefield surveillance and homeland security scenarios because WSNs are easy to deploy for those applications. However, in many hostile and tactical scenarios and important commercial applications, security mechanisms are required to protect WSNs from malicious attacks. Therefore, the security in WSNs becomes an important and a challenging design task.

NETWORK MODEL

A WSN is a large network of resource-constrained sensor nodes with multiple preset functions, such as sensing and processing, to fulfill different application objectives [1–3].

Usually, sensor nodes are deployed in a designated area by an authority such as the government or a military unit and then, automatically form a network through wireless communications. Sensor nodes are static most of the time, whereas mobile nodes can be deployed according to application requirements. One or several base stations (BSs) are deployed together with the network. A BS can be either static or mobile. Sensor nodes keep monitoring the network area after being deployed. After an event of interest occurs, one of the surrounding sensor nodes can detect it, generate a report, and transmit the report to a BS through multihop wireless links. Collaboration can be carried out if multiple surrounding nodes detect the same event. In this case, one of them generates a final report after collaborating with the other nodes. The BS can process the report and then forward it through either high-quality wireless or wired links to the external world for further processing. The WSN authority can send commands or queries to a BS, which spreads those commands or queries into the network. Hence, a BS acts as a gateway between the WSN and the external world. An example is illustrated in Fig. 1.

Because a WSN consists of a large number of sensor nodes, usually, each sensor node is limited in its resources due to the cost consideration in manufacturing. For example, MICA2 MPR400CB [4], which is the most popular sensor node platform, has only 128 KB of program memory and an 8-bit ATmega128L CPU [5]. Its data rate is 38.4 kbaud in 500 feet, and it is powered by only two AA batteries. The constrained resource cannot support complicated applications. On the other hand, usually, BSs are well designed and have more resources because they are directly attached to the external world.

This work was supported in part by the U.S. National Science Foundation under grant CNS-0716450, CNS-0716302, grant CNS-0626881, and grant ANI-0093241 (CAREER Award).

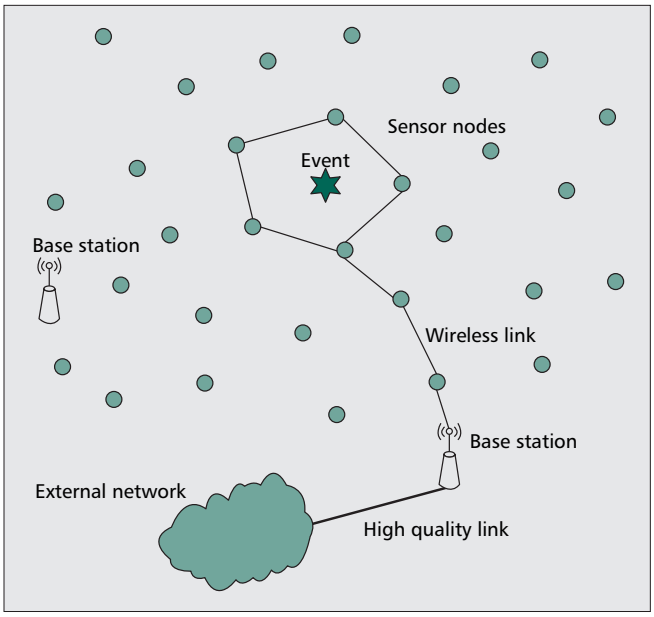

Figure 1. *A wireless sensor network.*

Security Challenges

WSNs have many characteristics that make them very vulnerable to malicious attacks in hostile environments such as a military battlefield:

- A wireless channel is open to everyone. With a radio interface configured at the same frequency band, anyone can monitor or participate in communications. This provides a convenient way for attackers to break into WSNs.
- As in the Internet, most protocols for WSNs do not include potential security considerations at the design stage. Due to standard activity, most protocols are known publicly. Therefore, attackers can easily launch attacks by exploiting security holes in those protocols.
- The constrained resources make it very difficult to implement strong security algorithms on a sensor platform due to the complexity of the algorithms. Most of the time, symmetric key cryptography is the first choice when designing a security protocol for WSNs, although public key cryptography is possible under careful optimization in design and implementation. In addition, a WSN can scale up to thousands of sensor nodes. These pose the demand for simple, flexible, and scalable security protocols. However, to design such security protocols is not an easy task. A stronger security protocol costs more resources on sensor nodes, which can lead to the performance degradation of applications. In most cases, a trade-off must be made between security and performance. However, weak security protocols can be broken easily by attackers.
- A WSN is usually deployed in hostile areas without any fixed infrastructure. It is difficult to perform continuous surveillance after network deployment. Therefore, a WSN may face various attacks.

Attacks

The attacks against WSNs can be classified from different points of view.

- *Attack techniques*: Attackers can disrupt a WSN by utilizing various techniques [6]. Because most communication protocols are known publicly, attackers can *eavesdrop* on the packets transmitted over the air for further cryptanalysis or traffic analysis. The packets that were eavesdropped upon can be *replayed* at a later time or at another place to incur inconsistency. False packets can be *injected* into the network to confuse sensor nodes. Malicious nodes also can *modify* received packets before forwarding them.

Node compromise is one of the most detrimental attacks to a WSN [6]. Because WSNs usually are deployed in a hostile environment without continuous monitoring, an attacker can capture a sensor node and extract all its secrets that are used in security protocols. The exposed secrets render the attacker more capability to launch other severe attacks.

Sometimes attackers are not interested in the data content in the network. They simply may introduce radio *jamming* interference into the same radio bands to disrupt communications between nodes [7]. If an attacker has an infinite power supply, it can jam the wireless channel *continuously* to stop normal communications. Otherwise, the attacker can introduce *intermittent* jamming interference to cause the channel conditions to deteriorate and cause packet loss. If communication protocols are known by the attacker, the intermittent jamming can be more efficient because the attacker knows the part of a packet that is of high value for the jamming attack.

- *Passive versus active*: According to the operation mode, attacks can be *passive* or *active*. In a passive attack, the attacker's goal is to obtain information without being detected. Usually, the attacker remains quiet to eavesdrop on the traffic. If it knows the communication protocols, the attacker can follow those protocols like normal sensor nodes. By passively participating in the network, the attacker collects a large volume of traffic data and carries out analysis on the data such that some secret information can be extracted. Those exposed secrets can be used for various purposes. Usually, the passive attack is very difficult to detect because the attacker does not leave much evidence.

In an active attack, the attacker exploits the security holes in the network protocol stack to launch various attacks such as packet modification, injection, or replaying. The impact of active attacks is more severe than passive attacks. However, additional anomalies can show evidence of malicious attacks because the attacker is actively involved in network communications.

- *External versus internal*: Usually, a WSN is deployed and managed by one authority. All the nodes in the network can be seen as honest and cooperative entities, whereas attackers are precluded from the network and have no right to access the network. Those external attackers can launch attacks only from outside the scope of the network. The impact of attack is limited.

If an attacker can obtain authorization to access the network, it becomes an internal attacker. In this case, the attacker can cause more severe damage because it is seen as a legitimate entity. Usually, an attacker can become an internal one by compromising a legitimate node or by deploying malicious nodes that can pass the network access control mechanism.

Security Requirements

The harsh environments and the existence of threats demand more careful security considerations in the design of WSN protocols. Typically, one or more of the following security services should be provided:

- *Confidentiality* is a basic security service to maintain the secrecy of important data transmitted between sensor nodes. Usually, critical parts of a packet are encrypted before the packet is transmitted from the sending node

and then, the parts are decrypted at the receiving node. Without the corresponding decryption keys, attackers are prevented from accessing the critical information. The kind of information that must be encrypted depends on the applications. In some cases, only the data part of a packet is encrypted, and in the other cases, the packet header also is encrypted to protect node identities.

- *Authenticity* is critical to provide the assurance of the identities of communicating nodes. Every node should check whether a received message comes from a real sender. Without authentication, attackers easily can spoof node identities to spread false information into the WSN. Usually, an attached message authentication code (MAC) can be used to authenticate the origin of a message.
- *Integrity* should be provided to guarantee that the transmitted messages are not modified by attackers. Attackers can introduce interference to some bits of transmitted packets to change their polarities. A malicious routing node can also change important data in packets before forwarding them. Like a cyclic redundancy checksum (CRC) used to detect random errors during packet transmissions, a keyed checksum, such as a MAC, can protect packets against modification.
- *Availability* indicates another important capability of a WSN to provide services whenever they are required. However, attackers can launch attacks to degrade the network performance or even destroy the entire network. A denial of service (DoS) attack [7] is the most detrimental threat to network availability; this occurs when attackers cause the network to lose the capability to provide services by sending radio interference, disrupting network protocols, or depleting the power of nodes through various tricky methods.

The Organization of This Article

The salient characteristics of WSNs and the existence of various malicious attacks pose significant challenges to the design and deployment of security protocols for WSNs. Although some surveys [8–15] describe in general the challenges of and rationales for designing security protocols for WSNs, they either do not explicitly classify or identify the security problems that most researchers are still investigating or they focus mostly on certain specific topics. Moreover, these papers do not cover the latest developments in this area, even though security-critical applications of WSNs have inspired several issues relating to security design for WSNs in the last few years. In this article, we present a survey of security issues and the latest solutions for WSNs, and we identify relevant open problems for future research. Key establishment is the first step to establishing a security infrastructure because all encryption and authentication operations must involve keys. We discuss current key establishment schemes. Authentication, as well as integrity protection, is also discussed. On availability, we have two topics. Routing is critical to provide data delivery services. We discuss the detection of malicious attacks that can corrupt network functionality. Typical applications of WSNs are discussed, and then the article is concluded.

Key Management

Most security protocols are based on cryptographic operations that involve keys. To provide confidentiality, an encryption operation requires a key to be fed into an algorithm so that plaintexts can be transformed into ciphertexts. To guarantee packet authenticity, the source node can attach a MAC to each packet, where usually, the MAC is computed by hashing the concatenation of the packet and a key.

Two types of keys are used in cryptographic systems. The first one is the *symmetric key*, of which the theoretical framework was established by Claude Shannon in his classic paper "Communication Theory of Secrecy Systems" [16]. In a symmetric key system, the sender and receiver share a common key that is kept secret from others. The sender encrypts a plaintext M with the key K by an encryption algorithm E to get a ciphertext $C = E(M, K)$. After receiving the ciphertext C, the receiver inputs C and the key K into a decryption algorithm D to get the original plaintext $M = D(C, K)$.

The other widely-used type is the *asymmetric key*, which was first studied in [17, 18]. In an asymmetric key system, each user has a pair of keys $\{K_s, K_p\}$. The user keeps secret his/her *private key*, K_s, while publishing his/her *public key*, K_p. When a sender wants to send a plaintext M to a receiver, the sender uses the receiver's public key, K_p, to encrypt M to get a ciphertext $C = E(M, K_p)$. Only the receiver can use his/her private key to decrypt the ciphertext and get $M = D(C, K_s)$ because only the receiver knows his/her own private key, K_s. Because a public key is used here, usually asymmetric key systems are called public key systems.

The security of a cryptographic system relies mainly on the secrecy of the key it uses. If an attacker can find the key, the entire system is broken because the attacker can use the key to decrypt the intercepted ciphertexts to find the original plaintexts. The attacker can achieve the goal by cryptanalysis on the eavesdropped packets that are transmitted over the wireless medium. Due to the existence of the redundancy of the message source in the real world, the attacker may know more or less information about the key used. Therefore, the sender and receiver may be required to update the key used between them from time to time. In a WSN, some sensor nodes may be captured by an attacker; thus, the key information is accessible to the attacker and can be used to launch other serious attacks. Therefore, a very important issue is how to securely manage the keys between the sender and the receiver.

Keys also can be classified into two categories according to different communication patterns in WSNs. One is the *unicast key* between a pair of nodes. A pairwise shared key must be established to secure the unicast communication. The other is the *broadcast/multicast key* among a group of nodes. A group key is required to secure the group communication.

In general, to establish keys in a WSN includes two steps. Before sensor nodes are deployed, each node is configured with some key material. After those nodes are deployed into a designated terrain, they perform several rounds of communications to agree on the keys computed with their key materials. Based on the algorithms used to establish pairwise keys, current solutions can be classified into symmetric key schemes and asymmetric key (or public key) schemes. In this section, we discuss pairwise key schemes, including symmetric and asymmetric ones; then, group key management; and last, open problems.

Symmetric Key Management

Most symmetric key algorithms, such as Data Encryption Standard (DES) [19] or Rivest Cipher 5 (RC5) [20], require simple hash, rotation, or scrambling operations, which can be efficiently implemented in hardware or software. On the other hand, asymmetric key algorithms, such as Diffie-Hellman [17] or Rivest Shamir Adleman (RSA) [18], require exponential

operations over a field modulo a large prime number, which are more complex than symmetric key operations. Therefore, the symmetric key technology is more viable on resource constrained low-end devices than the asymmetric key technology. Most of the security protocols in the literature for WSNs are based on symmetric key technology.

A basic problem for applying the symmetric key technology is how to establish a symmetric key between two sensor nodes. A simple approach is to distribute a *global key* [21] to all the sensor nodes. This approach is secure from external attackers that do not know the key but not from internal attackers because the key can be exposed if a node is compromised.

Due to the existence of BSs, centralized key distribution [22] can be used. In particular, each sensor node shares a unique key with a BS, which acts as a key distribution center (KDC). If two nodes must communicate securely, they can acquire a shared key from the BS, which unicasts the key to each of them. This centralized approach could incur a large amount of communication overhead because two neighboring nodes might be required to do handshakes through a central key server at a distant place. In addition, the key server could become a potential point of failure in that the entire network is disabled if the server is corrupted by an attacker.

Most recent solutions to key establishment in WSNs follow a distributed approach, called *key pre-distribution*, where every sensor node is preloaded with key material with which to establish shared keys with other nodes after being deployed into the network terrain. There are two components in this approach: one is how to establish a shared key with key materials, and the other is how to distribute key materials.

In the design of the distributed approach, several problems must be considered:

- *Memory cost*: The memory resource of sensor nodes is scarce. We cannot distribute a large amount of key material into each node.
- *Resilience to node compromise*: Usually, it is impossible to prevent an attacker from compromising some nodes. We can do nothing to rescue those compromised nodes. However, a good scheme should reduce the impact of the node compromise attacks on other normal nodes as much as possible. By compromising a node, an attacker can learn the keys the compromised node uses to communicate with other nodes, but it should not be able to learn keys that the compromised node does not know so that communications between normal nodes are still safe.
- *Local secure connectivity*: Because each node cannot store much key material, it usually can establish shared keys with a subset of its neighboring nodes. Local secure connectivity is the probability that two neighboring nodes establish a shared key directly, that is, the portion of neighbors with whom a node can establish shared keys in one hop. It is directly related to the communication overhead of key establishments. In WSNs, high local secure connectivity is desirable because it means that each node is not required to spend too much energy on the establishment of indirect keys with neighbors through multihop routing, thus saving a lot of communication overhead.

Next, we discuss basic key agreement models and then present various methods to distribute key materials.

Key Agreement Model — To enable a pair of sensor nodes to share a key, the simplest way is to *preload* them with a shared key before they are deployed to a WSN. The pair of nodes can use the shared key directly after deployment. To guarantee that every pair of nodes in a WSN of N nodes has a unique shared key, each node must store $N - 1$ keys, and the overall number of keys in the WSN is then

$$\frac{N(N-1)}{2}.$$

As the size of the WSN increases, this number becomes unacceptably large. Therefore, some scalable methods are desirable for WSNs.

Blom [23] proposed a key agreement method based on $(N, t + 1)$ maximum distance separable (MDS) linear codes, which can be used in the WSN security design. Before a WSN is deployed, a central authority first constructs a $(t + 1) \times N$ public matrix P over a finite field $GF(q) = \{1, 2, \ldots, q - 1\}$, where q is a large prime number. Then, the central authority selects a random $(t + 1) \times (t + 1)$ symmetric matrix S over $GF(q)$, where S is secret and known only to the central authority. An $N \times (t + 1)$ matrix $A = (S \cdot P)^T$ is computed, where $(\cdot)^T$ denotes the transpose operator. Because S is symmetric, it is easy to see:

$$K = A \cdot P = (S \cdot P)^T \cdot P = P^T \cdot S^T \cdot P = P^T \cdot S \cdot P = (A \cdot P)^T = K^T. \quad (1)$$

The node pair (i, j) uses K_{ij}, the element in row i and column j in K, as a shared key. Because K_{ij} is calculated as the product of the i-th row of A and the j-th column of P, the central authority assigns the i-th row of A and the i-th column of P to each node i, for $i = 1, 2, \ldots, N$. Therefore, when nodes i and j must establish a shared key, they first exchange their columns of P and then compute K_{ij} and K_{ji}, respectively, using their private rows of A.

Blom's scheme has a *t-secure* property in the sense that in a network with N nodes, the collusion of less than $t + 1$ nodes cannot reveal any key shared by other pairs of nodes. The memory cost per node in Blom's scheme is $t + 1$. Therefore, Blom's scheme trades the security for the memory cost. To guarantee perfect security in a WSN with N nodes, the $(N - 2)$-secure Blom's scheme should be used, which means the memory cost per node is $N - 1$.

Blundo *et al.* [24] proposed to use a t-degree bivariate symmetric polynomial to achieve key agreement. It is a special case of Blom's scheme in that a Vandermonde matrix is used as the generator matrix of MDS code. A t-degree bivariate polynomial is defined as

$$f(x,y) = \sum_{i=0}^{t} \sum_{j=0}^{t} \alpha_{i,j} x^i y^j \quad (2)$$

over a finite field $GF(q)$, where q is a prime that is large enough to accommodate a cryptographic key. By choosing $\alpha_{ij} = \alpha_{ji}$, we can have $f(x, y) = f(y, x)$. Assume that each sensor node has a unique, integer-valued, non-zero identity. For a pair of sensor nodes u and v, the network administrator assigns a polynomial share $f(u, y)$ to u and another share $f(v, y)$ to v. By assigning polynomial shares, we mean the coefficients of the univariate polynomials $f(u, y)$ and $f(v, y)$ are preloaded into the memory of nodes u and v, respectively. To establish a shared key, both nodes broadcast their IDs. Subsequently, node u can compute $f(u, v)$ by evaluating $f(u, y)$ at y $= v$, and as well, node v can compute $f(v, u)$ by evaluating $f(v, y)$ at $y = u$. Due to the polynomial symmetry, the shared key between nodes u and v has been established as $K_{uv} = f(u, v) = f(v, u)$. Like Blom's scheme, a t-degree bivariate polynomial also is $(t + 1)$-secure, meaning that attackers must compromise no less than $(t + 1)$ nodes holding shares of the same polynomial to reconstruct it.

Random Key Material Distribution — The key agreement models described previously can guarantee that every pair of nodes in a network of N nodes has a unique shared key, but the cost is that each node must store $N - 1$ keys. This is impractical for WSNs due to the memory constraints of sensor nodes and the possible large scale of sensor networks. Instead, most recent research papers in this field relax the security requirement and follow a *partial* pre-distribution approach, where key materials are pre-distributed such that some sensor nodes can establish shared keys directly and then help to establish indirect shared keys between other sensor nodes.

A typical scheme is the random key pre-distribution (RKP) [25], in which each node is pre-loaded with a subset of keys, called a *key ring*, randomly selected from a global pool of keys such that any pair of neighboring nodes can share at least one key with a certain probability. After deployment, two neighboring nodes can have a shared key *directly* or negotiate an *indirect key* through a *secure path*, along which every pair of neighboring nodes has a direct shared key.

The theoretical foundation of RKP is the random graph theory [26]. A random graph $G(n, p)$ is a graph of n nodes for which the probability that a link exists between two nodes is p. The graph does not have any edge if $p = 0$ or is fully connected if $p = 1$. There is a transition from the non-connected graph to the fully-connected graph when p increases. RKP exploits this property by setting p larger than a certain value such that the network is almost connected. Here the size of the global key pool and the size of the key subset for an individual node can be tuned to achieve such a property.

A major concern of RKP is node compromise. The random selection of a key ring for each node means the reuse of each key by multiple nodes. An attacker may compromise a node and expose its key ring, out of which some keys may be used by other non-compromised nodes. This leads to the failures of the links among those non-compromised nodes.

To mitigate the impact of node compromise, the following schemes are proposed: *q-composite RKP* [27] follows RKP except that a pair of neighboring nodes is required to share at least q keys with a certain probability; q-composite RKP can improve the resilience to node compromise when the number of compromised nodes is small. Unfortunately, it is not effective when the number is large. Spatial diversity is used in [28] to improve the resilience to node compromise. Particularly, there are some powerful anchor nodes that are assumed to be tamper proof. A global key is shared by all the anchor nodes and normal nodes, and each normal node is preloaded with a key ring following RKP. Each anchor node uses the global key to broadcast several rounds of random nonces at different power levels in its neighborhood. Each sensor node uses the received nonces to rebuild its key ring. Later, all the nodes can follow RKP to establish shared keys with their neighbors. Finally, each node deletes its original key ring and the global key. The introduced spatial diversity by anchor nodes results in the derived key rings being very different for two nodes that are far away from each other, while making neighboring nodes have more common keys in their derived key rings. Therefore, the impact of node compromise is limited in the local area. However, this scheme assumes that the new nodes can sustain node compromise in its initialization phase, which may not be true in some circumstances. Moreover, the introduction of anchor nodes increases the cost of deploying a WSN.

Another problem of RKP is the lack of authentication because of the reuse of the same key by multiple nodes. To solve the problem, node identity information is used to derive key rings for sensor nodes [29]. A similar approach is taken in the random-pairwise key (RPK) [27] scheme, where each node keeps a set of keys, each of which is uniquely shared with another node. Du *et al.* developed the multiple-space key pre-distribution (MSKP) scheme in [30], where the global key pool in [25] is replaced by a pool of Blom's matrices. Liu and Ning [31] presented the polynomial pool-based key pre-distribution (PPKP) scheme, which is basically the same as the MSKP scheme, but each Blom matrix is replaced by a polynomial. In those schemes, each key is tied to the identities of the nodes sharing it. In this way, the identity of a node can be verified through the normal *challenge-response* approach by other nodes that share the unique key with it. Particularly, a verifier node can send an encrypted random number, called a challenge, to the suspected node, and the suspected node can prove its identity by returning the decrypted result to the verifier node.

RKP requires the storage of a key ring by each node to make the network almost connected. In some cases where sensor nodes do not have enough memory resources, this becomes a problem. Hwang and Kim [32] revisited RKP and its variations and proposed to reduce the amount of key material that each node keeps, which means the probability p of direct key-sharing is smaller than that required in RKP. The smaller p cannot guarantee that the network is almost connected but only assures that the network consists of multiple isolated clusters, of which there is one cluster that is the largest and connects most nodes. In this way, less memory cost still achieves a certain network connectivity, but the trade-off is that some of the small clusters of nodes are isolated because they do not share keys with the largest one.

Deterministic Key Material Distribution — The probabilistic nature of the random distribution of key material cannot guarantee that two neighboring nodes establish a shared key according to the underlying random graph theory. This is not desirable because some sensor nodes may not be able to establish shared keys with their neighbors and thus are isolated. To solve the problem, two deterministic approaches were developed.

One approach is to use a strongly regular graph or a complete graph to replace the random graph to perform key pre-distribution [33–35]. In a (n, r, λ, μ) strongly regular graph, there are n nodes, each of which has a degree of r and any pair of which has λ common neighbors when they are adjacent and μ common neighbors when they are nonadjacent. In the strongly regular graph, every pair of nodes is connected through a path. Each link (edge) can be assigned with a unique key that is preloaded into the two end vertices (nodes). In addition to the regular graph, the block design in the set theory can be used in key pre-distribution, in which all the nodes form a complete graph at the network layer. The tool is the balanced incomplete block design (BIBD). A (v, r, λ)-BIBD is an arrangement into many blocks of v objects, such that each block contains r distinct objects and every pair of objects occurs in exactly λ blocks. For example, when an $(n^2 + n + 1, n + 1, 1)$-BIBD is applied in a WSN, each sensor node is preloaded with $n + 1$ keys that form a block out of a pool of $n^2 + n + 1$ keys, and every pair of nodes has one common key. In [36], the BIBD design is combined with Blundo's model [24] in the sense that each sensor node is preloaded with polynomials. Their scheme enables authentication in addition to those properties provided by the original BIBD design.

The other approach is to use a multi-dimensional grid to replace the random graph [37–42]. Particularly, each sensor node is assigned an ID $(n_1, n_2, \ldots, n_k)$ such that all of the nodes form a k-dimensional grid. Each node is preloaded with some key material such that it can establish shared keys with

other nodes along the same dimension. For example, in the scalable key agreement model developed by Zhou and Fang [37, 38], a t-degree $(k + 1)$-variate symmetric polynomial

$$f(x_1, x_2, \ldots, x_k, x_k + 1) = f(x_{\sigma(1)}, x_{\sigma(2)}, \ldots, x_{\sigma(k)}, x_{\sigma(k+1)}) \quad (3)$$

for any permutation

$$\sigma : [1, k + 1] \rightarrow [1, k + 1] \quad (4)$$

is used to compute the share

$$f_1(x_{k+1}) = f(n_1, n_2, \ldots, n_k, x_{k+1}) = \sum_{i_{k+1}=0}^{t} b_{i_{k+1}} x_{k+1}^{j_{k+1}} \quad (5)$$

for node $(n_1, n_2, \ldots, n_k)$. If two nodes u with ID $(u_1, u_2, \ldots, u_k)$ and v with ID $(v_1, v_2, \ldots, v_k)$ have only one component mismatch in their IDs (similar to the case that the Hamming distance between the two bit-patterns is one), say $u_i \neq v_i$ for some i but $u_j = c_j$ for other $j \neq i$, then nodes u and v can $= v_i = v_j$ compute a shared key as

$$K_{uv} = f(c_1, \ldots, u_i, \ldots, c_k, v_i) = f(c_1, \ldots, v_i, \ldots, c_k, u_i). \quad (6)$$

Two nodes with more than one component mismatch in their IDs can find a path to negotiate an indirect key because all the nodes are organized in the grid.

In other schemes, peer intermediaries for key establishment (PIKE) [39] simply assign a common key for each pair of nodes along each dimension. Hypercube [40] uses bivariate polynomials to achieve key agreement between nodes along each dimension. Delgosha and Fekri [41, 42] use multiple multivariate polynomials to establish multiple common keys between nodes along each dimension.

In those deterministic schemes, a node can find whether it has a direct shared key with another node based on the identity of that node. This can provide an authentication service in that the identity of a node can be challenged based on its keys that are related to its identity.

Location-Based Key Material Distribution — In the aforementioned random and deterministic key material distribution schemes, key materials are uniformly distributed in the entire terrain of a network. The uniform distribution makes the probability rather small that two neighboring nodes share a direct key at one hop, that is, local secure connectivity. Therefore, a lot of communication overhead is inevitable for the establishment of indirect keys over multihop paths. To improve the local secure connectivity, many researchers propose to involve location information in key establishment [43–54].

In the location-based key pre-distribution (LBKP) scheme [43], the entire sensor network is divided into square cells. Each cell is associated with a unique t-degree bivariate polynomial. Each sensor node is pre-loaded with shares of the polynomials of its home cell and four other cells adjoining to its home cell horizontally and vertically. After deployment, any two neighboring nodes can establish a pairwise key according to the polynomial approach [24] if they have shares of the same polynomial. There are other schemes that simply replace the polynomial model with other RKP schemes such as RKP [25] in [44] and MSKP [30] and RPK [27] in [45].

In addition to the square cells used in previous schemes, hexagon [46, 47] and triangle [48] grid models also are investigated to improve spatial diversity. Unlike those cell-based key material distribution methods, Zhou, Zhang, and Fang [47, 48] investigated a cell-pair-based method, in which each pair of neighboring cells is associated with a unique t-degree bivariate polynomial or a matrix. It has been shown in [47, 48] that distributing to each pair of neighboring cells instead of each individual cell can reduce the memory cost while increasing the resilience to node compromise. In addition, Zhou and Fang [49, 50] combined location information with the scalable key agreement model [37, 38] and developed more secure and efficient link-layer key and transport-layer key establishment schemes. Their schemes can improve the security and reduce the memory cost significantly while still maintaining highly secure connectivity.

Another approach is not cell-based. In contrast, it estimates the nodes that are supposed to be deployed close to each other and then preloads them with related key materials. In this way, these neighboring nodes may have shared keys between them after deployment. Liu, Ning, and Du [51] established a group-based key pre-distribution framework that may incorporate previous schemes. They divided the entire network into many *deployment groups*. In each group, a specific keying material distribution scheme can be applied to provide the in-group connectivity. They also picked one node from each group, and all of those picked nodes form a cross group. There is also a specific keying material distribution scheme for each *cross group*. Therefore, two nodes from different deployment groups can establish a shared key through a path in a cross group. A similar approach is taken in [52, 53]. The difference is that all the nodes in one group are preloaded with pairwise keys with each other, and each node can be preloaded with a pairwise key shared with a node in another group. Those preloaded keys can build up a secure path between any two nodes. In [54], a key-position map is used to map a location with a key. If a node is expected to reside in an area according to some probabilistic distribution, it is preloaded with keys corresponding to randomly selected locations around the expected resident point. Therefore, if two nodes are expected to be close to each other, they are more likely to share a common key.

Other Schemes — The aforementioned schemes have strong assumptions. For example, any node can be compromised at any time, and all the nodes have equal capabilities. Under those strong assumptions, security protocols may be complicated. Therefore, some researchers investigated security designs under relaxed assumptions.

Localized Encryption and Authentication Protocol (LEAP) [55] assumes that attackers cannot compromise a node during the network initialization phase. Under this assumption, a global key is used during the network initialization phase to facilitate the key establishment between neighboring nodes, where the pairwise key between two nodes can be derived by inputting the global key and the node identities into a pseudo-random function. To avoid the exposure of the global key due to node compromise, the global key is deleted by each node at the end of the network initialization phase. In [56], Deng *et al.* proposed an opaque transitory master key (OTMK) scheme that is based on LEAP but more secure in the sense that even if the global key is compromised, the pairwise keys between sensor nodes are still safe. The trick there is that the global key is used only to authenticate neighboring nodes but not to establish pairwise keys.

Another strong assumption used by the aforementioned schemes is a requirement that every sensor node can communicate securely with other arbitrary nodes within the network. This is a very strong and unnecessary assumption. In most data sensing and aggregation applications, usually sensor

Key Material Distribution Approach	Schemes	Memory Cost
random	[25, 27–32]	$O(N)$
deterministic	graph-based [33–36]	$O(N)$ or $O(\sqrt{N})$
	grid-based [37–42]	$O\left(k\sqrt[k]{N}\right)$,
location-based	[43–48, 51–54]	$O(N)$ or $O(\sqrt{N})$
	[49, 50]	$O\left(k\sqrt[k]{N}\right)$,

Table 1. *Memory cost.* N *is the total number of nodes in the network.* k *is the number of dimensions.*

nodes are organized into a tree, in which each sensor node communicates with its parent node. This motivates the key establishment between neighboring nodes along the aggregation tree [57]. Before a new node joins a network, it receives two tickets that can be verified by two existing nodes randomly selected by the network administrator. After the new node is deployed into the network, it generates a pairwise key for its parent node. To securely transmit the key to the parent, the new node splits the key into two parts and sends them with its tickets to the nodes selected by the administrator, which authenticates the new node and forwards key materials to the parent of the new node. The merit of a tree-based key distribution is that the memory cost can be reduced significantly.

A weaker attack model [58] assumes that an attacker can only eavesdrop on a small number of communications between sensor nodes during the node deployment phase. Hence, a node wishing to communicate securely with other nodes simply generates a symmetric key and sends it to its neighbors without cryptographic protection, hoping the attacker is unable to hear it. A similar approach is taken in [59], where multi-channel diversity is exploited to facilitate key establishment for WSNs. The assumption is that attackers can monitor only one channel at a time. Under this assumption, every node can randomly select a channel and broadcast key materials in plaintext in the channel. Two neighboring nodes can negotiate a shared key through a channel that is not monitored by attackers.

In [60], node diversity also is studied. A heterogeneous network consisting of powerful nodes and normal nodes is assumed here. RKP [25] is used, and each powerful node can accommodate many more key materials than normal nodes. Given an assumption that powerful nodes are tamper resistant, the scheme [60] reduces the memory cost of normal nodes and increases the resiliency to node compromise compared with other homogeneous cases. This idea is further developed in [61] in the sense that a KDC, if available, is involved in authentication and key establishment between sensor nodes, where powerful nodes act as gateways to the KDC.

In [62], a hierarchical sensor network consisting of several classes is considered. A tree of keys is constructed for the hierarchical network, where the keys at a certain level are distributed to the corresponding class of nodes. The keys at higher levels can be used to derive the keys at lower levels, but not vice versa. The intention of the hierarchical network is to facilitate data collection and fusion and query propagation in hostile environments. However, the distribution of keys into the network requires public key infrastructure (PKI) that could be a concern in terms of cost.

Comparisons of Symmetric Key Schemes

Here we compare random [25, 27–32], deterministic [33–42], and location-based [43–54] schemes.

First, in Table 1, we compare the memory costs of different schemes. Random distribution schemes [25, 27–32] require that each node store a key ring. To maintain a certain level of connectivity, the size of a key ring cannot be small and usually at the level of $O(N)$. Graph-based deterministic distribution schemes [33–36] also require that each node store a key ring. The memory cost of these schemes is either $O(N)$ for regular graph or $O(\sqrt{N})$ for BIBD design. Grid-based deterministic schemes [37–42] have the memory cost only at the level of

$$O\left(k\sqrt[k]{N}\right),$$

where $k > 1$, because they use a k-dimension grid to organize the network. Most location-based schemes [43–48, 51–54] combine location information and random distribution schemes and have less memory cost than random distribution schemes. Their memory cost is $O(N)$ or $O(\sqrt{N})$. One exception is [49, 50], where location information and the deterministic scheme [37, 38] are combined, and thus the memory cost of [49, 50] is

$$O\left(k\sqrt[k]{N}\right).$$

Next we compare the resilience to node compromise in Table 2. Usually, the probability of link compromise can be used to evaluate the resilience to node compromise because the key information in compromised nodes can be used to derive the keys used by the links between non-compromised nodes. For the schemes [25, 27–29, 32–35, 39, 52–54], in which keys are directly pre-distributed, the link compromise probability is approximately linear or quickly increasing with respect to the number of compromised nodes because every time one more node is compromised, more keys from the global key pool are disclosed. However, matrices or polynomial-based schemes [30, 31, 36–38, 40–51] have a useful property of threshold-based resilience, which means the network can tolerate up to a certain number of compromised nodes while still keeping the links between non-compromised nodes safe.

Table 3 shows the local secure connectivity of different schemes. The local secure connectivity of uniform distribution schemes [25, 27–42] is lower than that of location-based

Key Agreement Model	Schemes	Link Compromise Probability
predistributed keys	[25, 27–29, 32–35, 39, 52–54]	approximately linear or quickly increasing to number of compromised nodes
matrices or polynomials	[30, 31, 36–38, 40–51]	threshold-based

Table 2. *Resilience to node compromise.*

Key Material Deployment Pattern	Schemes	Local Secure Connectivity
uniform	[25, 27–42]	low
location-based	[43–54]	high

Table 3. *Local secure connectivity.*

schemes [43–54]. Therefore, by combining location information, each node can establish direct keys with additional neighboring nodes and thereby save energy on the establishment of indirect keys through multihop paths with other neighbors.

Last, in Table 4 we summarize the difference between the aforementioned key pre-distribution schemes [25, 27–54] and other special schemes [55–62]. The major difference is the security assumptions. The most discussed approach in the literature [25, 27–54] assumes strong attackers with powerful capabilities in terms of unlimited time and spatial coverage, but a weak attack model also is studied in [55–62]. The different designs result in different memory cost and connectivity performance, where the trade-off between security and performance can be manifested. Schemes [25, 27–54] require a large memory cost to tolerate node compromise and provide acceptable connectivity, whereas schemes [55–62] have a smaller memory cost and higher connectivity with weak resilience to node compromise.

Asymmetric Key Management

Though it is much more computationally expensive, asymmetric key technology is easier to manage and more resilient to node compromise than symmetric key technology. Each node can keep secret its private key and only publish its public key; therefore, compromised nodes cannot provide clues to the private keys of non-compromised nodes.

Computational Efficiency — Recently, some researchers began to investigate the feasibility of using asymmetric key technology on sensor platforms because of the rapid advances in hardware capability. The most challenging problem here is how to perform asymmetric key algorithms in an efficient way. One approach is to use specific parameters that can speed asymmetric key algorithms without compromising security. For example, Tiny public key (TinyPK) [63] uses RSA-based certificates to authenticate external parties before they can access the network, where the RSA [18] public key is chosen as $e = 3$, such that the signature verification at the sensor side is simplified. Moreover, the Diffie-Hellman algorithm [17] is used in TinyPK [63] to exchange keys between sensor nodes, where the base of exponentiation is chosen as 2, such that the exponential operation is simplified.

Another approach is to use customized hardware design to facilitate asymmetric key operation. Gaubatz, Kaps, and Sunar [64] showed the feasibility of implementing the Rabin scheme [65] and the NtruEncrypt scheme [66] on a customized hardware platform.

Meanwhile, researchers are looking for new algorithms that are more efficient than traditional asymmetric key algorithms. A more promising technology is the elliptic curve cryptography (ECC) [67, 68]. The fundamental operation underlying RSA is the modular exponentiation in integer rings. Its security stems from the difficulty of factorizing large integers. Currently, there exist only sub-exponential algorithms to solve the integer factorization problem of finding p and q given a positive integer $n = pq$, where p and q are large pairwise distinct primes. ECC operates on groups of points over elliptic curves and derives its security from the difficulty of the elliptic curve discrete logarithm problem (ECDLP) of finding an element $x \in GF(q)$, such that $xG = Q$, given a generator G of a finite cyclic point group $\mathbb{G}$ over an elliptic curve $E(GF(q))$ and a point Q in the group. The algorithms best known for solving ECDLP are exponential. Hence, ECC can achieve the same level of security as RSA with smaller key sizes. For example, the 163-bit ECC can provide comparable security to the conventional 1024-bit RSA [69]. Under the same security level, the smaller key sizes of ECC offer advantages of faster computational efficiency, as well as memory, energy, and bandwidth savings; thus ECC is better suited for resource constrained devices.

Computational efficiency is a critical issue if applying asymmetric key technology on a sensor platform. In many cases, high-level programming languages may not be optimized for specific hardware platforms, and therefore assembly languages are required to further reduce the computing time. Gura *et al.* [70] evaluated the assembly language implementations of ECC and RSA on the Atmel ATmega128 processor [5], which is popular for sensor platform such as Crossbow MICA Motes [4]. In their implementations, a 160-bit point multiplication of ECC requires only 0:81 s, whereas 1024-bit RSA public key operation and private key operation require 0.43 s and 10.99 s, respectively.

Applications — In addition to RSA for authentication and Diffie-Hellman for key establishment [63], ECC also is attracting interest for the security design of WSNs due to its efficiency. Huang *et al.* [71] considered a sensor network consisting of secure managers and several sensor nodes. An ECC-based authenticated key establishment protocol is proposed for the key establishment between secure managers and sensor nodes. To reduce the computational overhead of sensor nodes, most computationally expensive asymmetric key operations are put on the secure manager side.

Zhou, Zhang, and Fang [72] designed an access control protocol based on ECC. In particular, the Elliptic Curve Digital Signature Algorithm (ECDSA) [73] is used to authenticate new sensor nodes when they join the network, and the ECC-based Diffie-Hellman algorithm is used to establish shared keys between sensor nodes.

Authenticate Public Keys — Another critical issue of applying asymmetric key technology is the authenticity of public keys. A public key should be owned by the node that claims to

Schemes	Security Assumptions	Memory Cost	Connectivity
[25, 27–54]	strong attackers with powerful capability	large	low for uniform key material deployment and high for location-based key material deployment
[55–62]	weak attackers with limited capability	small	high

Table 4. *Comparisons with other schemes.*

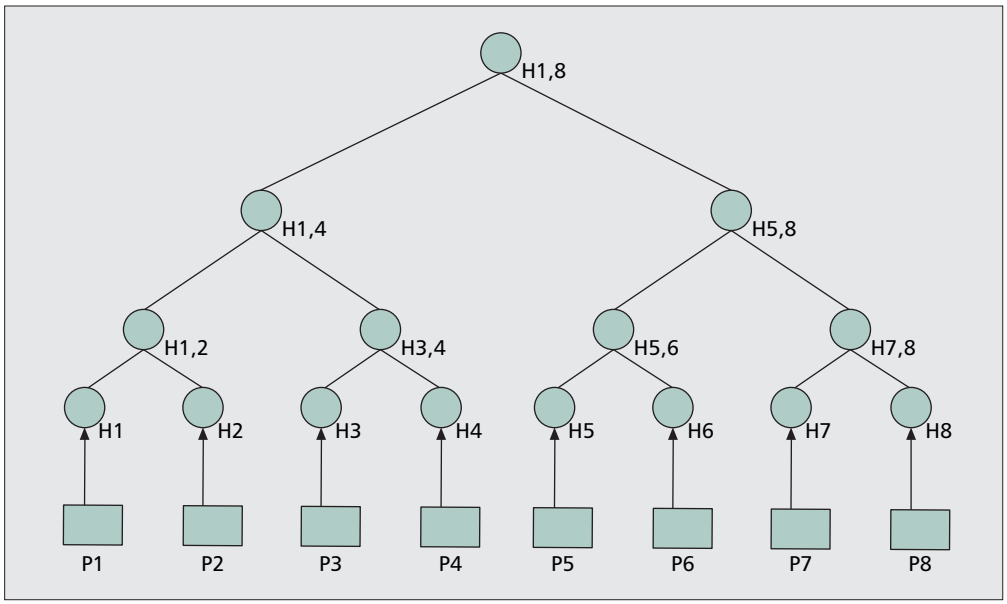

Figure 2. *An example of Merkle tree. Each leaf is a hash of a public key. Each internal node is the hash value on both its left and right children.*

have it. Otherwise, attackers can easily impersonate any node by claiming its public key and launch the *man-in-the-middle* attack. For example, a malicious node C may impersonate node B to node A and also impersonate A to B if A and B cannot verify the public key of each other. In this way, node C can act as an invisible router and learn all the messages between A and B. The conventional solution to public key authentication is to use a certificate signed by a trustful certificate authority (CA). Therefore, node B can send its public key with corresponding certificate to node A such that A can verify the correctness of the certificate with the well-known public key of the CA. Node B can verify the authenticity of A's public key by following the same procedure.

Though technical advances make the use of asymmetric key technology viable in WSNs, asymmetric key algorithms are more expensive than symmetric key algorithms. The authentication of public keys still may incur high energy consumption because authentication is likely to be performed many times. Du, Wang, and Ning [74] developed a public key authentication scheme based on a symmetric key technique, the *Merkle tree* [75]. In the Merkle tree, each parent is a hash of the concatenation of its children, and each leaf corresponds to a node and is calculated as a hash of the node ID and its public key. An example is illustrated in Fig. 2. Each leaf is a hash of a public key. Each internal node is the hash value on both its left and right children, and the root is the accumulator of these packets. The witness of one leaf is the set of the siblings of the nodes along the path from the leaf to the root. For example, the witness of the leaf P_3 is $\{H_4, H_{1,2}, H_{5,8}\}$, and the root can be recovered as $H_{1,8} = H((H_{1,2}, (H(P_3), H_4)), H_{5,8})$. When a sensor node wants to authenticate its public key to other nodes, it attaches the hash of its public key with its witness. Other sensor nodes verify whether they can recover the root and decide the authenticity of the public key.

An alternative approach is to use identity-based cryptography [76], where the publicly known identity information of a node is used as its public key. Identity-based cryptography removes the necessity of public key certificates and thus saves cost on certificate authentication. Based on this technique, Zhang *et al.* [77, 78] proposed the notion of *location-based* keys by binding private keys of individual nodes to their IDs and their locations. In this solution, the pairwise key of two neighboring nodes is the by-product of their mutual authentication process. In addition, any two nodes that are a multihop apart can establish a shared key on demand, based on their location-based keys. Their solution has perfect resilience to node compromise in that no matter how many nodes are compromised, the location-based keys of non-compromised nodes, as well as their pairwise keys, always remain secure.

Group Key Management

According to different scales, there are two types of group communication. One is the network broadcast/multicast. Usually, it is performed by a BS because the network broadcast/multicast requires a large communication overhead, which cannot be supported by sensor nodes. The other is the local broadcast, where each node collaborates with its neighbors to fulfill various purposes, such as routing information exchange or cluster head selection. Both types require a group key to encrypt communications.

LEAP [55] identifies a group key for network broadcast and cluster keys for local broadcast. A group key is a key shared by all the nodes in the network. To tolerate node compromise, the group key is updated occasionally, based on μTESLA [22]. A cluster key is a key shared by a node and all its neighbors, and it is mainly used for securing locally broadcast messages. A node encrypts a cluster key with the pairwise key shared with each neighbor and unicasts it to the neighbor. Because the number of neighbor is usually small, this unicast does not incur too much communication overhead.

A problem with LEAP is that each node learns a group key from the BS individually. If an attacker compromises a node, the group key is exposed. To mitigate this threat, local collaboration is introduced in the group key distribution [79]. In particular, each node must obtain extra secret information from its neighbors, as well as the broadcast information from the BS. Only by combining that secret information and its own secrets, preloaded before deployment, can each node recover the group key. If a node is detected by its neighbors as a malicious one, its neighbors will not collaborate with it. In this way, a malicious node can encounter difficulty in calculating the global key.

Most of the time, a BS takes charge of group key management for the entire network. This may introduce too much management overhead at the BS. To reduce the overhead, a

level key infrastructure for group communications is proposed in [80]. Particularly, all the nodes involved in broadcast/multicast are organized in a tree that is rooted at a BS, and each parent node takes charge of the key update for its immediate children nodes. In this way, the overhead of key management is localized, which is different from centralized group key management schemes such as LEAP [55].

Open Issues

Most current symmetric key schemes for WSNs aim at *link-layer security* for one-hop communications, but not the *transport layer security* for multihop communications, because usually, it is unlikely for each node to store a transport layer key for each of the other nodes in a network due to the huge number of nodes. Asymmetric key technology is expensive but has flexible manageability. Any pair of nodes can establish a shared key using asymmetric key techniques such as Diffie-Hellman [17]. A more promising approach is to combine both techniques such that each node is equipped with an asymmetric key system and relies on it to establish end-to-end symmetric keys with other nodes. To achieve this goal, a critical issue is to develop more efficient asymmetric key algorithms and/or their implementations so that they can be widely used on sensor platforms. How to prove the authenticity of public keys is another important problem. Identity-based cryptography [76] is a shortcut to avoid the problem.

There still is a demand for the development of more efficient symmetric key algorithms because encryption and authentication based on symmetric keys are very frequent in the security operations of sensor nodes.

Key revocation is another unaddressed problem. When a node is detected as a malicious one or as a compromised one, its key must be revoked such that it cannot participate in normal communications. Though some issues are discussed in [81], they mainly target RPK [27]. Because there are so many schemes following different approaches, it is very difficult to design a universal key revocation scheme. It is still an open problem for resource constrained WSNs.

Authentication and Integrity

Usually, authentication and integrity can be fulfilled simultaneously by attaching a MAC to each transmitted packet. In the symmetric key technology, a MAC can be generated by the shared key between the sender and the receiver. In a general approach, the sender concatenates a message M with the shared key K and then computes a MAC $C = H(M||K)$, where $H()$ is a collision-resistant hash function [82]. When the receiver obtains the packet $\{M, C\}$, he/she re-computes a MAC C' with the message M and the shared key K and checks whether $C' = C$. If the equation holds, the message is authenticated and not modified because only the sender knows the shared key. Otherwise, the packet is tampered with.

In the asymmetric key technology, the sender can compute a MAC C for a message M with its private key. When RSA is used, for example, the sender encrypts M to get $C = E(M, K_s)$, where K_s is the sender's private key. This operation is usually called *signing*, and the result is a *signature*. When the receiver obtains the packet $\{M, C\}$, he/she decrypts the signature C with the sender's public K_p and recovers a message M'. If $M' = M$, the message is authenticated and not modified because only the sender knows its private key. Otherwise, the packet is tampered with.

Authentication is very important because the wireless medium leaves a door open to unauthorized attackers who can inject or modify packets. Therefore, every packet must be authenticated before the receiver submits it to higher layer applications.

According to the communication pattern, authentication can be performed in one-hop unicast, multihop unicast, and broadcast. In one-hop unicast, every packet is authenticated between neighboring nodes. Usually, this is performed at the link layer by the link-layer key. It is not efficient for higher layers to authenticate one-hop unicast. Because of the fragmentation of a higher-layer data payload at the link layer, the receiver cannot authenticate each received link-layer packet until receiving a certain number of packets if higher layers perform the authentication. Moreover, the fragmentation also can be attractive to the DoS attack in the sense that attackers inject a large amount of false fragments to deplete the receiver's memory resource.

On the other hand, the link-layer authentication is not secure in multihop unicast because the intermediate nodes may not be trustful, and any one of them can modify the link layer data payload. In this case, higher-layer authentication is required. Usually this can be performed at the transport layer [83] because the transport layer manages the end-to-end connection.

Broadcast authentication also is indispensable. Using symmetric keys in broadcast means the sender and all the receivers must agree on a global shared key. Therefore, efficient group key management is desirable. On the other hand, asymmetric key techniques also can be used, where the sender signs broadcast packets with its private key, and receivers verify them with the sender's public key.

One-Hop Authentication

To support one-hop authentication, a shared link-layer key is required between neighboring nodes. Most symmetric key and asymmetric key management schemes discussed previously can achieve this goal. TinySec [84] is the first fully-implemented link-layer security architecture for WSNs, providing encryption and authentication. It defines two packet types: *TinySec-AE* and *TinySec-Auth*. In TinySec- AE packets, the data payload field is encrypted according to *Skipjack* [85], which is a light-weight block cipher. All the packets of the two types include MACs to provide the packet authentication service. However, TinySec does not discuss how to establish link-layer keys; therefore, TinySec can be combined with key establishment protocols discussed previously to provide a link-layer security solution.

Multihop Authentication

Like the case in one-hop authentication, an end-to-end shared key can support multihop authentication. Most symmetric key establishment schemes discussed previously target the link-layer key establishment. Based on the link-layer secure infrastructure, a multihop key can be negotiated between two end nodes through a multihop path. However, the multihop key negotiation may fail if one of the intermediate nodes along the path is compromised. To deal with this problem, multipath enhancement [86] combining secret sharing [87] can be performed [37, 38].

If an asymmetric key infrastructure is available, the establishment of a multihop key is more secure. Because only the two end nodes can encrypt and decrypt the negotiation messages, the compromise of intermediate nodes does not expose the end-to-end shared key.

Unlike the authentication based on a shared key, a public key certificate also can support multihop authentication. Bohge and Trappe [88] proposed an authentication frame-

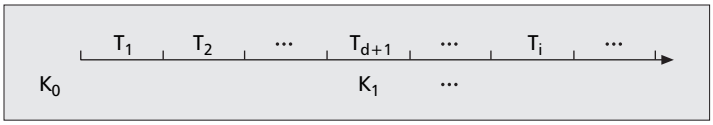

Figure 3. *μTESLA. The broadcasted packet in the* i*-th time slot carries an MAC generated by using the* i*-th key of the one-way hash chain. Every node does not know the* i*-th key when it receives the packet. In the (*i + d*)-th time slots, the base station discloses the* i*-th key.*

work for hierarchical ad hoc sensor networks that consists of access points, forwarding nodes, and mobile sensor nodes. Packets generated by sensor nodes are forwarded by forwarding nodes to access points, where they are routed to specific applications. Preloaded RSA-based initial certificates are used to authenticate sensor nodes and access points to external applications. During the lifetime of the network, applications continue to renew certificates for access points and sensor nodes. Considering the limited capability of access points and sensor nodes, applications authenticate new certificates using the *TESLA* [89] protocol.

The authentication based on the shared key is more efficient than the certificate-based one. A node at one end can verify the identity of a node at the other end through the challenge-response approach based on the shared key. The authentication involves only symmetric key operations such as hash. However, in certificate-based authentication, both end nodes must perform expensive asymmetric key operations.

Broadcast Authentication

Broadcast is a common method to disseminate information in a WSN when a source node intends to spread the same messages, such as commands or queries, to a group of nodes. Each broadcast packet should be authenticated so that attackers cannot inject false information. Currently, most broadcast authentication schemes are based on symmetric key techniques due to their efficiency.

A basic tool in the symmetric-key broadcast authentication is the one-way hash chain (OHC) [90]. An OHC is a sequence of numbers, $K_0, K_1, \ldots K_n$, such that $K_{j-1} = F(K_j)$, $\forall j, j \in \{1, 2, \ldots, n\}$, where the hash function F satisfies two properties:

- Given x, it is easy to compute $y = F(x)$
- Given y, it is computationally infeasible to compute x such that $y = F(x)$

The first number of the OHC K_0 is securely sent to all the receiving nodes. When the source node must broadcast a packet in the k-th round, it includes K_k in the packet. Then every member node can authenticate K_k by verifying $F^k(K_k) = K_0$. However, K_k cannot be used to authenticate packets directly in the k-th round. The reason is that if attackers can capture a packet, they can extract the key attached to it, generate a false packet, and send it to a node in the same round before the true packet arrives at that node.

To solve that problem, in security protocols for sensor networks (SPINS) [22], μTESLA simulates the asymmetry of asymmetric key technology based on OHC to provide authenticated broadcast services. The first key of the OHC is sent to all the receiving nodes as a commitment in advance. The broadcasted packet in the t-th time slot carries a MAC generated by using the t-th key of the OHC. Not every node knows the t-th key when it receives the packet. In the $(t + d)$-th time slot, the BS discloses the t-th key. Thus, every node can authenticate the cached packet by applying the function to the t-th key for t times and checking whether the result is equal to the first key of the OHC. This asymmetry, which is introduced by delayed key release, can efficiently prevent malicious nodes from impersonating the source node. An example of mTESLA is illustrated in Fig. 3.

However, μTESLA requires the distribution of the key chain commitment K_0 to all the nodes, which is not communication-efficient because the commitment must be unicasted to each node. Moreover, a key chain may not support broadcast for a long time because the length of a key chain is fixed. To solve these problems, multilevel key chains are used to extend the lifetime of an authenticated broadcast [91], where each key chain is generated from the keys in its next high-level key chain, and the first commitment key is pre-distributed to all the sensor nodes.

Moreover, the OHC-based approach has other shortcomings, such as the potential threat to the time synchronization procedure [92, 93] and the complicated OHC management. Therefore, μTESLA can be used only by BSs for the network broadcast [22] that targets the entire network. In some circumstances, each individual node requires the local broadcasts to fulfill specific functions in its neighborhood, such as exchanging routing information or cluster head election. Obviously, the local broadcast also must be authenticated.

Zhou and Fang [94] proposed a batch-based broadcast authentication scheme, called BABRA. In BABRA, broadcasted packets are sent in batches, and each batch is a burst sequence of packets. There is a key associated with each batch. All the packets in one batch carry a MAC, whose calculation is based on the associated key. The key is disclosed only after a certain delay from the end of the batch. Therefore, the asymmetry introduced by the delayed key disclosure prevents attackers from injecting bogus packets because the sender never sends packets of one batch after the key disclosure period. To authenticate each batch key, the hash of the key is embedded in the packets of previous batches. Thus, when one packet of the current batch is authenticated, the key hash in it can be used to authenticate the key of the next batch. In BABRA, each key is independent from the others. The elimination of key chains makes BABRA suitable for both the network broadcast by the BS and the local broadcast by sensor nodes.

Though the delayed key disclosure can be an efficient authentication tool, it requires each sensor node to buffer a certain number of packets that may undergo a DoS attack with flooding capability. To solve this problem, asymmetric key-based authentication can be used with a trade-off of increased computational overhead.

Ren *et al.* [95] proposed to use the asymmetric key-based authentication such that packets can be authenticated immediately after reception. To make asymmetric key authentication viable for WSNs, two efficient public key certificate-management methods are proposed in [95]. One is to use the Merkle tree [75] to verify whether a user is authorized to access the network. The other is to use an identity-based signature scheme [96] to authenticate external users. Both methods eliminate public key certificates and therefore can be used in WSNs.

In another scheme [97], Chang *et al.* proposed to use one-time signatures to authenticate broadcast. Each packet is signed by a signature consisting of several one-time private keys that are authenticated by corresponding public keys. To reduce the public key size, they exploit the Merkle tree [75] to associate one public key (root of tree) with several private keys (leaves of tree). Receivers authenticate private keys in each packet with the corresponding public key and then verify the packet signature.

Open Issues

Broadcast/multicast security is still an open problem for WSNs. Though mTESLA can provide authentication, it

induces cache delay for broadcast packets, which attracts DoS attacks. Efficient asymmetric key technology can address the problem in that each broadcast packet carries a signature computed by the source's private key, and receivers can verify the signature by the source's public key. However, development of efficient asymmetric key technology on resource constrained sensor platforms is very difficult. The bottle-neck here is the authentication of the public key. Though several schemes [95, 97] try to avoid the verification of public key certificates by using the Merkle tree [75], the trade-off is more communication overhead because each packet is attached with multiple hash values that may be even longer than public key certificates.

Encryption is another problem for broadcast/multicast. A group key is required to encrypt broadcast messages. However, the management of group keys is very difficult. Whenever a node joins or leaves a group, the current group key should be updated to protect previous or future messages. Usually, group keys can be managed in a centralized or a distributed way [98]. Centralized management requires a central manager in charge of the group key distribution and update; whereas in the distributed approach, all the nodes exchange contributory information in several rounds of communications and agree on a final group key. However, those conventional group key schemes are too expensive on the sensor platform. Centralized management minimizes the computation cost of sensor nodes in that only decryption is required. However, the central manager may become a failure point and a performance bottleneck. Distributed management avoids the problem ,but it requires each node to perform several rounds of computation. The communication overhead of both approaches can be very high when the group membership change is frequent because of the large number of sensor nodes. Though several efficient schemes are proposed in [55, 79, 80], they are prone to malicious attacks such as node compromise. Therefore, efficient and secure group key management is still an open topic.

Secure Routing

The goal of networking is to provide an infrastructure for delivering data from a source node to a destination node. Routing protocols are the most critical component because they address the problem of how to find a path from the source to the destination and finally take charge of data delivery. In this way, the nodes in a network can collaborate with each other to fulfill various applications deployed in advance. If routing protocols fail under malicious attacks, the high layer applications also fail and the network is useless. Therefore, secure routing is very important to guarantee the network functionality in the face of malicious attacks.

Problems

To minimize energy consumption, routing techniques for WSNs employ well-known tactics, as well as those particular to WSNs [99]. For example, in flat routing protocols such as directed diffusion [100], data aggregation and in-network processing are required to reduce the number of transmissions of redundant data. Clustering is a critical process to build up a hierarchical WSN in hierarchical routing protocols such as the low energy adaptive clustering hierarchy (LEACH) [101]. Sensor nodes in the local area cooperate to select a cluster head that may be more powerful so that it can perform more complex operations such as data aggregation or long distance routing. In location-based routing protocols such as Geographical and Energy-Aware Routing (GEAR) [102], the location of a sensor node, which can be estimated by global positioning system (GPS) devices or GPS-free methods, is used as the routing metric. However, most routing protocols [99] for WSNs do not consider security in their designs. Karlof and Wagner [6] pointed out that most routing protocols for WSNs are vulnerable to malicious attacks. Unencrypted packets that carry routing information can be easily subject to eavesdroppers so that attackers can discover the network topology. Attackers can inject false routing information to launch a Sybil attack [103, 104] or redirect packets to change network topology [105, 106]. Both of the attacks can change the network traffic pattern so that some malicious nodes can receive most of the traffic before it arrives at the BS. In data aggregation, the aggregation node can be compromised so that the aggregated data is tampered with. A non-aggregation node also can report false data to the aggregation node to disrupt the final report. A location deterministic operation is vulnerable because it requires cooperation among several nodes and may not be successful if some of them are malicious. A malicious node intentionally may drop some of the passing traffic. Of course, it can drop all the packets to act like a black hole, but this is easy to detect. Selective forwarding is more difficult to detect. A malicious node may drop the packets from some selected nodes and forward those from other nodes. A more subtle way is to drop packets intermittently so that it behaves like an unstable channel. Most routing protocols require that each sensor node periodically broadcast routing information to maintain the network topology. If the time synchronization in the maintenance operation is attacked [92, 93], the whole network fails. Though several proposals [107–112] tried to secure ad hoc routing protocols, they hardly can be applied in WSNs for three reasons. First, those proposals all target ad hoc networks, which are different from WSNs in terms of resources and communication patterns [6]. Second, those proposals are security extensions of existing ad hoc routing protocols such as dynamic source routing (DSR), ad hoc on-demand distance vector (AODV), or destination-sequenced distance vector (DSDV) [113], which are not suitable for WSNs. Third, those proposals require either asymmetric key cryptography or complicated symmetric key cryptography, which are expensive on sensor platforms.

Available Solutions

Several possible security countermeasures for securing routing protocols are discussed in [6]. Link-layer encryption and authentication by using a global key can protect WSNs against external attackers because they do not know the global key. However, this does not secure against node compromise because the global key can be exposed. A trustful BS can detect spoofed node identities if every node shares a unique key with it, which is studied in SPINS [22]. However, the centralized control can introduce too much communication or management overhead. To counter selective forwarding by malicious nodes, multipath routing can be used to increase the probability of data delivery. To support topology maintenance, authentication is required to protect broadcast of routing information in a local area. Though these methods effectively can prevent external attackers from spoofing, modifying, and replaying information and reduce the impact of selective forwarding, they cannot protect the network from internal malicious nodes efficiently.

In the INtrusion-tolerant routing protocol for wireless SEnsor NetworkS (INSENS) [114], the authenticated routing information can be collected by the BS so that it can calculate the routing table for every sensor node. The broadcast infor-

Security Threats	Countermeasures
eavesdropping	link layer encryption [6, 22, 114–116]
injection, modification	link layer authentication [6, 22, 114–116]
selective forwarding	multi-path routing [6, 114]
node impersonation	authentication through BS [22]
false routing information	authenticated broadcast [6], topology verification by BS [114]
DoS	prohibit network broadcast from sensor nodes [114]

■ Table 5. *Secure routing protocol.*

mation from the BS is authenticated by a one-way hash chain. To prevent DoS attacks, individual nodes are not allowed to broadcast information to the entire network. To increase the tolerance to node compromise, redundant multipath routing is used so that traffic can survive even if some paths are compromised. However, INSENS assumes an application scenario where communications can happen only between sensor nodes and the BS. It does not support in-network processing.

Pietro *et al.* proposed an extension of logical key hierarchy for WSN (LKHW) [115] to protect the directed diffusion protocol [100]. An LKH is a key tree structure with source nodes as leafs and a sink node as the root. Each leaf node holds keys along the path from it to the root node. In LKHW, an LKH is established before data are fused. Then the LKH is used to provide encryption and authentication for data fusion.

A Secure Routing Protocol for Sensor Networks (SRPSN) is developed in [116]. A hierarchical network is constructed with cluster heads and cluster member nodes. Messages from sensor nodes are routed by cluster heads. To protect data, a preloaded symmetric key is shared between all cluster heads and the base BS.

Several security threats to WSN routing protocols and countermeasures are summarized in Table 5.

Open Issues

For routing protocols, security considerations should be considered at the design stage. Considering specific application scenarios, the network administrator should analyze the possible black holes that may corrupt or disrupt the applications and deploy security countermeasure in advance.

A general approach to protect routing protocols is to authenticate the routing information exchanged between nodes. This can effectively prevent an external attacker from injecting false routing information. However, authenticated routing metrics may not be correct in that an internal malicious node can claim false routing metrics without being detected because it has correct keys. Therefore, it is necessary for high layers to verify routing metrics. Some metrics such as residual energy are very difficult to verify because each node cannot know the energy consumption of other nodes. The metrics that have local similarity are easy to verify. For example, most geographical routing protocols have inherited immunity to false routing information because the nodes that are close to each other should have similar geographical distances to the sink.

Considering the frequent node failure under malicious attacks, multipath routing is a promising approach to provide robust and secure data transmission services. Based on the secret sharing technique [87], a message can be decomposed into many shares, and those shares can be spread into multiple paths and collected by the sink to recover the original message [86]. How many shares each path can be assigned depends on the security or the reliability of that path. In this way, the confidentiality of data can be strengthened because attackers must compromise a certain number of routes to capture a message.

However, the integration of security measures into routing protocols can introduce additional overhead. Strong security primitives can ensure a high level of security but may not be acceptable because of the resource constraints of WSNs. How to achieve a trade-off between security and routing overhead is still an open problem.

Intrusion Detection and Countermeasures

In many cases, no matter how carefully we design a security infrastructure for a WSN, attackers can still find a way to break into it and launch attacks from inside the network. If they keep quiet to eavesdrop on traffic, they can remain undetected. If they behave actively to disrupt network communications, anomalies will be detected, indicating the existence of malicious attacks. Intrusion detection mechanisms can detect those attacks based on the anomalies. In this section, we discuss several typical attacks and the corresponding detection methods.

Node Compromise

Usually, a WSN is managed by an authority that can deploy a secure infrastructure to protect the network. At first, all the secrets deployed are unavailable to attackers. An attacker, without knowing any secrets, can eavesdrop on packets but may not be able to discover the content of the packet because most likely, the packet payload is encrypted. Thus, the attacker is an external attacker with limited attack capabilities.

However, a WSN usually is deployed in a hostile environment. An attacker may compromise a sensor node to extract all its keying material. Even if tamper-resistant devices are available for a sensor platform, they still cannot guarantee the perfect security of secrets [117]. Hence, node compromise usually is unavoidable in WSNs. By compromising one node, an external attacker can become an internal attacker and launch more severe attacks. The attacker can use the compromised node to monitor the network traffic. It is very hard to detect this attack because the attacker follows the normal network protocols without showing an anomaly. The attacker can also use the malicious node to launch various active attacks. This situation poses the demand for *compromise-tolerant* security design. The network should remain highly secure even when a certain number of nodes are compromised. In particular, the impact of node compromise should be limited to the local area where the compromised node resides.

In location-based key distribution schemes [43–49, 51–54], nodes close to each other have more correlated key materials than nodes that are far away from each other. Attackers can use the key materials in compromised nodes to derive keys shared by other non-compromised nodes in the local region but not in distant regions. Therefore, using location information can mitigate the impact of node compromise.

To further control the impact of node compromise, public key techniques can be used, because the symmetric key techniques used in [43–49, 51–54] cannot totally solve the node compromise problem. Zhang *et al.* [78] proposed a suite of location-based security mechanisms in which each node holds

a private key bound to both its ID and its geographic location. Based on location-based keys, they developed a neighborhood authentication protocol that can successfully localize the impact of compromised nodes to their vicinity. In addition, they demonstrated how location-based keys can act as efficient countermeasures against many notorious attacks against WSNs, such as a Sybil attack [103], a node replication attack [118], or a Wormhole attack [105].

Active Attacks

An attacker can exploit compromised nodes to launch many active attacks to disrupt normal operations in a WSN. Usually, these attacks are hard to prevent due to the possibility of node compromise. Therefore, some detection methods must be performed to counteract these attacks.

Selective Forwarding — Forwarding packets is a major responsibility of a routing node. However, a malicious node intentionally may drop any packet and forward other ones. Wang *et al.* [119] proposed a failure detection framework to detect the selective forwarding attack. The observation is that for a routing node, the number of packets it must forward must be equal to the number of packets it receives. In their framework, each sensor node can work under a promiscuous mode so that it can overhear the transmission of neighboring nodes. If a neighbor of a suspected node finds that the number of packets that the suspected node fails to forward exceeds a certain threshold, the neighbor can collaborate with other neighbors of the suspected node, and the opinions from the neighbors of the suspected node are collected to form a decision about the suspected node.

The Sybil Attack — The *Sybil attack* was first studied in the context of peer-to-peer networks [104]. Then it was found to be a serious threat to WSNs [103]. In the Sybil attack [103], a malicious node illegitimately takes on multiple identities. It has been shown that the Sybil attack may pose a serious threat to distributed storage and routing protocols [6]. In addition, it also can cause devastating consequences to other applications such as data aggregation, voting, fair resource allocation, and misbehavior detection [103]. To detect the Sybil attack, two methods were discussed in [103]. One method is radio resource testing in which each node assigns a unique channel to each of its neighbors, including fake neighbors, and tests whether its neighbors can communicate with it through the assigned channels. Because the radio of a sensor platform is usually incapable of simultaneously sending or receiving on more than one channel, the failure of communication through one channel may be a sign of the Sybil attack. The other method is to use the ID-based symmetric keys. For example, each sensor node is preloaded with a set of keys that are selected from a global key pool by its node ID. The ID of a suspected node is challenged by a set of validating nodes based on the keys shared between the suspected node and the validating nodes. Several other methods were suggested in [103], including registration, position verification, and code attestation. Moreover, ID-based public keys [78] also can defeat the Sybil attack because both the ID and location information were taken into the generation of key material during the initialization phase, hence multiple identities need multiple keys, and this is impossible for a malicious node to achieve.

The Node Replication Attack — In the *node replication attack* [118], an attacker intentionally puts replicas of a compromised node in many places in the network to incur inconsistency. Like the Sybil attack, the node replication attack also can enable attackers to subvert data aggregation, misbehavior detection, and voting protocols by injecting false data or suppressing legitimate data [118]. Conventional methods to detect a node replication attack usually include centralized computing based on node locations or the number of simultaneous connections, which is vulnerable to the single-point failure. Distributed detection of the node replication attack was proposed in [118], where each node is assumed to know its location, and it is required to send its location to a set of witness nodes. If a witness node finds a contradiction in the location claims of a suspected node identity, this suspected node identity must be replicated many times. Asymmetric key technology is used here to guarantee the authenticity of location claims. A similar approach is discussed in [78]: each node has a private key corresponding to its location, and the location-based key can be used to detect node replicas.

The Wormhole Attack — In the *Wormhole attack* [105], an attacker can tunnel packets through a secret, low-latency broadband channel between two distant places and replay them. This attack can distort the network topology by making two distant nodes believe they are neighbors, thus it becomes a serious attack on routing protocols [105]. To detect the Wormhole attack, Hu *et al.* proposed to use *packet leashes* [105], where location or timing information is embedded in packets, to limit the maximum range over which packets can be tunneled. They require that each node either knows its location or has a tightly synchronized clock so that this information can be used to calculate the maximum distance that a relayed packet could travel. *Directional antennas* [120] also were used to defend against the Wormhole attack, where some direction information is used to detect the replayed packets. However, these defenses target ad hoc networks and require expensive hardware devices, which may be infeasible for most resource-constrained sensor networks. Wang and Bhargava [121] proposed to use centralized computing to detect the Wormhole attack in sensor networks, in which a controller collects the location information for all nodes to reconstruct the network topology such that any topological distortion can be visualized. However, the visualization approach incurs too much communication overhead, especially when malicious nodes move around in the entire network because each location change of the Wormhole triggers a new round of execution of the topology reconstruction algorithm. Location-based keys [78] also can effectively address the Wormhole attack because each packet is authenticated by the location-based key.

The Rushing Attack — Most on-demand routing protocols rely on broadcast ROUTE-REQUESTs to find routes. In a *rushing attack*, an attacker can forward ROUTE-REQUESTs more quickly than legitimate nodes so that it is more possible that the chosen route includes the adversary. The widely used duplicate suppression technique makes the rushing attack possible. To counteract the attack, Hu, Perrig, and Johnson proposed the Route Access Protocol (RAP) [106], in which cached ROUTE-REQUESTs and the node lists embedded in those ROUTE- REQUESTs can be used to check the rushing attack.

Attacks Summary — The active attacks and their countermeasures are summarized in Table 6.

Node Monitoring

Through active attacks, an attacker displays many anomalies, which are the indications of a malicious attack. Intrusion

Attacks	Countermeasures
selective forwarding	local monitoring [119]
Sybil	radio resource testing or ID-based symmetric key [103], location-based key [78]
node replication	location verification by witness nodes [118], location-based key [78]
Wormhole	packet leashes [105], directional antennas [120], topology checking by central server [121]
rushing	path records by embedding node list in ROUTE-REQUEST [106]

■ Table 6. *Countermeasures to active attacks.*

detection mechanisms attempt to detect an attack based on those anomalies. Usually, the neighbors of a malicious node are the first entities to learn of the abnormal behaviors. Hence, it is convenient to let each node monitor its neighbors such that intrusion detection mechanisms can be triggered as soon as possible.

Khalil, Bagchi, and Nita-Rotaru proposed the detection, diagnosis, and isolation of control attacks in sensor networks (DICAS) protocol [122] to detect, diagnose, and isolate malicious nodes. Local monitoring capability is utilized in that a neighbor of both the sender and receiver can oversee the communication behaviors of the receiver. If the receiver has any abnormal behavior on the received packets, it can be detected. If the number of abnormal behaviors is larger than a threshold, the neighbors of the detected malicious node refuse to receive packets from and send packets to it so that the malicious node is isolated from the network.

Cumulative observations of anomalies can be used to evaluate the integrity of sensor nodes. Ganeriwal and Srivastava [123] developed a reputation-based framework, in which each node holds reputations for other nodes. Based on the observations of whether other nodes are cooperative or not, those reputations are updated through an iterative procedure and used as criteria to decide whether a node is malicious or not.

Continuous monitoring can be energy consuming. Therefore, a cluster-based detection approach is presented in [124]. The network is divided into clusters. Each cluster head monitors its cluster members. All the members in a cluster are further divided into groups, and the groups take turns monitoring the cluster head. Not all the sensor nodes continue to monitor, thus reducing the overall network energy cost.

In addition to monitoring neighbors, Seshadri *et al.* proposed a physical layer intrusion detection method called secure code update by attestation (SCUBA) in [125]. SCUBA implements a self-monitoring mechanism by a primitive function called indisputable code execution (ICE). ICE can create a correct execution environment for programs. An attacker can be detected by ICE if it tries to fake the execution environment.

Denial of Service

All the components of a WSN, including hardware architecture and software protocols, are integrated and cooperate to perform various tasks. If any of them malfunctions, the capability of the entire network may decrease. Hence, attackers may launch attacks to diminish the capability of the network to provide expected services, leading to a DoS attack [7]. In fact, DoS is the goal of the active attacks, in which attackers participate in the network operation, like packet dropping, but not the passive attacks, in which attackers stand outside of the network, like eavesdropping.

A typical technique to initiate a DoS attack is radio jamming [126]. Constant or random interferences can be introduced by attackers to corrupt normal communications. It was shown in [126] that using signal strength, carrier sensing time, or the packet delivery ratio, individually, is not enough to detect jamming. Instead, consistency checking based on signal strength or location information can detect a jamming attack. The jamming techniques studied in [126] are either inefficient (constant, deceptive, and reactive jamming) or ineffective (random jamming). By exploiting the link-layer semantics, a more efficient and effective jamming attack can be launched [127].

A trickier way to launch a DoS attack is to employ protocol defects. Aad, Hubaux, and Knightly [128] studied a particular DoS attack, namely, *Jelly-Fish*, in which relay nodes stealthily reorder, delay, or periodically drop packets that are expected to be forwarded, in a way that leads end-to-end congestion control protocols astray. This attack is protocol compliant but has a devastating impact on the throughput of closed-loop flows, such as TCP. However, a surprising observation is that the attack actually can increase the capacity of networks because they starve all multihop flows and provide all resources to one-hop flows that cannot be intercepted by JellyFish.

In [129], McCune *et al.* study the DoS attack in a broadcast, where sensor nodes are deprived of broadcast messages by an attacker. They propose secure implicit sampling (SIS), an algorithm by which a broadcasting BS detects the failure of nodes to receive its broadcast. The goal of SIS is to increase the probability that the attacker is detected when it increases disruption of broadcasts. SIS achieves this goal by having a subset of recipients acknowledge each broadcast, where this subset is computed deterministically but in a way that is hidden from the attacker.

Compared to DoS detection, it is more difficult to defend against a DoS attack. Attackers may have the capability of launching many attacks that result in DoS, and counteracting each possible attack is not an easy job. Usually, what we can do is to define the region under the DoS attack and avoid the region by reestablishing new routes. Also, careful consideration of DoS resistance should be addressed at the network design phase to reduce the loopholes that may be utilized by adversaries.

Deng *et al.* [130] studied a specific DoS attack called permanent DoS (PDoS), in which an attacker overwhelms sensor nodes a long distance away by flooding a multihop end-to-end communication path with either replayed packets or injected spurious packets. They proposed a solution using OHCs to protect end-to-end communications against PDoS. Each node along a path is configured with an OHC, enabling each intermediate node to detect a PDoS attack and prevent the propagation of spurious or replayed packets. In this mechanism, every packet sent by an end node includes a new OHC number. An intermediate node forwards a packet only if the included OHC number is verified to be new. This OHC-based solution is more resilient to compromise than the naive approach of sharing a single path key because an attacker given the current and earlier OHCs cannot generate a legitimate next OHC number and therefore, cannot flood the path with bogus packets or replayed packets.

Secure Base Stations

A BS is the gateway of a WSN to the external wired world. A WSN must exchange all information with the external world

through a BS. Usually, a BS has more powerful capabilities to perform centralized computation and is more resilient to malicious attacks. Hence, the conventional security schemes for WSNs assume that the BS is always secure. However, there is still a possibility that the BS may become a failure point if attackers are powerful enough to break it. Hence, BS security is also important and requires more consideration.

Deng, Han, and Mishra [131] proposed several methods to protect the BS from malicious attacks. The first method is to deploy multiple BSs to provide tolerance against individual BS failure. The second method tries to hide the identity of the BS. Particularly, a pairwise shared key is used to encrypt packets, including the address field in the packet headers, between two neighboring nodes. Instead of using node IDs in the address field, they use a hash function to construct several anonyms for each node. All the nodes use their anonyms as either source addresses or destination addresses. The pairwise shared keys are generated and distributed by the BS during the topology construction phase. In the third method, the BS is allowed to relocate so that its location is hard to track by attackers.

If an attacker wants to attack the BS, it must know where to find it. Through traffic analysis, the attacker can achieve the goal of finding the location of the BS. Therefore, it is very important to disguise the traffic pattern. Three methods are proposed in [132] to counteract traffic analysis attacks. First, a source node randomly can select a multihop path for each flow. Second, fake paths can be created randomly to cheat attackers. Finally, multiple random areas of high communication activity can be created to hide the real location of the BS.

Sometimes the BS is far away from the area of interest. To reduce the overhead of long path data transmission, the BS can send out mobile sinks for efficient data collection, data querying, localized sensor reprogramming, identifying and revoking compromised sensors, and other network maintenance. Employing mobile sinks, however, raises a new security challenge: if a mobile sink is given too many privileges, it will become very attractive to attack and compromise. Using a compromised mobile sink, an attacker easily can bring down or even take over the sensor network. Thus, security mechanisms that can tolerate mobile sink compromises are essential. In [133], Zhang *et al.* proposed to restrict the privilege of a mobile sink without impeding its capability of carrying out authorized operations for an assigned task. Basically, each mobile sink carries an authenticator for each task it may enforce. Each sensor node can verify the task authenticator from the mobile sink before performing the desired task. When a mobile task is compromised, the BS uses authenticated broadcast to revoke the ID of the mobile sink.

Open Issues

Although many intrusion detection techniques were proposed to detect malicious attacks, most of them target only one specific attack by using different approaches and hardware assumptions. It is very difficult to integrate those techniques into a uniform hardware platform due to cost and implementation considerations. A promising approach is to choose a set of criteria based on the characteristic analysis of different attacks and establish a simple, uniform intrusion detection framework. Reputation-based detection is good in that it is based on the statistics of anomalies but not specific detection techniques. However, it cannot recognize the kind of attack that is occurring. This is because the criteria for determining malicious behavior has not been addressed. It is beneficial to investigate how to detect a specific attack under the general framework.

Identifying an intrusion is a requirement for further active network protection measures. The network administrator utilizes the alert messages from the intrusion detection system to analyze the cause and impact of an attack and enforce some countermeasures. However, the whole defense process is in fact becoming more and more complex in the face of the large amount of emerging network techniques, protocols, applications, and attacks; and thus, the human involvement in the defense process may be inefficient and error-prone. To ease the management and protection of networks, which has been envisioned by autonomic computing [134–136], self-protection and self-healing are inevitable in future WSNs. In particular, realizing autonomic computing in a WSN requires two tasks. One is that the network is capable of defending itself against malicious attacks and recovering by self-healing measures. The other is that the network must be able to anticipate potential security problems based on historic log reports and take the required steps to avoid or mitigate them, for example, by automatically upgrading its defense systems.

Breaking the physical layer is the first step for adversaries to launch attacks. Most recent security solutions target the protocols of high layers, but not the physical layer. However, it would be beneficial if the physical layer can provide resilience to malicious attacks. Spread spectrum can reduce the impact of radio jamming by hiding a signal behind the environmental noise. Improved tamper-resistant techniques can reduce the probability of node compromise such that the possibility of internal attacks also can be reduced. Meanwhile, the changes in physical characteristics of sensor nodes, such as code length, may be used to detect malicious tampering.

Secure Applications

Under malicious attacks, WSN applications may not be performed successfully. In this section, we discuss some security issues related to the applications.

False Reports

When an event happens, the surrounding sensor nodes have similar observations of the event. If all the nodes send their notifications to the BS, it will result in the waste of too much energy. Therefore, the *in-network processing* becomes a major component of data fusion in WSNs, where the surrounding nodes cooperate with each other such that a final report can be generated by a data fusion node and sent by the fusion node to the BS. However, if some nodes are compromised and then send out false reports, this type of data fusion procedure will fail.

Report Verification — To deal with the false report problem, some witness nodes can be used [137] to provide data assurance. A report should be signed by m witness nodes, who have similar observations, before being forwarded by the fusion node, which also signs the report, to the BS. The BS performs verification based on the received $m + 1$ signatures to decide whether the report is valid or not. Przydatek, Song, and Perrig proposed a similar scheme called secure information aggregation (SIA) [138], in which a report, generated according to carefully designed aggregation algorithms, carries a commitment generated by the fusion node, and the BS verifies the report through interactive protocols with the fusion node.

En-Route Filtering — If false reports always are forwarded to the BS, a malicious node can keep sending false reports to deplete the resources of other forwarding nodes and the BS,

thus leading to the DoS attack. To counteract the attack, *en-route filtering* is proposed in [78, 139, 140], where the signatures carried by a report are verified by the nodes along the route from the data fusion node to the BS so that false reports can be filtered out before they arrive at the BS. This method requires that the nodes generating signatures have shared keys with those nodes on the route, where the shared keys can be established according to the key establishment schemes discussed earlier. To protect the shared keys from node compromise, Zhang and Cao [141] developed a family of pre-distribution and local collaboration-based group rekeying (PCGR) schemes based on the secret sharing technique, in which a key of a node can be secured and updated by its neighbors.

The security of the keys in [139–141] is threshold-based in that each key cannot be exposed until a certain number of nodes that have related key material are compromised. However, a graceful degradation of security is desirable in some cases. Hence, a location-based resilient security (LBRS) solution is proposed in [142]. The entire terrain is divided into a regular geographic grid, and each cell is associated with multiple keys. Based on its location, a node stores one key for each of its local neighboring cells and a few randomly chosen remote cells. Each detected event must be endorsed through multiple keys bound to the specific location of the event. When it is forwarded to the sink, a report must carry several MACs whose calculation is based on the keys of its original cell, and be verified by the nodes along the route to the sink. Because each event must be signed with keys bound to its location, attackers cannot abuse the keys compromised at one location to spoof reports related to other locations. Therefore, LBRS can achieve a graceful degradation of security. A similar approach also is described in [143], which differs in the selection of verifying cells.

Unlike the aforementioned schemes [78, 139–143], in [144], Yu and Guan proposed a verify-before-send method to filter out false reports. The sensing nodes attach MACs to an aggregated report with their own authentication keys. Those authentication keys are disclosed to forwarding nodes after they send out the report. Each forwarding node can verify the report with authentication keys received from its upstream node only after sending out the report to its downstream node. In this way, each forwarding node can verify but not modify the report. If the report is invalid, the forwarding node informs its downstream node to drop the report. It is shown in [144] that this approach can achieve a better filtering result on false reports.

Hop-by-Hop Aggregation — All the schemes [78, 139–144] assume that the network is flat. Data fusion is performed only between nodes around the detected event. However, most data aggregation protocols [100, 145–150] organize sensor nodes into a tree hierarchy rooted at the BS, thereby eliminating the data redundancy in the sensor data of the network, and hence reducing the communication cost and energy expenditure in data collection. The non-leaf nodes act as aggregators, fusing the data collected from their child nodes before forwarding the results toward the BS. In this way, data are processed and fused at each hop on the way to the BS, and communication overhead largely can be reduced. During a normal hop-by-hop aggregation process in a tree topology, the high-level nodes (i.e., nodes closer to the root) are of greater interest to attackers than the low-level nodes. In other words, if a compromised node is closer to the root, the bogus aggregated data from it will have a larger impact on the final result computed by the root. To alleviate this impact, SDAP [151] takes the approach of reducing the trust on high-level nodes, which is realized by the principle of divide-and-conquer. More specifically, by using a probabilistic grouping method, SDAP dynamically partitions the topology tree into multiple logical groups (sub-trees) of similar sizes. SDAP performs hop-by-hop aggregation in each logical group and generates one aggregate with a commitment from each group. After a group commits its aggregate, this group cannot deny it later. After the BS has collected all the group aggregates, it then identifies the suspicious groups. Finally, each group under suspicion participates in an attestation process to prove the correctness of its group aggregate. The BS discards the individual group aggregate if a group under attestation fails to support its earlier commitment made in the collection phase. The final aggregate is calculated over all the group aggregates that are either normal or have passed the attestation procedure. Because fewer nodes are under a high level node in a logical sub-tree, the potential security threat by a compromised high-level node is reduced.

Resilient Fusion — Another proactive approach to deal with false reports is to design a resilient fusion function so that the impact of attacks is reduced as much as possible at the fusion node. Wagner [152] proposed to use statistical estimation to model the data fusion procedure so that the well-established estimation theory can be used to evaluate the resilience of fusion function. He modeled the fusion function as a random function

$$\hat{\Theta} = f(X_1, \ldots, X_n), \tag{7}$$

and computed the fusion error as

$$rms(f) = \mathbb{E}[(\hat{\Theta} - \theta)^2]^{1/2}, \tag{8}$$

where $X_1, \ldots, X_n$ are the random variables and θ is the real value that should be reported. If k out of n samples are tampered with by adversaries, then the fusion error becomes

$$rms^*(f, A) = \mathbb{E}[(\hat{\Theta}^* - \theta)^2]^{1/2}, \tag{9}$$

where A is a k-node attack and $\hat{\Theta}^*$ is the fusion result based on the samples that were tampered with. Then a fusion function is said to be (k, α)-resilient if $\max\{rms^*(f, A)\} \le \alpha \cdot rms(f)$. The intuition is that the (k, α)-resilient functions, for small α, are the ones that can be computed meaningfully and securely in the presence of up to k compromised or malicious nodes, a feature used for fighting against false reports. This approach provides a sound theoretical basis for designing fusion functions that satisfy specific security requirements.

Location Services

Location information is critical to many location-based services like target tracking. The assurance of the location information is a very important security service because malicious nodes can claim false location information. In the *echo protocol* [153], each node can communicate by using both radio frequency and ultrasound. When a prover node claims a location, a nearby verifier node can send a nonce to the prover node by radio and require the prover to echo the nonce by ultrasound. If the round-trip time of the nonce exceeds an upper bound, the location claim must be false. Multiple verifier nodes can cooperate with each other to verify the location of a prover node.

Lazos and Poovendran [154] proposed a secure range-independent localization (SeRLoc) scheme to determine sensor locations in an untrusted environment. They assume that there

are some powerful nodes, called locators, in a WSN. Each locator is equipped with a GPS device and directional antennas so that it can determine both location and direction. Each locator periodically broadcasts beacons including its location and the broadcasting angle of the antenna. Based on the beacon information and the transmission range of the locator, a sensor node can determine an area that is an overlap of the broadcasting of several locators. The location of the sensor node can be estimated as the center of gravity of the overlapping area. Each sensor shares a symmetric key with a locator, and the key is used to encrypt location information. The broadcast beacons are authenticated by a one-way hash chain.

A distance-based approach is proposed in [155], in which the network manager can use multilateration to determine the position of a device from a set of reference points whose positions are known. The distance between the device and each of the reference points is measured according to the radio frequency (RF)-based distance bounding protocol and collected by the network manager to verify the position of the device. Another similar scheme is secure localization in wireless sensor networks (SLS) [156, 157], in which the distance measures are encrypted and authenticated by the keys shared between reference points and sensor nodes. All the reference points collaborate to determine the location of each node.

Source Location Privacy

An important application of WSNs is to monitor a valuable target that may be of interest to attackers. In this scenario, the information about the target should not be easy to access by attackers. This leads to privacy issues in WSNs. Usually, privacy can be defined as the guarantee that information, which may be associated with message transmission, in addition to message content, is observable or decipherable only by those who intentionally are meant to observe or decipher it [158, 159]. Cryptographic tools can protect the message content, but cannot totally ensure privacy of the information associated with the message content.

A topic of privacy in WSNs is the source location privacy [158, 159]. It is very important in the scenario where the target is of high value to attackers such that attackers want to find out its location by tracing back to the source node monitoring the target. To provide the source location privacy, the routing protocol for WSNs should be resistant to tracing while delivering information to the sink. Two metrics are defined in [158, 159] to evaluate the source location privacy in WSNs. One is the *safety period*, which is the number of new messages initiated by the source node before an attacker locates the target. The other is the *capture likelihood*, which is the probability that the attacker locates the target within a specific time period.

It has been shown in [158, 159] that two popular routing mechanisms for WSNs, flooding and single-path routing, cannot provide efficient source location privacy while maintaining desirable system performance in terms of energy consumption. The reason is that the attacker easily can trace back along the shortest path among all the paths generated by flooding and along the single path.

To enhance the source location privacy, a new routing protocol, called *phantom routing*, is introduced in [158, 159]. In phantom routing, the delivery of every message experiences two phases:

- The random walk phase, which may be a pure random walk or a directed walk, meant to direct the message to a phantom source, that is, a fake source
- A subsequent flooding/single-path routing stage meant to deliver the message to the sink

When the source sends out a message, the message is unicasted randomly for a total of h hops. After the h hops, the message can be flooded or unicasted through a single-path routing to the sink. The introduced phantom source significantly increases the source location privacy while marginally increasing the communication overhead compared with the flooding and the single-path routing.

Intellectual Property Protection

When data sensing and processing becomes the major application of a WSN, the intellectual property is of crucial importance to prevent the duplication of the sensed or processed data. Therefore, there is an urgent requirement to develop intellectual property protection (IPP) techniques. In [160], several watermarking techniques were developed to embed cryptographically encoded authorship signatures into data and information acquired by WSNs. The key idea is to impose additional constraints during the data acquisition or sensor data processing. One technique is to watermark raw data during the sensing operations by modifying the location and orientation of a sensor, time management discipline (e.g., frequency and phase of intervals between consecutive data capturing), and its resolution. The second technique embeds a signature during data processing, in which some variations are introduced into data processing procedures such as error minimization procedures, physical world model building, and solving of computationally intractable problems.

Open Issues

Though WSNs have been found to be useful in a wide range of applications, researchers and engineers are still looking for new and promising applications that can boost the development of the entire area and inspire a profitable market. Different applications may have different security requirements, which may be dependent on specific security designs. Careful security design at low layers including encryption and authentication may not effectively prevent attacks at the application layer because they cannot understand the semantics at the application layer. Adversaries can spread any information that can be verified only by applications. Therefore, it is necessary to incorporate security resilience into the application layer before we deploy a new WSN application.

Conclusion

Security is becoming a major concern for WSN protocol designers because of the wide security-critical applications of WSNs. In this article, we discussed general security problems in WSNs and described corresponding solutions. However, there are still many open issues. On the one hand, WSNs are still under development, and many protocols designed so far for WSNs have not taken security into consideration. On the other hand, the salient features of WSNs make it very challenging to design strong security protocols while still maintaining low overheads. Hence, wireless security for WSNs is still a very fruitful research area to be explored.

References

[1] F. Akyildiz *et al.*, "A Survey on Sensor Networks," *IEEE Commun. Mag.*, vol. 40, no. 8, Aug. 2002, pp. 102–14.

[2] J. M. Kahn, R. H. Katz, and K. S. J. Pister, "Next Century Challenges: Mobile Networking for Smart Dust," *Proc. ACM Int'l. Conf. Mobile Computing and Networking (MobiCom'99)*, Aug. 1999, pp. 217–78.

[3] G. J. Pottie and W. J. Kaiser, "Wireless Integrated Network Sensors," *Commun. ACM*, vol. 43, no. 5, May 2000, pp. 51–58.
[4] Crossbow Technology; http://www.xbow.com/, 2006.
[5] Atmel Corporation; http://www.atmel.com/, 2006.
[6] C. Karlof and D. Wagner, "Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures," Proc. 1st IEEE Int'l. Wksp. Sensor Network Protocols and Applications (SNPA'03), May 2003.
[7] A. Wood and J. Stankovic, "Denial of Service in Sensor Networks," IEEE Computer Mag., vol. 35, no. 10, Oct. 2002, pp. 54–62.
[8] A. Perrig, J. Stankovic, and D. Wagner, "Security in Wireless Sensor Networks," *Commun. ACM*, vol. 47, no. 6, June 2004, pp. 53–57.
[9] E. Shi and A. Perrig, "Designing Secure Sensor Networks," IEEE Wireless Commun. Mag., vol. 11, no. 6, Dec 2004, pp. 38–43.
[10] F. Hu and N. K. Sharma, "Security Considerations in Ad Hoc Sensor Networks," *Elsevier Ad Hoc Networks*, vol. 3, no. 1, 2005, pp. 69–89.
[11] J. P. Walters and Z. Liang, "Wireless Sensor Network Security: A Survey," *Security in Distributed, Grid, and Pervasive Computing*, Ed. Y. Xiao, Auerbach Publications, CRC Press, 2006.
[12] S. Avancha *et al.*, "Security for Wireless Sensor Networks: Overview," *Wireless Sensor Networks*, Ed. C. S. Raghavendra, K. M. Sivalingam, and T. Znati, Kluwer Academic Publishers, 2004.
[13] H. Chan and A. Perrig, "Security for Sensor Networks: Key Management," *Wireless Sensor Networks*, Ed. C. S. Raghavendra, K. M. Sivalingam, and T. Znati, Kluwer Academic Publishers, 2004.
[14] J. Wong *et al.*, "Security in Sensor Networks: Watermarking Techniques," *Wireless Sensor Networks*, Ed. C. S. Raghavendra, K. M. Sivalingam, and T. Znati, Kluwer Academic Publishers, 2004.
[15] S. A. Camtepe and B. Yener, "Key Distribution Mechanisms for Wireless Sensor Networks: A Survey," Technical Report TR-05-07 Rensselaer Polytechnic Institute, Mar. 2005.
[16] C. E. Shannon, "Communication Theory of Secrecy Systems," *Bell Sys. Tech. J.*, vol. 28, Oct. 1949, pp. 656–715.
[17] W. Diffie and M. E. Hellman, "New Directions in Cryptography," *IEEE Trans. Info. Theory*, vol. IT-22, no. 6, 1976, pp. 644–54.
[18] R. L. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Commun. ACM*, vol. 21, no. 2, Feb. 1978, pp. 120–26.
[19] FIPS PUB 46-2, "Data Encryption Standard (DES)," Dec. 1993.
[20] IETF RFC 2040, "The rc5, rc5-cbc, rc5-cbc-pad, and rc5-cts algorithms," Oct. 1996.
[21] S. Basagni *et al.*, "Secure Pebblenets," *Proc. 2nd ACM Int'l. Symp. Mobile Ad Hoc Networking and Computing (Mobihoc'01)*, Long Beach, CA, 2001.
[22] A. Perrig *et al.*, "SPINS: Security Protocols for Sensor Networks," *ACM Wireless Networks*, vol. 8, no. 5, Sept. 2002, pp. 521–34.
[23] R. Blom, "An Optimal Class of Symmetric Key Generation Systems," *Advances in Cryptology: Proc. EUROCRYPT'84*, Lecture Notes in Computer Science, Springer-Verlag, vol. 209, 1985, pp. 335–38.
[24] C. Blundo *et al.*, "Perfectly Secure Key Distribution for Dynamic Conferences," *Advances in Cryptology: Proc. CRYPTO'92*, Lecture Notes in Computer Science, Springer-Verlag, vol. 740, 1992, pp. 471–86.
[25] L. Eschenauer and V. Gligor, "A Key Management Scheme for Distributed Sensor Networks," *Proc. 9th ACM Conf. Computer and Communications Security (CCS'02)*, Washington, DC, Nov. 2002.
[26] J. Spencer, *The Strange Logic of Random Graphs*, Algorithms and Combinatorics 22, Springer-Verlag 2000.
[27] H. Chan, A. Perrig, and D. Song, "Random Key Predistribution Schemes for Sensor Networks," *Proc. 2003 IEEE Symp. Security and Privacy*, May 2003, pp. 197–213.
[28] F. Anjum, "Location Dependent Key Management Using Random Key-Predistribution in Sensor Networks," *Proc. 5th ACM Wksp. Wireless Security (WiSe'06)*, Los Angeles, Sept. 2006.
[29] R. D. Pietro, L. V. Mancini, and A. Mei, "Random Key-Assignment for Secure Wireless Sensor Networks," *Proc. 10th ACM Conf. Computer and Communications Security (CCS'03)*, Washington, DC, Oct. 2003.
[30] W. Du *et al.*, "A Pairwise Key Pre-Distribution Scheme for Wireless Sensor Networks," *Proc. 10th ACM Conf. Computer and Communications Security (CCS'03)*, Washington, DC, Oct. 2003.
[31] D. Liu and P. Ning, "Establishing Pairwise Keys in Distributed Sensor Networks," *Proc. 10th ACM Conf. Computer and Communications Security (CCS'03)*, Washington, DC, Oct. 2003.
[32] J. Hwang and Y. Kim, "Revisiting Random Key Pre-Distribution Schemes for Wireless Sensor Networks," *Proc. 2nd ACM Wksp. Security of Ad Hoc and Sensor Networks (SASN'04)*, Washington, DC, Oct. 2004.
[33] J. Lee and D. R. Stinson, "Deterministic Key Pre-Distribution Schemes for Distributed Sensor Networks," *Proc. 11th Int'l. Wksp. Selected Areas in Cryptography (SAC'04)*, Lecture Notes in Computer Science, Springer-Verlag, vol. 3357/2004, 2004, pp. 294–307.
[34] J. Lee and D. R. Stinson, "A Combinatorial Approach to Key Pre-Distribution Mechanisms for Wireless Sensor Networks," *Proc. 2005 IEEE Wireless Commun. and Networking Conf. (WCNC'05)*, New Orleans, Mar. 2005.
[35] S. A. Camtepe and B. Yener, "Combinatorial Design of Key Distribution Mechanisms for Wireless Sensor Networks," *Proc. 9th European Symp. Research in Computer Security (ESORICS 2004)*, Sophia Antipolis, France, Sept. 2004.
[36] D. S. Sanchez and H. Baldus, "A Deterministic Pairwise Key Pre-Distribution Scheme for Mobile Sensor Networks," *Proc. 1st IEEE Int'l. Conf. Security and Privacy for Emerging Areas in Communications Networks (SecureComm'05)*, Sep. 2005.
[37] Y. Zhou and Y. Fang, "A Scalable Key Agreement Scheme for Large Scale Networks," *Proc. 2006 IEEE Int'l. Conf. Networking, Sensing and Control (ICNSC'06)*, Ft. Lauderdale, Florida, Apr. 2006.
[38] Y. Zhou and Y. Fang, "Scalable and Deterministic Key Agreement for Large Scale Networks," *IEEE Trans. Wireless Commun.*, vol. 6, no. 11, Nov. 2007.
[39] H. Chan and A. Perrig, "PIKE: Peer Intermediaries for Key Establishment in Sensor Networks," *Proc. 24th Annual IEEE Joint Conf. IEEE Computer and Communications Societies (INFOCOM'05)*, Miami, FL, Mar. 2005.
[40] D. Liu, P. Ning, and R. Li, "Establishing Pairwise Keys in Distributed Sensor Networks," *ACM Trans. Info. and System Security*, vol. 8, no. 1, Feb. 2005, pp. 41–77.
[41] F. Delgosha and F. Fekri, "Key Pre-Distribution in Wireless Sensor Networks Using Multivariate Polynomials," *Proc. 2nd Annual IEEE Communications Society Conf. Sensor and Ad Hoc Communications and Networks (SECON'05)*, Sept. 2005.
[42] F. Delgosha and F. Fekri, "Threshold Key-Establishment in Distributed Sensor Networks Using a Multivariate Scheme," *Proc. the 25th Annual IEEE Joint Conf. IEEE Computer and Communications Societies (INFOCOM'06)*, Barcelona, Spain, Apr. 2006.
[43] D. Liu and P. Ning, "Location-Based Pairwise Key Establishments for Relatively Static Sensor Networks," *Proc. 2003 ACM Wksp. Security of Ad Hoc and Sensor Networks(SASN'03)*, Fairfax, VA, Oct. 2003.
[44] W. Du *et al.*, "A Key Management Scheme for Wireless Sensor Networks Using Deployment Knowledge," *Proc. 23rd Annual IEEE Joint Conf. IEEE Computer and Communications Societies (INFOCOM'04)*, Hong Kong, Mar. 2004.
[45] D. Huang *et al.*, "Location-Aware Key Management Scheme for Wireless Sensor Networks," *Proc. 2nd ACM Wksp. Security of Ad Hoc and Sensor Networks (SASN'04)*, Washington, DC, Oct. 2004.
[46] Z. Yu and Y. Guan, "A Robust Group-Based Key Management Scheme for Wireless Sensor Networks," *Proc. 2005 IEEE Wireless Commun.and Networking Conf. (WCNC'05)*, New Orleans, LA, Mar. 2005.
[47] Y. Zhou, Y. Zhang, and Y. Fang, "LLK: A Link-Layer Key Establishment Scheme in Wireless Sensor Networks," *Proc. 2005 IEEE Wireless Commun. and Networking Conf. (WCNC'05)*, New Orleans, LA, Mar. 2005.
[48] Y. Zhou, Y. Zhang, and Y. Fang, "Key Establishment in Sensor

Networks Based on Triangle Grid Deployment Model," *Proc. 2005 IEEE Military Communications Conf. (MILCOM'05)*, Atlantic City, NJ, Oct. 2005.
[49] Y. Zhou and Y. Fang, "Scalable Link-Layer Key Agreement in Sensor Networks," *Proc. 2006 IEEE Military Communications Conf. (MILCOM'06)*, Washington, DC, Oct. 2005.
[50] Y. Zhou and Y. Fang, "A Two-Layer Key Establishment Scheme for Wireless Sensor Networks," *IEEE Trans. Mobile Computing*, vol. 6, no. 9, Sept. 2007, pp. 1009–20.
[51] D. Liu, P. Ning, and W. Du, "Group-Based Key Pre-Distribution in Wireless Sensor Networks," *Proc. 4th ACM Wksp. Wireless Security (WiSe'05)*, Cologne, Germany, Sept. 2005.
[52] L. Zhou, J. Ni, and C.V. Ravishankar, "Efficient Key Establishment for Group-Based Wireless Sensor Deployments," *Proc. 4th ACM Wksp. Wireless Security (WiSe'05)*, Cologne, Germany, Sept. 2005.
[53] L. Zhou, J. Ni, and C. V. Ravishankar, "Supporting Secure Communication and Data Collection in Mobile Sensor Networks," *Proc. the 25th Annual IEEE Joint Conf. IEEE Computer and Commun. Societies (INFOCOM'06)*, Barcelona, Spain, Apr. 2006.
[54] T. Ito *et al.*, "A Key Pre-Distribution Scheme for Secure Sensor Networks Using Probability Density Function of Node Deployment," *Proc. the 3rd ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'05)*, Alexandria, VA, Nov. 2005.
[55] S. Zhu, S. Setia, and S. Jajodia, "LEAP: Efficient Security Mechanism for Large-Scale Distributed Sensor Networks," *Proc. 10th ACM Conf. Computer and Commun. Security (CCS'03)*, Washington, DC, Oct. 2003.
[56] J. Deng *et al.*, "A Practical Study of Transitory Master Key Establishment for Wireless Sensor Networks," *Proc. 1st IEEE Int'l. Conf. Security and Privacy for Emerging Areas in Commun. Networks (SecureComm'05)*, Sept. 2005.
[57] E-O Blab and M. Zitterbart, "An Efficient Key Establishment Scheme for Secure Aggregating Sensor Networks," *Proc. 2006 ACM Symp. Information, Computer and Commun. Security (ASIACCS'06)*, Taipei, Taiwan, Mar. 2006.
[58] R. Anderson, H. Chan, and A. Perrig, "Key Infection: Smart Trust for Smart Dust," *Proc. 12th IEEE Int'l. Conf. Network Protocols (ICNP'04)*, Berlin, Germany, Oct. 2004.
[59] M. J. Miller and N. H. Vaidya, "Leveraging Channel Diversity for Key Establishment in Wireless Sensor Networks," *Proc. 25th Annual IEEE Joint Conf. IEEE Computer and Commun. Societies (INFOCOM'06)*, Barcelona, Spain, Apr. 2006.
[60] P. Traynor *et al.*, "Establishing Pair-Wise Keys in Heterogeneous Sensor Networks," *Proc. 25th Annual IEEE Joint Conf. IEEE Computer and Commun. Societies (INFOCOM'06)*, Barcelona, Spain, Apr. 2006.
[61] P. Traynor *et al.*, "LIGER: Implementing Efficient Hybrid Security Mechanisms for Heterogeneous Sensor Networks," *Proc. 4th Int'l. Conf. Mobile Systems, Applications and Services (MobiSys'06)*, Uppsala, Sweden, June 2006.
[62] M. Shehab, E. Bertino, and A. Ghafoor, "Efficient Hierarchical Key Generation and Key Diffusion for Sensor Networks," *Proc. 2nd Annual IEEE Communications Society Conf. Sensor and Ad Hoc Communications and Networks (SECON'05)*, Sept. 2005.
[63] R. Watro *et al.*, "TinyPK: Securing Sensor Networks with Public Key Technology," *Proc. 2nd ACM Wksp. Security of Ad Hoc and Sensor Networks (SASN'04)*, Washington, DC, Oct. 2004.
[64] G. Gaubatz, J. Kaps, and B. Sunar, "Public Key Cryptography in Sensor Networks — Revisited," *Proc. 1st European Wksp. Security in Ad Hoc and Sensor Networks (ESAS'04)*, Heidelberg, Germany, Aug. 2004, Revised Selected Papers, Lecture Notes in Computer Science, Springer-Verlag, vol. 3313/2005, 2005, pp. 2–18.
[65] A. Menezes, P. van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, Oct. 1996.
[66] J. Hoffstein, J. Pipher, and J. H. Silverman, "NTRU: A Ring-Based Public Key Cryptosystem," *Proc. ANTS III,* Lecture Notes in Computer Science, Springer-Verlag, vol. 1423/1998, 1998, pp. 267–88.
[67] N. Koblitz, "Elliptic Curve Cryptosystems," *Mathematics of Computation*, vol. 48, 1987, pp. 203–9.
[68] V. Miller, "Uses of Elliptic Curves in Cryptography," *Proc. Advances in Cryptology — CRYPTO'85*, Lecture Notes in Computer Science, Springer-Verlag, vol. 218, 1986, pp. 417–26.
[69] D. J. Malan, M. Welsh, and M. D. Smith, "A Public-Key Infrastructure for Key Distribution in TinyOS Based on Elliptic Curve Cryptography," *Proc. 1st IEEE Int'l. Conf. Sensor and Ad Hoc Commun. and Networks (SECON'04)*, Santa Clara, CA, Oct. 2004.
[70] N. Gura *et al.*, "Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs," *Proc. 6th Int'l. Wksp. Cryptographic Hardware and Embedded Systems (CHES'04)*, Lecture Notes in Computer Science, Springer-Verlag, vol. 3156/2004, 2004, pp. 119–32.
[71] Q. Huang *et al.*, "Fast Authenticated Key Establishment Protocols for Self-Organizing Sensor Networks," *Proc. 2nd ACM Int'l. Conf. Wireless Sensor Networks and Applications (WSNA'03)*, San Diego, CA, Sept. 2003.
[72] Y. Zhou, Y. Zhang, and Y. Fang, "Access Control in Wireless Sensor Networks," *Elsevier Ad Hoc Networks Journal*, Special Issue on Security in Ad Hoc and Sensor Networks, vol. 5, 2007, pp. 3–13.
[73] S. Vanstone, "Responses to NIST's Proposal," *Commun. ACM*, vol. 35, July 1992, pp. 50–52.
[74] W. Du, R. Wang, and P. Ning, "An Efficient Scheme for Authenticating Public Keys in Sensor Networks," *Proc. 6th ACM Int'l. Symp. Mobile Ad Hoc Networking and Computing (MobiHoc'05)*, Urbana-Champaign, IL, May 2005.
[75] R. Merkle, "Protocols for Public Key Cryptosystems," *Proc. 1980 IEEE Symp. Research in Security and Privacy (SP'80)*, Los Alamitos, CA, Apr. 1980.
[76] D. Boneh and M. Franklin, "Identity-Based Encryption from the Weil Pairing," Proc. Advances in Cryptology — CRYPTO'01, Lecture Notes in Computer Science, Springer-Verlag, vol. 2139, 2001, pp. 213–29.
[77] Y. Zhang *et al.*, "Securing Sensor Networks with Location-Based Keys," *Proc. 2005 IEEE Wireless Commun. and Networking Conf. (WCNC'05)*, New Orleans, LA, Mar. 2005.
[78] Y. Zhang *et al.*, "Location-Based Compromise-Tolerant Security Mechanisms for Wireless Sensor Networks," *IEEE JSAC*, special issue on Security in Wireless Ad Hoc Networks, vol. 24, no. 2, Feb. 2006, pp. 247–60.
[79] A. Chadha, Y. Liu, and S. K. Das, "Group Key Distribution via Local Collaboration in Wireless Sensor Networks," *Proc. 2nd Annual IEEE Communications Society Conf. Sensor and Ad Hoc Communications and Networks (SECON'05)*, Sep. 2005.
[80] J. Huang, J. Buchingham, and R. Han, "A Level Key Infrastructure for Secure and Efficient Group Communication in Wireless Sensor Networks," *Proc. 1st IEEE Int'l. Conf. Security and Privacy for Emerging Areas in Commun. Networks (SecureComm'05)*, Sept. 2005.
[81] H. Chan *et al.*, "On the Distribution and Revocation of Cryptographic Keys in Sensor Networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 2, no. 3, July-Sept. 2005, pp. 233–47.
[82] M. Bellare, R. Canetti, and H. Krawczyk, "Keying Hash Functions for Message Authentication," *Proc. Advances in Cryptology Crypto' 96*, Lecture Notes in Computer Science, Springer-Verlag, vol. 1109, 1996.
[83] IETF RFC 4346, The Transport Layer Security (TLS) Protocol Version 1.1, Apr. 2006.
[84] Chris Karlof, Naveen Sastry, and David Wagner, "TinySec: A Link Layer Security Architecture for Wireless Sensor Networks," *Proc. 2nd Int'l. Conf. Embedded Networked Sensor Systems (SenSys'04),* Baltimore, MD, Nov. 2004.
[85] National Security Agency, Skipjack and KEA algorithm specifications. May 1998.
[86] W. Lou, W. Liu, and Y. Fang, "SPREAD: Enhancing Data Confidentiality in Mobile Ad Hoc Networks," *Proc. 23rd Annual IEEE Joint Conf. IEEE Computer and Commun. Societies (INFOCOM'04)*, Hong Kong, Mar 2004.
[87] A. Shamir, "How to Share a Secret," *Commun. ACM*, vol. 22, no. 11, Nov. 1979, pp. 612–13.
[88] M. Bohge and W. Trappe, "An Authentication Framework for Hierarchical Ad Hoc Sensor Networks," *Proc. 2nd ACM Workshop on Wireless Security (WiSe'03)*, San Diego, CA, 2003.
[89] A. Perrig *et al.*, "TESLA: Multicast Source Authentication Transform Introduction," IETF working draft, draft-ietf-msec-

tesla-intro-01.txt.
[90] J. Deng, R. Han, and S. Mishra, "Security Support for In-Network Processing in Wireless Sensor Networks," *Proc. 2003 ACM Wksp. Security of Ad Hoc and Sensor Networks (SASN'03)*, Fairfax, VA, Oct. 2003.
[91] D. Liu and P. Ning, "Efficient Distribution of Key Chain Commitments for Broadcast Authentication in Distributed Sensor Networks," *Proc. 10th Annual Network and Distributed System Security Symp. (NDSS'03)*, San Diego, CA, Feb. 2003.
[92] M. Manzo, T. Roosta, and S. Sastry, "Time Synchronization Attacks in Sensor Networks," *Proc. 3rd ACM Wksp. Security of Ad Hoc and Sensor Networks (SASN'05)*, Alexandria, VA, Nov. 2005.
[93] S. Ganeriwal *et al.*, "Secure Time Synchronization Service for Sensor Networks," *Proc. 4th ACM Wksp. Wireless Security (WiSe'05)*, Cologne, Germany, Sept. 2005.
[94] Y. Zhou and Y. Fang, "BABRA: Batch-Based Broadcast Authentication in Wireless Sensor Networks," *Proc. 49th annual IEEE Global Telecommun. Conf. (GLOBECOM'06)*, San Francisco, CA, Nov. 2006.
[95] K. Ren *et al.*, "On Broadcast Authentication in Wireless Sensor Networks," *Proc. Int'l. Conf. Wireless Algorithms, Systems, and Applications (WASA'06)*, Xi'an, China, Aug. 2006.
[96] F. Hess, "Efficient Identity-Based Signature Schemes Based on Pairings," *Proc. 9th Annual Int'l. Wksp. Selected Areas in Cryptography (SAC'02)*, Aug. 2002.
[97] S-M Chang *et al.*, "An Efficient Broadcast Authentication Scheme in Wireless Sensor Networks," *Proc. 2006 ACM Symp. Information, Computer and Communications Security (ASIACCS'06)*, Taipei, Taiwan, Mar. 2006.
[98] S. Rafaeli and D. Hutchison, "A Survey of Key Management for Secure Group Communication," *ACM Computing Surveys*, vol. 35, no. 3, Sept. 2003, pp. 309–29.
[99] J. A. Al-Karaki and A. E. Kamal, "Routing Techniques in Wireless Sensor Networks: A Survey," *IEEE Wireless Commun.*, vol. 11, no. 6, Dec. 2004.
[100] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," *Proc. 6th Annual ACM Int'l. Conf. Mobile Computing and Networking (MobiCom'00)*, Boston, MA, Aug. 2000.
[101] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," *Proc. 33rd Hawaii Int'l. Conf. System Sciences (HICSS'00)*, Jan. 2000.
[102] Y. Yu, D. Estrin, and R. Govindan, "Geographical and Energy-Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks," UCLA Comp. Sci. Dept. Tech. Rep., UCLA-CSD TR-010023, May 2001.
[103] J. Newsome *et al.*, "The Sybil Attack in Sensor Networks: Analysis and Defenses," *Proc. 3rd IEEE Int'l. Symp. Information Processing in Sensor Networks (IPSN'04)*, Berkeley, CA, Apr. 2004.
[104] J. R. Douceur, "The Sybil Attack," *Proc. 1st ACM Int'l. Wksp. Peer-to-Peer Systems (IPTPS'02)*, Mar. 2002.
[105] Y. Hu, A. Perrig, and D. B. Johnson, "Packet Leashes: A Defense against Wormhole Attacks in Wireless Networks," *Proc. 22nd Annual Joint Conf. IEEE Computer and Communications Societies (INFOCOM'03)*, San Francisco, CA, Mar. 2003.
[106] Y. Hu, A. Perrig, and D. Johnson, "Rushing Attacks and Defense in Wireless Ad Hoc Network Routing Protocols," *Proc. 2003 ACM Wksp. Wireless Security (WiSe'03)*, San Diego, CA, Sept. 2003.
[107] P. Papadimitrators and Z. Haas, "Secure Routing for Mobile Ad Hoc Networks," *Proc. SCS Commun. Networks and Distributed System Modeling and Simulation Conf. (CNDS'02)*, San Antonio, TX, Jan. 2002.
[108] Y. Hu, A. Perrig, and D. Johnson, "Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks," *Proc. 8th Annual ACM Int'l. Conf. Mobile Computing and Networking (Mobicom'02*), Atlanta, GA, Sept. 2002.
[109] M. Zapata and N. Asokan, "Securing Ad Hoc Routing Protocols," *Proc. 3rd ACM Wksp. Wireless Security (WiSe'02)*, Atlanta, GA, Sep. 2002.
[110] Y. Hu, D. Johnson, and A. Perrig, "SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks," *Proc. 4th IEEE Wksp. Mobile Computing Systems and Applications (WMCSA'02)*, Callicoon, NY, June 2002.
[111] P. Papadimitrators and Z. Haas, "Secure Link State Routing for Mobile Ad Hoc Networks," *Proc. 2003 IEEE Symp. Applications and the Internet Wksps.*, Jan. 2003.
[112] B. Dahill *et al.*, "A Secure Protocol for Ad Hoc Networks," *Proc. 10th IEEE Int'l. Conf. Network Protocols (ICNP'02)*, Paris, France, Nov. 2002.
[113] H. Yang *et al.*, "Security in Mobile Ad Hoc Networks: Challenges and Solutions," *IEEE Wireless Commun.*, vol. 11, no. 1, Feb. 2004, pp. 38–47.
[114] J. Deng, R. Han, and S. Mishra, "A Performance Evaluation of Intrusion-Tolerant Routing in Wireless Sensor Networks," *Proc. 2nd IEEE Int'l. Wksp. Info. Processing in Sensor Networks (IPSN'03)*, Palo Alto, CA, Apr. 2003.
[115] R. D. Pietro *et al.*, "LKHW: A Directed Diffusion-Based Secure Multicast Scheme for Wireless Sensor Networks," *Proc. 2003 IEEE Int'l. Conf. Parallel Processing Wksps. (ICPP'03)*, Oct. 2003.
[116] M. Tubaishat *et al.*, "A Secure Hierarchical Model for Sensor Networks," *ACM SIGMOD Record*, vol. 33, no. 1, Mar. 2004.
[117] Ross Anderson and Markus Kuhn, "Tamper Resistance — A Cautionary Note," *Proc. 2nd USENIX Wksp. Electronic Commerce*, Oakland, CA, Nov. 1996.
[118] B. Parno, A. Perrig, and V. Gligor, "Distributed Detection of Node Replication Attacks in Sensor Networks," *Proc. 2005 IEEE Symp. Security and Privacy (SP'05)*, Oakland, CA, May 2005.
[119] G. Wang *et al.*, "On Supporting Distributed Collaboration in Sensor Networks," *Proc. 2003 IEEE Military Commun.Conf. (MILCOM'03)*, Boston, MA, Oct. 2003.
[120] L. Hu and D. Evans, "Using Directional Antennas to Prevent Wormhole Attacks," *Proc. 11th Annual Network and Distributed System Security Symp. (NDSS'04)*, San Diego, CA, Feb. 2004.
[121] W. Wang and B. Bhargava, "Visualization of Wormholes in Sensor Networks," *Proc. 2004 ACM Wksp. Wireless Security (Wise'04)*, Philadelphia, PA, Oct. 2004.
[122] I. Khalil, S. Bagchi, and C. Nita-Rotaru, " DICAS: Detection, Diagnosis and Isolation of Control Attacks in Sensor Networks," *Proc. 1st IEEE Int'l. Conf. Security and Privacy for Emerging Areas in Commun. Networks (SecureComm'05)*, Sep. 2005.
[123] S. Ganeriwal and M. Srivastava, "Reputation-Based Framework for High Integrity Sensor Networks," *Proc. 2nd ACM Wksp. Security of Ad Hoc and Sensor Networks (SASN'04)*, Washington, DC, Oct. 2004.
[124] C-C. Su *et al.*, "The New Intrusion Prevention and Detection Approaches for Clustering-Based Sensor Networks," *Proc. 2005 IEEE Wireless Commun. and Networking Conf. (WCNC'05)*, New Orleans, LA, Mar. 2005.
[125] A. Seshadri *et al.*, "SCUBA: Secure Code Update by Attestation in Sensor Networks," *Proc. 5th ACM Wksp. Wireless Security (WiSe'06)*, Los Angeles, Sept. 2006.
[126] W. Xu *et al.*, "The Feasibility of Launching and Detecting Jamming Attacks in Wireless Networks," *Proc. 6th ACM Int'l. Symp. Mobile Ad Hoc Networking and Computing (MobiHoc'05)*, Urbana-Champaign, IL, May. 2005.
[127] Y. W. Law, "Energy-Efficient Link-Layer Jamming Attacks against Wireless Sensor Network MAC Protocols," *Proc. 3rd ACM Wksp. Security of Ad Hoc and Sensor Networks (SASN'05)*, Alexandria, VA, Nov. 2005.
[128] I. Aad, J. Hubaux, and E. Knightly, "Denial of Service Resilience in Ad Hoc Networks," *Proc. 10th Annual ACM Int'l. Conf. Mobile Computing and Networking (MobiCom'04)*, Philadelphia, PA, Sept. 2004.
[129] J. M. McCune, "Detection of Denial-of-Message Attacks on Sensor Network Broadcasts," *Proc. 2005 IEEE Symp. Security and Privacy (SP'05)*, Oakland, CA, May 2005.
[130] J. Deng, R. Han, and S. Mishra, "Defending against Path-Based DoS Attacks in Wireless Sensor Networks," *Proc. 3rd ACM Wksp. Security of Ad Hoc and Sensor Networks (SASN'05)*, Alexandria, VA, Nov. 2005.
[131] J. Deng, R. Han, and S. Mishra, "Enhancing Base Station Security in Wireless Sensor Networks," Technical Report CU-CS-

951-03, Department of Computer Science, Univ. of Colorado, Apr. 2003.
[132] J. Deng, R. Han, and S. Mishra, "Countermeasures against Traffic Analysis Attacks in Wireless Sensor Networks," *Proc. 1st IEEE Int'l. Conf. Security and Privacy for Emerging Areas in Commun. Networks (SecureComm'05)*, Sept. 2005.
[133] W. Zhang *et al.*, "Least Privilege and Privilege Deprivation: Towards Tolerating Mobile Sink Compromises in Wireless Sensor Networks," *Proc. 6th ACM Int'l. Symp. Mobile Ad Hoc Networking and Computing (MobiHoc'05)*, Urbana-Champaign, IL, May 2005.
[134] J. O. Kephart and D. M. Chess, "The Vision of Autonomic Computing," *IEEE Computer Mag.*, vol. 36, no. 1, Jan. 2003, pp. 41–50.
[135] C. Boutilier *et al.*, "Cooperative Negotiation in Autonomic Systems Using Incremental Utility Elicitation," *Proc. 19th Conf. Uncertainty in Artificial Intelligence (UAI'03)*, Acapulco, Mexico, Aug. 2003.
[136] G. Pacifici, "Performance Management for Cluster-Based Web Services," IBM Technical Report, 2003.
[137] W. Du *et al.*, "A Witness-Based Approach for Data Fusion Assurance in Wireless Sensor Networks," *Proc. 2003 IEEE Global Commun. Conf. (GLOBECOM'03)*, San Francisco, CA, Dec. 2003.
[138] B. Przydatek, D. Song, and A. Perrig, "SIA: Secure Information Aggregation in Sensor Networks," *Proc. 1st ACM Int'l. Conf. Embedded Networked Sensor Systems (SenSys'03)*, Los Angeles, CA, Nov. 2003.
[139] Fan Ye *et al.*, "Statistical En-Route Filtering of Injected False Data in Sensor Networks," *Proc. 23rd Annual IEEE Joint Conf. IEEE Computer and Commun. Societies (INFOCOM'04)*, Hong Kong, China, Mar. 2004.
[140] S. Zhu *et al.*, "An Interleaved Hop-by-Hop Authentication Scheme for Filtering False Data in Sensor Networks," *Proc. IEEE Symp. Security and Privacy (SP'04)*, Oakland, CA, May 2004.
[141] W. Zhang and G. Cao, "Group Rekeying for Filtering False Data in Sensor Networks: A Predistribution and Local Collaboration-Based Approach," *Proc. 24th Annual IEEE Joint Conf. IEEE Computer and Commun. Societies (INFOCOM'05)*, Miami, FL, Mar. 2005.
[142] H. Yang *et al.*, "Toward Resilient Security in Wireless Sensor Networks," *Proc. 6th ACM Int'l. Symp. Mobile Ad Hoc Networking and Computing (MobiHoc'05)*, Urbana-Champaign, IL, May 2005.
[143] K. Ren, W. Lou, and Y. Zhang, "LEDS: Providing Location-Aware End-to-End Data Security in Wireless Sensor Networks," *Proc. 25th Annual IEEE Joint Conf. IEEE Computer and Commun. Societies (INFOCOM'06)*, Barcelona, Spain, Apr. 2006.
[144] Z. Yu and Y. Guan, "A Dynamic En-Route Scheme for Filtering False Data Injection in Wireless Sensor Networks," *Proc. 25th Annual IEEE Joint Conf. IEEE Computer and Commun. Societies (INFOCOM'06)*, Barcelona, Spain, Apr. 2006.
[145] D. Estrin *et al.*, "Next Century Challenges: Scalable Coordination in Sensor Networks," *Proc. 5th Annual ACM/IEEE Int'l. Conf. Mobile Computing and Networking (MobiCom'99)*, Seattle, WA, Aug. 1999.
[146] C. Intanagonwiwat *et al.*, "Impact of Network Density on Data Aggregation in Wireless Sensor Networks," *Proc. 22nd IEEE Int'l. Conf. Distributed Computing Systems (ICDCS'02)*, Vienna, Austria, July 2002.
[147] B. Krishnamachari, D. Estrin, and S. Wicker, "The Impact of Data Aggregation in Wireless Sensor Networks," *Proc. 2002 Int'l. Wksp. Distributed Event-Based Systems, (DEBS'02)*, Vienna, Austria, July 2002.
[148] S. Madden *et al.*, "TAG: A Tiny Aggregation Service for Ad Hoc Sensor Networks," *Proc. 5th USENIX Symp. Operating Systems Design and Implementation (OSDI'02)*, Boston, MA, Dec. 2002.
[149] C. Castelluccia, E. Mykletun, and G. Tsudik, "Efficient Aggregation of Encrypted Data in Wireless Sensor Networks," *Proc. 2nd Annual Int'l. Conf. Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous'05)*, San Diego, CA, July 2005.
[150] J-Y. Chen, G. Pandurangan, and D. Xu, "Robust Computation of Aggregates in Wireless Sensor Networks: Distributed Randomized Algorithms and Analysis," *Proc. 4th Int'l. Symp. Information Processing in Sensor Networks (IPSN'05)*, Los Angeles, CA, Apr. 2005.
[151] Y. Yang, X. Wang, and S. Zhu, "SDAP: A Secure Hop-by-Hop Data Aggregation Protocol for Sensor Networks," *Proc. 7th ACM Int'l. Symp. Mobile Ad Hoc Networking and Computing (Mobihoc'06)*, Florence, Italy, May 2006.
[152] D. Wagner, "Resilient Aggregation in Sensor Networks," *Proc. 2nd ACM Wksp. Security of Ad Hoc and Sensor Networks (SASN'04)*, Washington, DC, Oct. 2004.
[153] N. Sastry, U. Shankar, and D. Wagner, "Secure Verification of Location Claims," *Proc. 2003 ACM Wksp. Wireless Security (WiSe'03)*, San Diego, CA, Sept. 2003.
[154] L. Lazos and R. Poovendran, "SeRLoc: Secure Range-Independent Localization for Wireless Sensor Networks," *Proc. 2004 ACM Wksp. Wireless Security (WiSe'04)*, Philadelphia, PA, Oct. 2004.
[155] S. Capkun and J-P. Hubaux, "Secure Positioning of Wireless Devices with Application to Sensor Networks," *Proc. 24th Annual IEEE Joint Conf. IEEE Computer and Communications Societies (INFOCOM'05)*, Miami, FL, Mar. 2005.
[156] Y. Zhang, W. Liu, and Y. Fang, "Secure Localization in Wireless Sensor Networks," *Proc. 2005 IEEE Military Commun. Conf. (MILCOM'05)*, Atlantic City, NJ, Oct. 2005.
[157] Y. Zhang *et al.*, "Secure Localization and Authentication in Ultra-Wideband Sensor Networks," *IEEE JSAC*, Special Issue on UWB Wireless Communications — Theory and Applications, vol. 24, no. 4, Apr. 2006, pp. 829–35.
[158] C. Ozturk, Y. Zhang, and W. Trappe, "Source-Location Privacy in Energy-Constrained Sensor Network Routing," *Proc. 2nd ACM Wksp. Security of Ad Hoc and Sensor Networks (SASN'04)*, Washington, DC, Oct. 2004.
[159] P. Kamat *et al.*, "Enhancing Source-Location Privacy in Sensor Network Routing," *Proc. 25th IEEE Int'l. Conf. Distributed Computing Systems (ICDCS'05)*, Columbus, OH, June 2005.
[160] J. Feng and M. Potkonjak, "Real-Time Watermarking Techniques for Sensor Networks," *ES&T/SPIE, Security and Watermarking of Multimedia Contents*, Santa Clara, CA, Jan. 2003.

BIOGRAPHIES

YUN ZHOU [S] (yzufl@ufl.edu) received a B.E. degree in electronic information engineering (2000) and an M.E. degree in communication and information systems (2003) from the Department of Electronic Engineering and Information Science at the University of Science and Technology of China, Hefei, China. He is currently pursuing a Ph.D. degree in the Department of Electrical and Computer Engineering at the University of Florida, Gainesville. His research interests are in the areas of security, cryptography, wireless communications and networking, signal processing, and operating systems.

YANCHAO ZHANG (yczhang@njit.edu) received a B.E. degree in computer communications from Nanjing University of Posts and Telecommunications, Nanjing, China, in 1999 and an M.E. degree in computer applications from Beijing University of Posts and Telecommunications, Beijing, China, in 2002. After receiving a Ph.D. degree in electrical and computer engineering from the University of Florida in 2006, he joined the Department of Electrical and Computer Engineering at the New Jersey Institute of Technology as an assistant professor. His research interests are network and distributed system security, wireless networking, and mobile computing, with emphasis on mobile ad hoc networks, wireless sensor networks, wireless mesh networks, and heterogeneous wired/wireless networks.

YUGUANG FANG [SM] (fang@ece.ufl.edu) received a Ph.D. degree in Systems Engineering from Case Western Reserve University in January 1994 and a Ph.D. degree in Electrical Engineering from Boston University in May 1997. He was an assistant professor in the Department of Electrical and Computer Engineering at New Jersey Institute of Technology from July 1998 to May 2000. He then joined the Department of Electrical and Computer Engineering at University of Florida in May 2000 as an assistant professor and got an early promotion to an associate professor with tenure

in August 2003 and to a full professor in August 2005. He holds a University of Florida Research Foundation (UFRF) Professorship from 2006 to 2009. He has published over 200 papers in refereed professional journals and conferences. He received the National Science Foundation Faculty Early Career Award in 2001 and the Office of Naval Research Young Investigator Award in 2002, and is the recipient of the Best Paper Award in IEEE International Conference on Network Protocols (ICNP) in 2006 and the recipient of the IEEE TCGN Best Paper Award in the IEEE High-Speed Networks Symposium, IEEE Globecom in 2002. He is also active in professional activities. He is a Fellow of IEEE and a member of ACM. He has served on several editorial boards of technical journals including *IEEE Transactions on Communications*, *IEEE Transactions on Wireless Communications*, *IEEE Wireless Communications Magazine* and *ACM Wireless Networks*. He was an editor for *IEEE Transactions on Mobile Computing* and currently serves on its Steering Committee. He has been actively participating in professional conference organizations such as serving as the Steering Committee Co-Chair for QShine, the Technical Program Vice-Chair for IEEE INFOCOM'05, Technical Program Symposium Co-Chair for IEEE Globecom'04, and a member of Technical Program Committee for IEEE INFOCOM (1998, 2000, 2003–2009).