Security in Distributed, Grid, and Pervasive Computing
Yang Xiao,(Eds.) pp. – – –

Chapter 17

Wireless Sensor Network Security: A Survey*

John Paul Walters, Zhengqiang Liang,
Weisong Shi, and Vipin Chaudhary
Department of Computer Science
Wayne State University
E-mail: {jwalters, sean, weisong, vipin}@wayne.edu

1 Abstract

As wireless sensor networks continue to grow, so does the need for effective security mechanisms. Because sensor networks may interact with sensitive data and/or operate in hostile unattended environments, it is imperative that these security concerns be addressed from the beginning of the system design. However, due to inherent resource and computing constraints, security in sensor networks poses different challenges than traditional network/computer security. There is currently enormous research potential in the field of wireless sensor network security. Thus, familiarity with the current research in this field will benefit researchers greatly. With this in mind, we survey the major topics in wireless sensor network security, and present the obstacles and the requirements in the sensor security, classify many of the current attacks, and finally list their corresponding defensive measures.

2 Introduction

Wireless sensor networks are quickly gaining popularity due to the fact that they are potentially low cost solutions to a variety of real-world chal-

*This research was supported in part by NSF IGERT grant 9987598, NSF MRI grant 9977815, and NSF ITR grant 0081696 as well as the Wayne State Institute for Scientific Computing.

lenges [1]. Their low cost provides a means to deploy large sensor arrays in a variety of conditions capable of performing both military and civilian tasks. But sensor networks also introduce severe resource constraints due to their lack of data storage and power. Both of these represent major obstacles to the implementation of traditional computer security techniques in a wireless sensor network. The unreliable communication channel and unattended operation make the security defenses even harder. Indeed, as pointed out in [65], wireless sensors often have the processing characteristics of machines that are decades old (or longer), and the industrial trend is to reduce the cost of wireless sensors while maintaining similar computing power. With that in mind, many researchers have begun to address the challenges of maximizing the processing capabilities and energy reserves of wireless sensor nodes while also securing them against attackers. All aspects of the wireless sensor network are being examined including secure and efficient routing [15, 41, 62, 79], data aggregation [22, 33, 54, 68, 75, 91], group formation [6, 42, 69], and so on.

In addition to those traditional security issues, we observe that many general-purpose sensor network techniques (particularly the early research) assumed that all nodes are cooperative and trustworthy. This is not the case for most, or much of, real-world wireless sensor networking applications, which require a certain amount of trust in the application in order to maintain proper network functionality. Researchers therefore began focusing on building a sensor trust model to solve the problems beyond the capability of cryptographic security [23, 49, 48, 50, 70, 80, 90, 92]. In addition, there are many attacks designed to exploit the unreliable communication channels and unattended operation of wireless sensor networks. Furthermore, due to the inherent unattended feature of wireless sensor networks, we argue that physical attacks to sensors play an important role in the operation of wireless sensor networks. Thus, we include a detailed discussion of the physical attacks and their corresponding defenses [3, 4, 30, 34, 43, 71, 74, 84, 85, 88], topics typically ignored in most of the current research on sensor security.

We classify the main aspects of wireless sensor network security into four major categories: *the obstacles to sensor network security*, *the requirements of a secure wireless sensor network*, *attacks*, and *defensive measures*. The organization then follows this classification. For the completeness of the chapter, we also give a brief introduction of related security techniques, while providing appropriate citations for those interested in a more detailed discussion of a particular topic.

The remainder of this chapter is organized as follows. In Section 3, we summarize the obstacles for the sensor network security. The security requirements of a wireless sensor network are listed in Section 4. The major attacks in sensor network are categorized in Section 5, and we outline the corresponding defensive measures in Section 6. Finally, we conclude the chapter in Section 7.

3 Obstacles of Sensor Security

A wireless sensor network is a special network which has many constraints compared to a traditional computer network. Due to these constraints it is difficult to directly employ the existing security approaches to the area of wireless sensor networks. Therefore, to develop useful security mechanisms while borrowing the ideas from the current security techniques, it is necessary to know and understand these constraints first [10].

3.1 Very Limited Resources

All security approaches require a certain amount of resources for the implementation, including data memory, code space, and energy to power the sensor. However, currently these resources are very limited in a tiny wireless sensor.

- **Limited Memory and Storage Space** A sensor is a tiny device with only a small amount of memory and storage space for the code. In order to build an effective security mechanism, it is necessary to limit the code size of the security algorithm. For example, one common sensor type (TelosB) has an 16-bit, 8 MHz RISC CPU with only 10K RAM, 48K program memory, and 1024K flash storage [14]. With such a limitation, the software built for the sensor must also be quite small. The total code space of TinyOS, the de-facto standard operating system for wireless sensors, is approximately 4K [32], and the core scheduler occupies only 178 bytes. Therefore, the code size for the all security related code must also be small.

- **Power Limitation** Energy is the biggest constraint to wireless sensor capabilities. We assume that once sensor nodes are deployed in a sensor network, they cannot be easily replaced (high operating cost) or recharged (high cost of sensors). Therefore, the battery charge taken with them to the field must be conserved to extend the life of the individual sensor node and the entire sensor network. When implementing a cryptographic function or protocol within a sensor node, the energy impact of the added security code must be considered. When adding security to a sensor node, we are interested in the impact that security has on the lifespan of a sensor (i.e., its battery life). The extra power consumed by sensor nodes due to security is related to the processing required for security functions (e.g., encryption, decryption, signing data, verifying signatures), the energy required to transmit the security related data or overhead (e.g., initialization vectors needed for encryption/decryption), and the energy required to store security parameters in a secure manner (e.g., cryptographic key storage).

3.2 Unreliable Communication

Certainly, unreliable communication is another threat to sensor security. The security of the network relies heavily on a defined protocol, which in turn depends on communication.

- **Unreliable Transfer** Normally the packet-based routing of the sensor network is connectionless and thus inherently unreliable. Packets may get damaged due to channel errors or dropped at highly congested nodes. The result is lost or missing packets. Furthermore, the unreliable wireless communication channel also results in damaged packets. Higher channel error rate also forces the software developer to devote resources to error handling. More importantly, if the protocol lacks the appropriate error handling it is possible to lose critical security packets. This may include, for example, a cryptographic key.

- **Conflicts** Even if the channel is reliable, the communication may still be unreliable. This is due to the broadcast nature of the wireless sensor network. If packets meet in the middle of transfer, conflicts will occur and the transfer itself will fail. In a crowded (high density) sensor network, this can be a major problem. More details about the effect of wireless communication can be found at [1].

- **Latency** The multi-hop routing, network congestion, and node processing can lead to greater latency in the network, thus making it difficult to achieve synchronization among sensor nodes. The synchronization issues can be critical to sensor security where the security mechanism relies on critical event reports and cryptographic key distribution. Interested readers please refer to [78] on real-time communications in wireless sensor networks.

3.3 Unattended Operation

Depending on the function of the particular sensor network, the sensor nodes may be left unattended for long periods of time. There are three main caveats to unattended sensor nodes:

- **Exposure to Physical Attacks** The sensor may be deployed in an environment open to adversaries, bad weather, and so on. The likelihood that a sensor suffers a physical attack in such an environment is therefore much higher than the typical PCs, which is located in a secure place and mainly faces attacks from a network.

- **Managed Remotely** Remote management of a sensor network makes it virtually impossible to detect physical tampering (i.e., through tamperproof seals) and physical maintenance issues (e.g., battery replacement). Perhaps the most extreme example of this is a sensor node used for remote reconnaissance missions behind enemy lines. In such a case, the node may not have any physical contact with friendly forces once deployed.

- **No Central Management Point** A sensor network should be a distributed network without a central management point. This will increase the vitality of the sensor network. However, if designed incorrectly, it will make the network organization difficult, inefficient, and fragile.

Perhaps most importantly, the longer that a sensor is left unattended the more likely that an adversary has compromised the node.

4 Security Requirements

A sensor network is a special type of network. It shares some commonalities with a typical computer network, but also poses unique requirements of its own as discussed in Section 3. Therefore, we can think of the requirements of a wireless sensor network as encompassing both the typical network requirements and the unique requirements suited solely to wireless sensor networks.

4.1 Data Confidentiality

Data confidentiality is the most important issue in network security. Every network with any security focus will typically address this problem first. In sensor networks, the confidentiality relates to the following [10, 65]:

- A sensor network should not leak sensor readings to its neighbors. Especially in a military application, the data stored in the sensor node may be highly sensitive.

- In many applications nodes communicate highly sensitive data, e.g., key distribution, therefore it is extremely important to build a secure channel in a wireless sensor network.

- Public sensor information, such as sensor identities and public keys, should also be encrypted to some extent to protect against traffic analysis attacks.

The standard approach for keeping sensitive data secret is to encrypt the data with a secret key that only intended receivers possess, thus achieving confidentiality.

4.2 Data Integrity

With the implementation of confidentiality, an adversary may be unable to steal information. However, this doesn't mean the data is safe. The adversary can change the data, so as to send the sensor network into disarray. For example, a malicious node may add some fragments or manipulate the data within a packet. This new packet can then be sent to the original receiver. Data loss or damage can even occur without the presence of a malicious node due to the harsh communication environment. Thus, data integrity ensures that any received data has not been altered in transit.

4.3 Data Freshness

Even if confidentiality and data integrity are assured, we also need to ensure the freshness of each message. Informally, data freshness suggests that the data is recent, and it ensures that no old messages have been replayed. This requirement is especially important when there are shared-key strategies employed in the design. Typically shared keys need to be changed over time. However, it takes time for new shared keys to be propagated to the entire network. In this case, it is easy for the adversary to use a replay attack. Also, it is easy to disrupt the normal work of the sensor, if the sensor is unaware of the new key change time. To solve this problem a nonce, or another time-related counter, can be added into the packet to ensure data freshness.

4.4 Availability

Adjusting the traditional encryption algorithms to fit within the wireless sensor network is not free, and will introduce some extra costs. Some approaches choose to modify the code to reuse as much code as possible. Some approaches try to make use of additional communication to achieve the same goal. What's more, some approaches force strict limitations on the data access, or propose an unsuitable scheme (such as a central point scheme) in order to simplify the algorithm. But all these approaches weaken the availability of a sensor and sensor network for the following reasons:

- Additional computation consumes additional energy. If no more energy exists, the data will no longer be available.

- Additional communication also consumes more energy. What's more, as communication increases so too does the chance of incurring a communication conflict.

- A single point failure will be introduced if using the central point scheme. This greatly threatens the availability of the network.

The requirement of security not only affects the operation of the network, but also is highly important in maintaining the availability of the whole network.

4.5 Self-Organization

A wireless sensor network is a typically an ad hoc network, which requires every sensor node be independent and flexible enough to be self-organizing and self-healing according to different situations. There is no fixed infrastructure available for the purpose of network management in a sensor network. This inherent feature brings a great challenge to wireless sensor network security as well. For example, the dynamics of the whole network inhibits the idea of pre-installation of a shared key between the base station and all sensors [21]. Several random key predistribution schemes have been proposed in the context of symmetric encryption techniques [13, 21, 37, 53]. In the context of applying public-key cryptography techniques in sensor networks, an efficient mechanism for public-key distribution is necessary as well. In the same way that distributed sensor networks must self-organize to support multihop routing, they must also self-organize to conduct key management and building trust relation among sensors. If self-organization is lacking in a sensor network, the damage resulting from an attack or even the hazardous environment may be devastating.

4.6 Time Synchronization

Most sensor network applications rely on some form of time synchronization. In order to conserve power, an individual sensor's radio may be turned off for periods of time. Furthermore, sensors may wish to compute the end-to-end delay of a packet as it travels between two pairwise sensors. A more collaborative sensor network may require group synchronization for tracking applications, etc. In [24], the authors propose a set of secure synchronization protocols for sender-receiver (pairwise), multihop sender-receiver (for use when the pair of nodes are not within single-hop range), and group synchronization.

4.7 Secure Localization

Often, the utility of a sensor network will rely on its ability to accurately and automatically locate each sensor in the network. A sensor network designed to locate faults will need accurate location information in order to pinpoint the location of a fault. Unfortunately, an attacker can easily manipulate non-secured location information by reporting false signal strengths, replaying signals, etc.

A technique called verifiable multilateration (VM) is described in [81]. In multilateration, a device's position is accurately computed from a series of known reference points. In [81], authenticated ranging and distance bounding are used to ensure accurate location of a node. Because of distance bounding, an attacking node can only increase its claimed distance from a reference point. However, to ensure location consistency, an attacking node would also have to prove that its distance from another reference point is shorter [81]. Since it cannot do this, a node manipulating the localization protocol can be found. For large sensor networks, the SPINE (Secure Positioning for sensor NEtworks) algorithm is used. It is a three phase algorithm based upon verifiable multilateration [81].

In [47], SeRLoc (Secure Range-Independent Localization) is described. Its novelty is its decentralized, range-independent nature. SeRLoc uses locators that transmit beacon information. It is assumed that the locators are trusted and cannot be compromised. Furthermore, each locator is assumed to know its own location. A sensor computes its location by listening for the beacon information sent by each locator. The beacons include the locator's location. Using all of the beacons that a sensor node detects, a node computes an approximate location based on the coordinates of the locators. Using a majority vote scheme, the sensor then computes an overlapping antenna region. The final computed location is the "center of gravity" of the overlapping antenna region [47]. All beacons transmitted by the locators are encrypted with a shared global symmetric key that is pre-loaded to the sensor prior to deployment. Each sensor also shares a unique symmetric key with each locator. This key is also pre-loaded on each sensor.

4.8 Authentication

An adversary is not just limited to modifying the data packet. It can change the whole packet stream by injecting additional packets. So the receiver needs to ensure that the data used in any decision-making process originates from the correct source. On the other hand, when constructing the sensor network, authentication is necessary for many administrative tasks (e.g. network reprogramming or controlling sensor node duty cycle). From the above, we can see that message authentication is important for many applications in sensor networks. Informally, data authentication allows a receiver to verify that the data really is sent by the claimed sender. In the case of two-party communication, data authentication can be achieved through a purely symmetric mechanism: the sender and the receiver share a secret key to compute the message authentication code (MAC) of all communicated data.

Adrian Perrig *et al.* propose a key-chain distribution system for their μTESLA secure broadcast protocol [65]. The basic idea of the μTESLA system is to achieve asymmetric cryptography by delaying the disclosure of the symmetric keys. In this case a sender will broadcast a message generated with a secret key. After a certain period of time, the sender will disclose the secret key. The receiver is responsible for buffering the packet until the secret key has been disclosed. After disclosure the receiver can authenticate the packet, provided that the packet was received before the key was disclosed.

One limitation of μTESLA is that some initial information must be unicast to each sensor node before authentication of broadcast messages can begin.

Liu and Ning [51, 52] propose an enhancement to the μTESLA system that uses broadcasting of the key chain commitments rather than μTESLA's unicasting technique. They present a series of schemes starting with a simple pre-determination of key chains and finally settling on a multi-level key chain technique. The multi-level key chain scheme uses pre-determination and broadcasting to achieve a scalable key distribution technique that is designed to be resistant to denial of service attacks, including jamming.

5 Attacks

Sensor networks are particularly vulnerable to several key types of attacks. Attacks can be performed in a variety of ways, most notably as denial of service attacks, but also through traffic analysis, privacy violation, physical attacks, and so on. Denial of service attacks on wireless sensor networks can range from simply jamming the sensor's communication channel to more sophisticated attacks designed to violate the 802.11 MAC protocol [64] or any other layer of the wireless sensor network.

Due to the potential asymmetry in power and computational constraints, guarding against a well orchestrated denial of service attack on a wireless sensor network can be nearly impossible. A more powerful node can easily jam a sensor node and effectively prevent the sensor network from performing its intended duty.

We note that attacks on wireless sensor networks are not limited to simply denial of service attacks, but rather encompass a variety of techniques including node takeovers, attacks on the routing protocols, and attacks on a node's physical security. In this section, we first address some common denial of service attacks and then describe additional attacking, including those on the routing protocols as well as an identity based attack known as the Sybil attack.

5.1 Background

Wood and Stankovic define one kind of denial of service attack as "any event that diminishes or eliminates a network's capacity to perform its expected function" [88]. Certainly, denial of service attacks are not a new phenomenon. In fact, there are several standard techniques used in traditional computing to cope with some of the more common denial of service techniques, although this is still an open problem to the network security community. Unfortunately, wireless sensor networks cannot afford the computational overhead necessary in implementing many of the typical defensive strategies.

What makes the prospect of denial of service attacks even more alarming is the projected use of sensor networks in highly critical and sensitive applications. For example, a sensor network designed to alert building occupants in the event of a fire could be highly susceptible to a denial of service attack. Even worse, such an attack could result in the deaths of building occupants due to the non-operational fire detection network.

Other possible uses for wireless sensors include the monitoring of traffic flows which may include the control of traffic lights, and so forth. A denial of service attack on such a sensor network could prove very costly, especially on major roads.

For this reason, researchers have spent a great deal of time both identifying the various types of denial of service attacks and devising strategies to subvert such attacks. We describe now some of the major types of denial of service attacks.

5.2 Types of Denial of Service attacks

A standard attack on wireless sensor networks is simply to jam a node or set of nodes. Jamming, in this case, is simply the transmission of a radio signal that interferes with the radio frequencies being used by the sensor network [88]. The jamming of a network can come in two forms: constant jamming, and intermittent jamming. Constant jamming involves the complete jamming of the entire network. No messages are able to be sent or received. If the jamming is only intermittent, then nodes are able to exchange messages periodically, but not consistently. This too can have a detrimental impact on the sensor network as the messages being exchanged between nodes may be time sensitive [88].

Attacks can also be made on the link layer itself. One possibility is that an attacker may simply intentionally violate the communication protocol, e.g., ZigBee [94] or IEEE 801.11b (Wi-Fi) protocol, and continually transmit messages in an attempt to generate collisions. Such collisions would require the retransmission of any packet affected by the collision. Using this technique it would be possible for an attacker to simply deplete a sensor node's power supply by forcing too many retransmissions.

At the routing layer, a node may take advantage of a multihop network by simply refusing to route messages. This could be done intermittently or constantly with the net result being that any neighbor who routes through the malicious node will be unable to exchange messages with, at least, part of the network. Extensions to this technique including intentionally routing messages to incorrect nodes (misdirection) [88].

The transport layer is also susceptible to attack, as in the case of flooding. Flooding can be as simple as sending many connection requests to a susceptible node. In this case, resources must be allocated to handle the connection request. Eventually a node's resources will be exhausted, thus rendering the node useless.

5.3 The Sybil attack

Newsome *et al.* describe the Sybil attack as it relates to wireless sensor networks [59]. Simply put, the Sybil attack is defined as a "malicious device illegitimately taking on multiple identities"[59]. It was originally described as an attack able to defeat the redundancy mechanisms of distributed data storage systems in peer-to-peer networks [18]. In addition to defeating distributed data storage systems, the Sybil attack is also effective against routing algorithms, data aggregation, voting, fair resource allocation and foiling misbehavior detection. Regardless of the target (voting, routing, aggregation), the Sybil algorithm functions similarly. All of the techniques involve utilizing multiple identities. For instance, in a sensor network voting scheme, the Sybil attack might utilize multiple identities to generate additional "votes." Similarly, to attack the routing protocol, the Sybil attack would rely on a malicious node taking on the identity of multiple nodes, and thus routing multiple paths through a single malicious node.

5.4 Traffic Analysis Attacks

Wireless sensor networks are typically composed of many low-power sensors communicating with a few relatively robust and powerful base stations. It is not unusual, therefore, for data to be gathered by the individual nodes where it is ultimately routed to the base station. Often, for an adversary to effectively render the network useless, the attacker can simply disable the base station. To make matters worse, Deng *et al.* demonstrate two attacks that can identify the base station in a network (with high probability) without even understanding the contents of the packets (if the packets are themselves encrypted) [16].

A rate monitoring attack simply makes use of the idea that nodes closest to the base station tend to forward more packets than those farther away from the base station. An attacker need only monitor which nodes are sending packets and follow those nodes that are sending the most packets. In a time correlation attack, an adversary simply generates events and monitors to whom a node sends its packets. To generate an event, the adversary could simply generate a physical event that would be monitored by the sensor(s) in the area (turning on a light, for instance) [16].

5.5 Node Replication Attacks

Conceptually, a node replication attack is quite simple: an attacker seeks to add a node to an existing sensor network by copying (replicating) the node ID of an existing sensor node [63]. A node replicated in this fashion can severely disrupt a sensor network's performance: packets can be corrupted or even misrouted. This can result in a disconnected network, false sensor readings, etc. If an attacker can gain physical access to the entire network he can copy cryptographic keys to the replicated sensor and can also insert the replicated node into strategic points in the network [63]. By inserting the replicated nodes at specific network points, the attacker could easily manipulate a specific segment of the network, perhaps by disconnecting it altogether.

5.6 Attacks Against Privacy

Sensor network technology promises a vast increase in automatic data collection capabilities through efficient deployment of tiny sensor devices. While these technologies offer great benefits to users, they also exhibit significant potential for abuse. Particularly relevant concerns are privacy problems, since sensor networks provide increased data collection capabilities [28]. Adversaries can use even seemingly innocuous data to derive sensitive information if they know how to correlate multiple sensor inputs. For example, in the famous "panda-hunter problem" [61], the hunter can imply the position of pandas by monitoring the traffic.

The main privacy problem, however, is not that sensor networks enable the collection of information. In fact, much information from sensor networks could probably be collected through direct site surveillance. Rather, sensor networks aggravate the privacy problem because they make large volumes of information easily available through remote access. Hence, adversaries need not be physically present to maintain surveillance. They can gather information in a low-risk, anonymous manner. Remote access also allows a single adversary to monitor multiple sites simultaneously [11]. Some of the more common attacks [28, 11] against sensor privacy are:

- **Monitor and Eavesdropping** This is the most obvious attack to privacy. By listening to the data, the adversary could easily discover the communication contents. When the traffic conveys the control information about the sensor network configuration, which contains potentially more detailed information than accessible through the location server, the eavesdropping can act effectively against the privacy protection.

- **Traffic Analysis** Traffic analysis typically combines with monitoring and eavesdropping. An increase in the number of transmitted packets between certain nodes could signal that a specific sensor has registered activity. Through the analysis on the traffic, some sensors with special roles or activities can be effectively identified.

- **Camouflage** Adversaries can insert their node or compromise the nodes to hide in the sensor network. After that these nodes can masquerade as a normal node to attract the packets, then misroute the packets, e.g. forward the packets to the nodes conducting the privacy analysis.

It is worth noting that, as pointed out in [64], the current understanding of privacy in wireless sensor networks is immature, and more research is needed.

5.7 Physical Attacks

Sensor networks typically operate in hostile outdoor environments. In such environments, the small form factor of the sensors, coupled with the unattended and distributed nature of their deployment make them highly susceptible to physical attacks, i.e., threats due to physical node destructions [86]. Unlike many other attacks mentioned above, physical attacks destroy sensors permanently, so the losses are irreversible. For instance, attackers can extract cryptographic secrets, tamper with the associated circuitry, modify programming in the sensors, or replace them with malicious sensors under the control of the attacker [85]. Recent work has shown that standard sensor nodes, such as the MICA2 motes, can be compromised in less than one minute [30]. While these results are not surprising given that the MICA2 lacks tamper resistant hardware protection, they provide a cautionary note about the speed of a well-trained attacker. If an adversary compromises a sensor node, then the code inside the physical node may be modified.

6 Defensive Measures

Now we are in a position to describe the measures for satisfying security requirements, and protecting the sensor network from attacks. We start with *key establishment in wireless sensor networks*, which lays the foundation for the security in a wireless sensor network, followed by *defending against DoS attacks*, *secure broadcasting and multicasting*, *defending against attacks on routing protocols*, *combating traffic analysis attacks*, *defending against attacks on sensor privacy*, *intrusion detection*, *secure data aggregation*, *defending against physical attacks*, and *trust management.*

6.1 Key Establishment

One security aspect that receives a great deal of attention in wireless sensor networks is the area of key management. Wireless sensor networks are unique (among other embedded wireless networks) in this aspect due to their size, mobility and computational/power constraints. Indeed, researchers envision wireless sensor networks to be orders of magnitude larger than their traditional embedded counterparts. This, coupled with the operational constraints described previously, makes secure key management an absolute necessity in most wireless sensor network designs. Because encryption and key management/establishment are so crucial to the defense of a wireless sensor network, with nearly all aspects of wireless sensor network defenses relying on solid encryption, we first begin with an overview of the unique key and encryption issues surrounding wireless sensor networks before discussing more specific sensor network defenses.

6.1.1 Background

Key management issues in wireless networks are not unique to wireless sensor networks. Indeed, key establishment and management issues have been studied in depth outside of the wireless networking arena. Traditionally, key establishment is done using one of many public-key protocols. One of the more common is the Diffie-Hellman public key protocol, but there are many others.

Most of the traditional techniques, however, are unsuitable in low power devices such as wireless sensor networks. This is due largely to the fact that typical key exchange techniques use asymmetric cryptography, also called public key cryptography. In this case, it is necessary to maintain two mathematically related keys, one of which is made public while the other is kept private. This allows data to be encrypted with the public key and decrypted only with the private key. The problem with asymmetric cryptography, in a wireless sensor network, is that it is typically too computationally intensive for the individual nodes in a sensor network. This is true in the general case, however, [25, 29, 55, 87] show that it is feasible with the right selection of algorithms.

Symmetric cryptography is therefore the typical choice for applications that cannot afford the computational complexity of asymmetric cryptography. Symmetric schemes utilize a single shared key known only between the two communicating hosts. This shared key is used for both encrypting and decrypting data. The traditional example of symmetric cryptography is DES (Data Encryption Standard). The use of DES, however, is quite limited due to the fact that it can be broken relatively easily. In light of the shortcomings of DES, other symmetric cryptography systems have been proposed including 3DES (Triple DES), RC5, AES, and so on [73].

An analysis of the various ciphers is presented in [44] with a summary of their results shown in Table 1. The table ranks ciphers in several ways - key setup efficiency, encryption/decryption efficiency, data memory, and code memory (seeTable 1). Further, each cipher is both size optimized and speed optimized. Skipjack is the overall winner, where either the size optimized or speed optimized proves best in every category. While RC5 and RC6 rank fairly high in code size, their poor performance in both encryption/decryption efficiency and key setup lower their rankings significantly. Both MISTY1 and KASUMI perform average in all categories except for code memory [44].

One major shortcoming of symmetric cryptography is the key exchange problem. Simply put, the key exchange problem derives from the fact that two communicating hosts must somehow know the shared key before they can communicate securely. So the problem that arises is how to ensure that the shared key is indeed shared between the two hosts who wish to communicate and no other rogue hosts who may wish to eavesdrop. How to distribute a shared key securely to communicating hosts is a non-trivial

Code memory		Data memory		En/decryption efficiency		Key setup efficiency	
CBC	OFB	CBC	OFB	CBC	OFB	Encryption	Decryption
Skipjack$_z$	Skipjack$_z$	Skipjack$_z$	Skipjack$_z$	Skipjack$_s$	Skipjack$_s$	Skipjack$_z$	Skipjack$_z$
RC5$_z$	Skipjack$_s$	MISTY1$_z$	MISTY1$_z$	Rijndael$_s$	Rijndael$_s$	Skipjack$_s$	Skipjack$_s$
Skipjack$_s$	RC5$_z$	Skipjack$_s$	Skipjack$_s$	Twofish$_s$	Twofish$_s$	MISTY1$_s$	MISTY1$_s$
RC6$_z$	RC6$_z$	MISTY1$_s$	MISTY1$_s$	MISTY1$_s$	MISTY1$_s$	MISTY1$_z$	MISTY1$_z$
RC6$_s$	RC6$_s$	KASUMI$_z$	KASUMI$_s$	Camellia$_s$	Camellia$_s$	Rijndael$_s$	KASUMI$_s$
RC5$_s$	Rijndael$_z$	KASUMI$_s$	KASUMI$_z$	MISTY1$_z$	Rijndael$_z$	KASUMI$_s$	KASUMI$_z$
MISTY1$_z$	RC5$_s$	RC5	RC5	KASUMI$_s$	MISTY1$_z$	Rijndael$_z$	Rijndael$_s$
MISTY1$_s$	MISTY1$_z$	Twofish$_z$	Twofish$_s$	Rijndael$_z$	Skipjack$_z$	KASUMI$_z$	Twofish$_s$
Rijndael$_z$	Rijndael$_s$	Twofish$_s$	Twofish$_z$	Skipjack$_z$	KASUMI$_s$	Twofish$_s$	Rijndael$_z$
Twofish$_z$	MISTY1$_s$	RC6$_z$	Rijndael	KASUMI$_z$	KASUMI$_z$	Twofish$_z$	Twofish$_z$
KASUMI$_z$	KASUMI$_z$	Rijndael$_z$	RC6	Twofish$_z$	Twofish$_z$	Camellia$_s$	Camellia$_s$
Rijndael$_s$	KASUMI$_s$	RC6$_s$	Camellia$_z$	Camellia$_z$	Camellia$_z$	Camellia$_z$	Camellia$_z$
KASUMI$_s$	Twofish$_z$	Rijndael$_s$	Camellia$_s$	RC6$_s$	RC6$_s$	RC5$_s$	RC5$_s$
Twofish$_s$	Twofish$_s$	Camellia$_z$		RC5$_s$	RC5$_s$	RC6$_s$	RC6$_s$
Camellia$_z$	Camellia$_z$	Camellia$_s$		RC6$_z$	RC6$_z$	RC5$_z$	RC5$_z$
Camellia$_s$	Camellia$_s$			RC5$_z$	RC5$_z$	RC6$_z$	RC6$_z$

Table 1: A summary of cipher performance from [44]. Subscript z means size-optimised, s means speed-optimised.

problem since pre-distributing the keys is not always feasible.

6.1.2 Key Establishment and Associated Protocols

Random key pre-distribution schemes have several variants [13, 21, 37, 53]. Eschenauer and Gligor propose a key pre-distribution scheme [21] that relies on probabilistic key sharing among nodes within the sensor network. Their system works by distributing a key ring to each participating node in the sensor network before deployment. Each key ring should consist of a number randomly chosen keys from a much larger pool of keys generated offline. An enhancement to this technique utilizing multiple keys is described in [13]. Further enhancements are proposed in [19, 53] with additional analysis and enhancements provided by [37].

Using this technique, it is not necessary that each pair of nodes share a key. However, any two nodes that do share a key may use the shared key to establish a direct link to one another. Eschenauer and Gligor show that, while not perfect, it is probabilistically likely that large sensor networks will enjoy shared-key connectivity. Further, they demonstrate that such a technique can be extended to key revocation, re-keying, and the addition/deletion of nodes.

The LEAP protocol described by Zhu *et al.* [93] takes an approach that utilizes multiple keying mechanisms. Their observation is that no single security requirement accurately suites all types of communication in a wireless sensor network. Therefore, four different keys are used depending on whom the sensor node is communicating with. Sensors are preloaded with an initial key from which further keys can be established. As a security precaution, the initial key can be deleted after its use in order to ensure that a compromised sensor cannot add additional compromised nodes to the network.

In PIKE [12], Chan and Perrig describe a mechanism for establishing a key between two sensor nodes that is based on the common trust of a third node somewhere within the sensor network. The nodes and their shared keys are spread over the network such that for any two nodes A and B, there is a node C that shares a key with both A and B. Therefore, the key establishment protocol between A and B can be securely routed through C.

Huang *et al.* [36] propose a hybrid key establishment scheme that makes use of the difference in computational and energy constraints between a sensor node and the base station. They posit that an individual sensor node possesses far less computational power and energy than a base station. In light of this, they propose placing the major cryptographic burden on the base station where the resources tend to be greater. On the sensor side, symmetric-key operations are used in place of their asymmetric alternatives. The sensor and the base station authenticate based on elliptic curve cryptography. Elliptic curve cryptography is often used in sensors due to the fact that relatively small key lengths are required to achieve a given level of security.

Huang *et al.* also use certificates to establish the legitimacy of a public key. The certificates are based on an elliptic curve implicit certificate scheme [36]. Such certificates are useful to ensure both that the key belongs to a device and that the device is a legitimate member of the sensor network. Each node obtains a certificate before joining the network using an out-of-band interface.

6.1.3 Public Key Cryptography

Two of the major techniques used to implement public-key cryptosystems are RSA and elliptic curve cryptography (ECC) [73]. Traditionally, these have been thought to be far too heavyweight for use in wireless sensor networks. Recently, however, several groups have successfully implemented public-key cryptography (to varying degrees) in wireless sensor networks.

In [29] Gura *et al.* report that both RSA and elliptic curve cryptography are possible using 8-bit CPUs with ECC, demonstrating a performance advantage over RSA. Another advantage is that ECC's 160 bit keys result in shorter messages during transmission compared the 1024 bit RSA keys. In particular Gura *et al.* demonstrate that the point multiplication operations in ECC are an order of magnitude faster than private-key operations within RSA, and are comparable (though somewhat slower) to the RSA public-key operation [29].

In [87], Watro *et al.* show that portions of the RSA cryptosystem can be successfully applied to actual wireless sensors, specifically the UC Berkeley MICA2 motes [32]. In particular, they implemented the public operations on the sensors themselves while offloading the private operations to devices better suited for the larger computational tasks. In this case, a laptop was used.

The TinyPK system described by [87] is designed specifically to allow authentication and key agreement between resource constrained sensors. The agreed upon keys may then be used in conjunction with the existing cryptosystem, *TinySec* [39]. To do this, they implement the Diffie-Hellman key exchange algorithm and perform the public-key operations on the Berkeley motes.

The Diffie-Hellman key exchange algorithm used in [55] is depicted in Figure 1. In this case, a point G is selected from an elliptic curve E, both of which are public. A random integer K_A is selected, which will act as the private key. The public key (T_A in the case of Alice from Figure 1) is then $T_A = K_A * G$. Bob performs a similar set of operations to compute $T_B = K_B * G$. Alice and Bob can now easily compute the shared-secret using their own private keys and the public keys that have been exchanged. In this case, Alice computes $K_A * T_B = K_A * K_B * G$ while Bob computes $K_B * T_A = K_B * K_A * G$. Because $K_A * T_B = K_B * T_A$, Alice and Bob now share a secret key.

As stated above, the elliptic curve cryptography shows promise over that of RSA due to its efficiency compared to the private-key operations of RSA. Further, using ECC, the key length required to securely transmit TinySec keys can be as small as 163 bits rather than the 1024 bits required in RSA. In [55], Malan *et al.* demonstrate a working implementation of Diffie-Hellman based on the Elliptic Curve Discrete Logarithm Problem (Figure 1). And while key generation is by no means fast or inexpensive (34.161 seconds to generate a public/private-key pair and 34.173 seconds to generate a

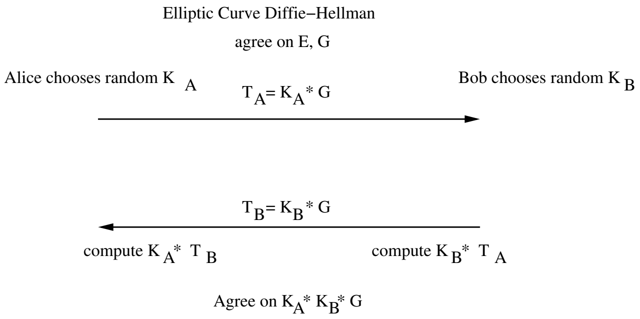

Figure 1: The Diffie-Hellman Elliptic Curve Key Exchange Algorithm [55].

Network Layer	Attacks	Defenses
Physical	Jamming	Spread-spectrum, priority messages, lower duty cycle, region mapping, mode change
	Tampering	Tamper-proof, hiding
Link	Collision	Error correcting code
	Exhaustion	Rate limitation
	Unfairness	Small frames
Network and routing	Neglect and greed	Redundancy, probing
	Homing	Encryption
	Misdirection	Egress filtering, authorization monitoring
	Black holes	Authorization, monitoring, redundancy
Transport	Flooding	Client Puzzles
	Desynchronization	Authentication

Table 2: Sensor network layers and DoS attacks/defenses [88].

shared secret with Diffie-Hellman [55]), it is sufficient for infrequent use in generating keys in the TinySec protocols.

6.2 Defending Against DoS Attacks

In Table 2 the most common layers of a typical wireless sensor network are summarized along with their attacks and defenses. Since denial of service attacks are so common (see Section 5), effective defenses must be available to combat them. One strategy in defending against the classic jamming attack is to identify the jammed part of the sensor network and effectively route around the unavailable portion. Wood and Stankovic [88] describe a two phase approach where the nodes along the perimeter of the jammed region report their status to their neighbors who then collaboratively define the jammed region and simply route around it.

To handle jamming at the MAC layer, nodes might utilize a MAC admission control that is rate limiting. This would allow the network to ignore those requests designed to exhaust the power reserves of a node. This, however, is not fool-proof as the network must be able to handle any legitimately large traffic volumes.

Overcoming rogue sensors that intentionally misroute messages can be done at the cost of redundancy. In this case, a sending node can send the message along multiple paths in an effort to increase the likelihood that the message will ultimately arrive at its destination. This has the advantage of effectively dealing with nodes that may not be malicious, but rather may have simply failed as it does not rely on a single node to route its messages.

To overcome the transport layer flooding denial of service attack Aura, Nikander and Leiwo suggest using the client puzzles posed by Juels and Brainard [5] in an effort to discern a node's commitment to making the connection by utilizing some of their own resources. Aura *et al.* advocate that a server should force a client to commit its own resources first. Further, they suggest that a server should always force a client to commit more resources up front than the server. This strategy would likely be effective as long as the client has computational resources comparable to those of the server.

6.3 Secure Broadcasting and Multicasting

The wireless sensor network research community has progressively reached a consensus that the major communication pattern of wireless sensor networks is broadcasting and multicasting, e.g., 1-to-N, N-to-1, and M-to-N, instead of the traditional point-to-point communication on the Internet. Next we examine the current state of research in secure broadcasting and multicasting. As we will see, in wireless sensor networks, a great deal of the security derives from ensuring that only members of the broadcast or multicast group possess the required keys in order to decrypt the broadcast or multicast messages. Because of this, most of the work presented in 6.1 is still applicable. Here, however, we will address those schemes that have been specifically designed to support broadcasting and multicasting in wireless sensor networks.

6.3.1 Traditional Broadcasting and Multicasting

Traditionally, multicasting and broadcasting techniques have been used to reduce the communication and management overhead of sending a single message to multiple receivers. In order to ensure that only certain users receive the multicast or broadcast, encryption techniques must be employed. In both a wired and wireless network this is done using cryptography. The problem then is one of key management. To handle this, several key management schemes have been devised: centralized group key management protocols, decentralized management protocols, and distributed management protocols [69].

In the case of the centralized group key management protocols, a central authority is used to maintain the group. Decentralized management protocols, however, divide the task of group management amongst multiple nodes. Each node that is responsible for part of the group management is responsible for a certain subset of the nodes in the network. In the last case, distributed key management protocols, there is no single key management authority. Therefore, the entire group of nodes are responsible for key management [69].

In order to efficiently distribute keys, one well known technique is to use a logical key tree. Such a technique falls into the centralized group key management protocols. This technique has been extended to wireless sensor networks in [66, 46, 45]. While centralized solutions are often not ideal, in the case of wireless sensor networks a centralized solution offers some utility. Such a technique allows a more powerful base station to offload some of the computations from the less powerful sensor nodes.

6.3.2 Secure Multicasting

Di Pietro *et al.* describe a directed diffusion based multicast technique for use in wireless sensor networks that also takes advantage of a logical key hierarchy [66]. In a standard logical key hierarchy a central key distribution center is responsible for disbursing the keys throughout the network. The key distribution center, therefore, is the root of the key hierarchy while individual nodes make up the leaves. The internal nodes of the key hierarchy contain keys that are used in the re-keying process [66].

Directed diffusion is a data-centric, energy efficient dissemination technique that has been designed for use in wireless sensor networks [38]. In directed diffusion, a query is transformed into an interest (due to the data-centric nature of the network). The interest is then diffused throughout the network and the network begins collecting data based on that interest. The dissemination technique also sets up certain gradients designed to draw events toward the interest. Data collected as a result of the interest can then be sent back along the reverse path of the interest propagation [38].

Using the above mentioned directed diffusion technique, Di Pietro *et al.* enhance the logical key hierarchy to create a directed diffusion based logical key hierarchy. The logical key hierarchy technique provides mechanisms for nodes joining and leaving groups where the key hierarchy is used to effectively re-key all nodes within the leaving node's hierarchy [66]. The directed diffusion is also used in node joining and leaving. When a node declares an intent to join, for example, a join "interest" is generated which travels down the gradient of "interest about interest to join" [66]. When a node joins, a key set is generated for the new node based on keys within the key hierarchy.

Kaya *et al.* discuss the problem of multicast group management in [42]. In this case, nodes are grouped based on locality and attach to a security tree. However, they assume that nodes within the mobile network are somewhat more powerful than a traditional sensor in a wireless sensor network.

6.3.3 Secure Broadcasting

Lazos and Poovendran describe a tree based key distribution scheme that is similar to [66]. They suggest a routing-aware based tree where the leaf nodes are assigned keys based on all relay nodes above them. They argue that their technique, which takes advantage of routing information, is more energy efficient than routing schemes that arbitrarily arrange nodes into the routing tree. They propose a greedy routing-aware key distribution algorithm [45].

In [46], Lazos and Poovendran use a similar technique to [45], but instead use geographic location information (e.g., GPS) rather than routing information. In this case, however, nodes (with the help of the geographic location system) are grouped into clusters with the observation that nodes within a cluster will be able to reach one another with a single broadcast. Using the cluster information, a key hierarchy is constructed as in [45].

6.4 Defending Against Attacks on Routing Protocols

Routing in wireless sensor networks has, to some extent, been reasonably well studied. However, most current research has focused primarily on providing the most energy efficient routing. There is a great need for both secure and energy efficient routing protocols in wireless sensor networks as attacks such as the sinkhole, wormhole and Sybil attacks demonstrate [35, 40, 59]. As wireless sensor networks continue to grow in size and utility, routing security must not be an after-thought, but rather they must be included as part of the overall sensor network design. This section describes the current state of routing security as it applies to wireless sensor networks.

6.4.1 Background

Because wireless sensors are designed to be widely distributed power and computationally constrained networks, efficient routing protocols must be used in order to maximize the battery life of each node. There are a variety of routing protocols in use in wireless sensor networks, so it is not possible to provide a single security protocol that will be able to secure each type of routing protocol. Before introducing several techniques used to provide secure routing in wireless sensor networks, we will begin with a general overview of several routing protocols that are currently in use. An excellent discussion on many of the attacks on routing protocols is also discussed in [40].

In general, packet routing algorithms are used to exchange messages with sensor nodes that are outside of a particular radio range. This is different than to sensors that are within radio range where packets can be transmitted using a single hop. In such single hop networks security is still a concern, but is more accurately addressed through secure broadcasting and multicasting.

The first packet routing algorithm is based on node identifiers similar to traditional routing. In this case, each sensor is identified by an address and routing to/from the sensor is based on the address. This is generally considered inefficient in sensor networks, where nodes are expected to be addressed by their location, rather than their identifier.

As a consequence of the distaste of routing based on node identifiers, geographic routing protocols have been introduced [41, 7]. One common routing protocol, GPSR [41] allows nodes to send a packet to a region, rather than a particular node. Such a routing protocol lends itself nicely to the concept of data-centric networks. A data-centric network is one in which data are stored by name in the sensor network. Data with the same name are stored at the same node. In fact, data need not be stored anywhere near the sensor responsible for generating the data. When searching the network, searches are therefore based on the data's general name, rather than the identity responsible for holding the data. Security specific to this type of network is discussed in [79].

6.4.2 Techniques for Securing the Routing Protocol

Deng, Han, and Mishra describe an intrusion tolerant routing protocol, INSENS, that is designed to limit the scope of an intruder's destruction and route despite network intrusion without having to identify the intruder [15].

They note that an intruder need not be an actual intrusion on the sensor network, but might simply be a node that is malfunctioning for no particularly malicious reason. Identifying an actual intruder versus a malfunctioning node can be extremely difficult, and for this reason Deng *et al.* make no distinction between the two. The first technique they describe to mitigate the damage done by a potential intruder is to simply employ the use of redundancy. In this case, as described previously under denial of service, multiple identical messages are routed between a source and destination. A message is sent once along several distinct paths with the hope that at least one will arrive at the destination. To discern which, if any, of the messages arriving at the destination are authentic, an authentication scheme can be employed to confirm the message's integrity [15].

Deng *et al.* also make use of an assumed asymmetry between base stations and wireless sensor nodes. They assume that the base stations are somewhat less resource constrained than the individual sensor node. For this reason, they suggest using the base station to compute routing tables on behalf of the individual sensor nodes. This is done in three phases. In the first phase, the base station broadcasts a request message to each neighbor which is then propagated throughout the network. In the second phase, the base station collects local connectivity information from each node. Finally, the base station computes a series of forwarding tables for each node. The forwarding tables will include the redundancy information used for the redundant message transmission described above.

There are several possible attacks that can be made on the routing protocol during each of the three stages described above. In the first phase, a node might spoof the base station by sending a spurious request message [15]. A malicious node might also include a fake path(s) when forwarding the request message to its neighbors. It may not even forward the request message at all.

To counter this, Deng *et al.* use a scheme similar to μTESLA where one-way key chains are used to identify a message originating from the base station.

Tanachaiwiwat, *et al.* present a novel technique named TRANS (Trust Routing for Location Aware Sensor Networks) [79]. The TRANS routing protocol is designed for use in data centric networks. It also makes use of a loose-time synchronization asymmetric cryptographic scheme to ensure message confidentiality. In their implementation, μTESLA is used to ensure message authentication and confidentiality. Using μTESLA, TRANS is able to ensure that a message is sent along a path of trusted nodes while also using location aware routing. The strategy is for the base station to broadcast an encrypted message to all of its neighbors. Only those neighbors who are trusted will possess the shared key necessary to decrypt the message. The trusted neighbor(s) then adds its location (for the return trip), encrypts the new message with its own shared key and forwards the message to its neighbor closest to the destination. Once the message reaches the destination, the recipient is able to authenticate the source (base station) using the MAC that will correspond to the base station. To acknowledge or reply to the message, the destination node can simply forward a return message along the same trusted path from which the first message was received [79].

One particular challenge to secure routing in wireless sensor networks is that it is very easy for a single node to disrupt the entire routing protocol by simply disrupting the route discovery process. Papadimitratos and Haas propose a secure route discovery protocol that guarantees, subject to several conditions, that correct topological information will be obtained [62]. This scenario is somewhat similar to the TRANS protocol mentioned above. The security relies on the MAC (message authentication code) and an accumulation of the node identities along the route traversed by a message. In so doing, a source can discover the sensor network topology as each node along the route from source to destination appends its identity to the message. In order to ensure that the message has not been tampered with, a MAC is constructed and can be verified both at the destination and the source (for the return message from the destination).

A related problem is the concept of wormholes in a sensor network. A wormhole attack is one in which a malicious node eavesdrops on a packet or series of packets, tunnels them through the sensor network to another malicious node, and then replays the packets. This can be done to misrepresent the distance between the two colluding nodes. It can also be used to more generally disrupt the routing protocol by misleading the neighbor discovery process [40].

Often additional hardware, such as a directional antenna [34], is used to defend against wormhole attacks. This, however, can be cost-prohibitive when it comes to large-scale network deployment. Instead, Wang and Bhargava use a visualization approach to identifying wormholes [83]. They first compute a distance estimation between all neighbor sensors, including possible existing wormholes. Using multi-dimensional scaling, they then compute a virtual layout of the sensor network. A surface smoothing strategy is then used to adjust for roundoff errors in the multi-dimensional scaling. Finally, the shape of the resulting virtual network is analyzed. If a wormhole exists within the network, the shape of the virtual network will bend and curve towards the offending nodes. Using this strategy the nodes that participate in the wormhole can be identified and removed from the network. If a network does not contain a wormhole, the virtual network will appear flat [83].

6.4.3 Defending Against the Sybil Attack

To defend against the Sybil attack described previously in Section 5.3, the network needs some mechanism to validate that a particular identify is the only identity being held by a given physical node [59]. Newsome *et al.* describe two methods to validate identities, direct validation and indirect validation. In direct validation a trusted node directly tests whether the joining identity is valid. In indirect validation, another trusted node is allowed to vouch for (or against) the validity of a joining node [59]. Newsome *et al.* primarily describe direct validation techniques, including a radio resource test. In the radio test, a node assigns each of its neighbors a different channel on which to communicate. The node then randomly chooses a channel and listens. If the node detects a transmission on the channel it is assumed that the node transmitting on the channel is a physical node. Similarly, if the node does not detect a transmission on the specified channel, the node assumes that the identity assigned to the channel is not a physical identity.

Another technique to defend against the Sybil attack is to use random key pre-distribution techniques. The idea behind this technique is that with a limited number of keys on a keyring, a node that randomly generates identities will not possess enough keys to take on multiple identities and thus will be unable to exchange messages on the network due to the fact that the invalid identity will be unable to encrypt or decrypt messages.

6.5 Detecting Node Replication Attacks

In [63], Parno, et al. describe two algorithms: randomized multicast, and line-selected multicast. Randomized multicast is an evolution of a node broadcasting strategy. In the simple node broadcasting strategy each sensor propagates an authenticated broadcast message throughout the entire sensor network. Any node that receives a conflicting or duplicated claim revokes the conflicting nodes [63]. This strategy will work, but the communication cost is far too expensive. In order to reduce the communication cost, a deterministic multicast could be employed where nodes would share their locations with a set of witness nodes. In this case, witnesses are computed based on a node's ID. In the event that a node has been replicated on the network, two conflicting locations will be forwarded to the same witness who can then revoke the offending nodes [63]. But since a witness is based on a node's ID, it can easily be computed by an attacker who can then compromise the witness nodes. Thus, securely utilizing a deterministic multicast strategy would require too many witnesses and the communication cost would be too high.

Randomized multicast improves upon the insecurity of deterministic multicast by randomly choosing the witnesses. In the event that a node is replicated two sets of witness nodes are chosen. Assuming a network of size n, if each node derives $\sqrt{n}$ witnesses then the birthday paradox suggests that there will likely be at least one collision [63]. In the event that a collision is detected, the offending nodes can easily be revoked by propagating a revocation throughout the network. Unfortunately, the communication cost of the randomized multicast algorithm is still $O(n^2)$ - too high for large networks.

The line-selected multicast algorithm seeks to further reduce the communication costs of the randomized multicast algorithm. It is based upon rumor routing described in [8]. The idea is that a location claim travelling from source s to destination d will also travel through several intermediate nodes. If each of these nodes records the location claim, then the path of the location claim through the network can be thought of as a line segment [63]. In this case the destination of the location claims is one of the randomly chosen witnesses described in the multicast algorithm. As the location claim routes through the network towards a witness node, the intermediate sensors check the claim. If the claim results in an intersection of a line segment then the nodes originating the conflicting claims are revoked. The line selected multicast algorithm reduces the communication cost to $O(n\sqrt{n})$ as long as each line segment is of length $O(\sqrt{n})$ nodes. The storage cost of the line-selected multicast algorithm is $O(\sqrt{n})$ [63].

6.6 Combating Traffic Analysis Attacks

Strategies to combat the traffic analysis attacks previously described are possible. Deng *et al.* propose using a random walk forwarding technique that occasionally forwards a packet to a node other than the sensor's parent node [16]. This would make it difficult to discern a clear path from the senor to the base station and would help to mitigate the rate monitoring attack, but would still be vulnerable to the time correlation attack. To defend against the time correlation attack, Deng *et al.* suggest a fractal propagation strategy [16]. In this technique a node will (with a certain probability) generate a fake packet when its neighbor is forwarding a packet to the base station. The fake packet is sent randomly to another neighbor who may also generate a fake packet. These packets essentially use a time-to-live (TTL) to decide when forwarding should stop. This effectively hides the base station from time correlation attacks. Since traffic analysis is closely related to privacy violation, we discus traffic analysis to the next subsection.

6.7 Defending Against Attacks on Sensor Privacy

Regarding the attacks on privacy mentioned earlier, there exist effective techniques to counter many of the attacks levied against a sensor. Here we describe several common techniques [28].

6.7.1 Anonymity Mechanisms

Location information that is too precise can enable the identification of a user, or make the continued tracking of movements feasible. This is a threat to privacy. Anonymity mechanisms depersonalize the data before the data is released, which present an alternative to privacy policy-based access control. Researchers have discussed several approaches using anonymity mechanisms, for example, Gruteser and Grunwald [26] analyze the feasibility of anonymizing location information for location-based services in an automotive telematics environment; Beresford and Stajano [6] independently evaluate anonymity techniques for an indoor location system based on the *Active Bat.*

Total anonymity is a difficult problem given the lack of knowledge concerning a node's location. Therefore, a tradeoff is required between anonymity and the need for public information when solving the privacy problem. In [27, 28, 67, 76], three main approaches are proposed:

- **Decentralize Sensitive Data** The basic idea of this approach is to distribute the sensed location data through a spanning tree, so that no single node holds a complete view of the original data.

- **Secure Communication Channel** Using secure communication protocols, such as SPINS [65], the eavesdropping and active attacks can be prevented.

- **Change Data Traffic** De-patterning the data transmissions can protect against traffic analysis. For example, inserting some bogus data can intensively change the traffic pattern when needed.

- **Node Mobility** Making the sensor movable can be effective in defending privacy, especially the location. For example, the *Cricket* system [67] is a location-support system for in-building, mobile, location dependent applications. It allows applications running on mobile and static nodes to learn their physical location by using listeners that hear and analyze information from beacons spread throughout the building. Thus the location sensors can be placed on the mobile device as opposed to the building infrastructure, and the location information is not disclosed during the position determination process and the data subject can choose the parties to which the information should be transmitted.

6.7.2 Policy-based Approaches

Policy-based approaches are currently a hot approach to address the privacy problem. The access control decisions and authentication are made based on the specifications of the privacy policies. In [57], Molnar and Wagner present the concept of private authentication, and give a general scheme for building private authentication with work logarithmic in the number of tags in (but not limited by) RFID (radio frequency identification) applications. In the automotive telematics domain, Duri and colleagues [20] propose a policy-based framework for protecting sensor information, where an in-car computer can act as a trusted agent. Snekkenes [77] presents advanced concepts for specifying policies in the context of a mobile phone network. These concepts enable access control based on criteria such as time of the request, location, speed, and identity of the located object. Myles and colleagues [58] describe an architecture for a centralized location server that controls access from client applications through a set of validator modules that check XML-encoded application privacy policies. Hengartner and Steenkiste [31] point out that access control decisions can be governed by either room or user policies. The room policy specifies who is permitted to find out about the people currently in a room, while the user policy states who is allowed to get location information about another user.

6.7.3 Information Flooding

Ozturk *et al.* propose anti-traffic analysis mechanisms to prevent an outside attacker from tracking the location of a data source, since that information will release the location of sensed objects [61]. The randomized data routing mechanism and phantom traffic generation mechanism are used to disguise the real data traffic, so that it is difficult for an adversary to track the source of data by analyzing network traffic. Based on flooding-based routing protocols, Ozturk *et al.* have developed comparable methods for single path routing to try to solve the privacy problems in sensor network.

- **Baseline Flooding** In the baseline implementation of flooding, every node in the network only forwards a message once, and no node retransmits a message that it has previously transmitted. When a message reaches an intermediate node, the node first checks whether it has received and forwarded that message before. If this is its first time,the node will broadcast the message to all its neighbors. Otherwise, it just discards the message.

- **Probabilistic Flooding** In probabilistic flooding, only a subset of nodes within the entire network will participate in data forwarding, while the others simply discard the messages they receive. One possible weakness of this approach is that some messages may get lost in the network and as a result affect the overall network connectivity. However, as [61] explain later in this section, this problem does not appear to be a significant factor.

- **Flooding with Fake Messages** The previous flooding strategies can only decrease the chances of a privacy violation. An adversary still has a chance to monitor the general traffic and even the individual packets. This observation suggests that one approach to alleviate the risk of source-location privacy breaching is to augment the flooding protocols to introduce more sources that inject fake messages into the network. By doing so, even if the attacker captures the packets, he will have no idea whether the packets are real.

- **Phantom Flooding** Phantom flooding shares the same insights as probabilistic flooding in that they both attempt to direct messages to different locations of the network so that the adversary cannot receive a steady stream of messages to track the source. Probabilistic flooding is not very effective in achieving this goal because shorter paths are more likely to deliver more messages. Therefore, Ozturk *et al.* [61] suggest enticing the attacker away from the real source and towards a fake source, called the phantom source. In phantom flooding, every message experiences two phases: (1) a walking phase,which may be a random walk or a directed walk, and (2) a subsequent flooding meant to deliver the message to the sink. When the source sends out a message, the message is unicast in a random fashion within the first h_{walk} hops (referred to as random walk phase). After the h_{walk} hops, the message is flooded using the baseline flooding technique (referred to as flooding phase).

Similar mechanisms are also used to disguise an adversary from finding the location of a base station by analyzing network traffic [29]. One key problem for these anti-traffic analysis mechanisms is the energy cost incurred by anonymization.

Another strategy used to mask location information from eavesdroppers is presented in [89]. They propose a two way greedy random-walk strategy *GROW* (Greedy Random Walk). In this case, the random walk is taken from both the source and the sink. The sink first initiates a N-hop random walk. The source then initiates a M-hop random walk. Once the source packet reaches an intersection of these two paths, it is forwarded through the path created by the sink. Local broadcasting is used to detect when the two paths intersect. In order to minimize the chance of backtracking along the random walk, the nodes are stored in a bloom filter as the walk progresses. At each stage, the intermediate nodes are checked against the bloom filter to ensure that backtracking is minimized [89].

6.8 Intrusion Detection

We now turn to the area of intrusion detection in wireless sensor networks. It is important to note that in this section we cover intrusion detection as it applies to detecting attacks on the sensor network itself, rather than the popular intrusion detection application being researched for such uses as perimeter monitoring, and so forth.

With that in mind, we note that intrusion detection is not necessarily a category unto itself, but rather has its place in nearly every aspect of sensor network security. Many secure routing schemes attempt to identify network intruders, and key establishment techniques are used in part to prevent intruders from overhearing network data.

Despite the necessity of effective intrusion detection schemes for wireless sensor networks, a good solution has not yet been devised. Of course, this is due largely to the resource constraints present in wireless sensor networks. However, resource constraints are not the only reason. Another problem is that researchers have not yet been able to develop methods of reliably detecting intruders in sensor networks. As such, it is difficult to define characteristics (or signatures) that are specific to a network intrusion as opposed to the normal network traffic that might occur as the result of normal network operations or malfunctions resulting from the environment change.

6.8.1 Background on Intrusion Detection

Traditionally, intrusion detection has focused on two major categories: anomaly based intrusion detection (AID), and misuse intrusion detection (MID) [72]. Anomaly based intrusion detection relies on the assumption that intruders will demonstrate abnormal behavior relative to the legitimate nodes. Thus, the object of anomaly based detection is to detect intrusion based on unusual system behavior. Typically this is done by first developing a profile of the system in normal use. Once the profile has been generated it can be used to evaluate the system in the face of intruders.

The advantage of using an anomaly based system is that it is able to detect previously unknown attacks based only upon knowing that the system behavior is unusual. This is particularly advantageous in wireless sensor networks where it can be difficult to boil an attack down to a signature. However, such flexible intrusion detection comes at a cost. The first is that the anomaly based approach is susceptible to false positives. This is due largely to the fact that it can be difficult to define normal system behaviors. To help combat this, new profiles can be taken of the network to ensure that the profile in use is up-to-date. However, this takes time. And further, even with the most up-to-date profile possible, it can still be difficult to discern unusual, but legitimate, behavior from an actual intrusion. Another fault in the anomaly based intrusion detection techniques is that the computational cost of comparing the current system activity to the profile can be quite high [72]. In the case of a wireless sensor network, such added computation can severely impact the longevity of the network.

In systems based on misuse intrusion detection, the system maintains a database of intrusion signatures. Using these signatures, the system can easily detect intrusions on the network. Further, the system is less prone to false positives as the intrusion signatures are narrowly defined. Such narrowly defined signatures, while leading to fewer false positives, also imply that the intrusion detection system will be unable to detect unknown attacks. This problem can be somewhat mitigated by maintaining an up-to-date signature database. However, since it can be difficult to characterize attacks on wireless sensor networks, such databases may be inherently limited and difficult to generate. An advantage, however, is that the misuse intrusion detection system requires less computation in order to identify intruders as the comparison of network events to the available signatures is relatively low cost [72].

Because both techniques have their strengths and weaknesses, traditional intrusion detection systems use systems that implement both anomaly based intrusion detection and misuse intrusion detection models. This allows such systems to utilize the fast evaluation of the misuse intrusion detection system, but still recognize abnormal system behavior.

6.8.2 Intrusion Detection in Wireless Sensor Networks

Typically a wireless sensor network uses cryptography to secure itself against unauthorized external nodes gaining entry into the network. But cryptography can only protect the network against the external nodes and does little to thwart malicious nodes that already possess one or more keys. Brutch and Ko classify intrusion detection systems (IDS) into two categories: *host-based* and *network-based*. They further classify intrusion detection schemes into those that are signature based, anomaly based, and specification based [9].

Simply put, a host based IDS system operates on operating systems audit trails, system call audit trails, logs, and so on. A network based IDS, on the other hand, operates entirely on packets that have been captured from the network [9]. A signature based IDS simply monitors the network for specific pre-determined signatures that are indicative of an intrusion. In an anomaly based scheme, a standard behavior is defined and any deviation from that

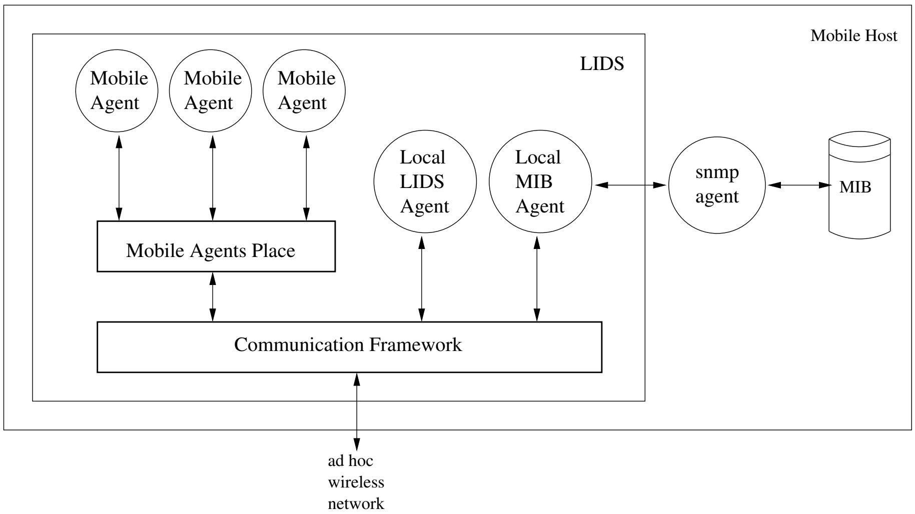

Figure 2: The LIDS architecture from [2].

behavior triggers the intrusion detection system. Finally, a specification based scheme defines a set of constraints that are indicative of a program's or protocol's correct operation [9].

Brutch and Ko describe a series of attacks against several aspects of a wireless sensor network and also introduce three architectures for intrusion detection in wireless sensor networks. The first is termed the stand-alone architecture. In this case, as its name implies, each node functions as an independent intrusion detection system and is responsible for detecting attacks directed toward itself. Nodes do not cooperate in any way [9].

The second architecture is the distributed and cooperative architecture. In this case, an intrusion detection agent still resides on each node (as in the case of the stand-alone architecture) and nodes are still responsible for detecting attacks against themselves (local attacks), but also cooperate to share information in order to detect global intrusion attempts [9].

The third technique proposed by Brutch and Ko is called the hierarchical architecture. These architectures are suitable for multi-layered wireless sensor networks. In this case, Brutch and Ko describe a multi-layered network as one in which the network is divided into clusters with cluster-head nodes responsible for routing within the cluster. The multi-layered network is used primarily for event correlation.

Albers *et al.* describe an intrusion detection architecture based on the implementation of a local intrusion detection system (LIDS) at each node [2]. In order to extend each node's "vision" of the network, Albers suggests that the LIDS existing within the network should collaborate with one another. All LIDS within the network will exchange two types of data, security data and intrusion alerts. The security data is simply used to exchange information with other network hosts. The intrusion alerts, however, are used to inform other LIDS of a locally detected intrusion [2].

A pictorial representation of the LIDS architecture is depicted in Figure 2. MIB (management information base) variables are accessed through SNMP running on the mobile host, where the LIDS components are depicted within the block labeled LIDS. The local MIB is designed to interface with the SNMP agent to provide MIB variable collection from the local LIDS agent or mobile agents. The mobile agents are responsible for both the collection and processing of data from remote hosts, specifically SNMP requests. The agents are capable of migration between individual hosts and are capable of transferring data back to their home LIDS. The local LIDS agent is responsible for detecting and responding to local intrusions as well as responding to events generated by remote nodes [2].

Albers *et al.* propose to use SNMP auditing as the audit source for each LIDS. Rather than simply sending the SNMP messages over an unreliable UDP connection, it is suggested that mobile agents will be responsible for message transporting. In order to detect an intrusion, Albers suggests using either misuse or anomaly detection. When a LIDS detects an intrusion, it should communicate this intrusion to other LIDS on the network. Possible responses include forcing the potential intruder to re-authenticate, or to simply ignore the suspicious node when performing cooperative actions [2]. Although this approach can not be applied to wireless sensor network directly, it is an interesting idea that explores the local information only, which is the key to any intrusion detection techniques in sensor network [22]. In summary, we envision that the intrusion detection in wireless sensors remains an open problem, and more study is needed. Taking the pre-deployment information, such as sensing data distribution, into consideration is a possible direction.

6.9 Secure Data Aggregation

As wireless sensor networks continue to grow in size, so does the amount of data that the sensor networks are capable of sensing. However, due to the computational constraints placed on individual sensors, a single sensor is typically responsible for only a small part of the overall data. Because of this, a query of the wireless sensor network is likely to return a great deal of raw data, much of which is not of interest to the individual performing the query.

Thus, it is advantageous for the raw data to first be processed so that more meaningful data can be gleaned from the network. This is typically done using a series of aggregators. An aggregator is responsible for collecting the raw data from a subset of nodes and processing/aggregating the raw data from the nodes into more usable data.

However, such a technique is particularly vulnerable to attacks as a single node is used to aggregate multiple data. Because of this, secure information aggregation techniques are needed in wireless sensor networks where one or more nodes may be malicious.

6.9.1 Introduction to Data Aggregation and Its Utility

Before discussing the security aspects of secure information aggregation, we first begin with an overview of several information aggregating techniques. Clustering techniques are discussed in [22]. They develop a localized algorithm that uses the directed diffusion technique to achieve a global perspective using only local nodes. In their algorithm, nodes are assigned levels, with level 0 being the lowest level. When a node transmits a message, the number of hops that the message travels is proportional to the node's level. A node can be promoted and demoted. Using this technique, higher level nodes are able to communicate across clusters, while their lower level siblings cannot. This effectively enables localized cluster computation while the higher level nodes can coordinate their cluster's local information to achieve a global solution [22].

If an aggregation node is itself compromised, then all of the data being delivered from the sensor network to the base station may be forged. To detect this, Ye *et al.* describe a statistical en-route filtering mechanism [91]. It utilizes multiple MACs along the path from the aggregator to the base station. Any packet that fails any of the MAC tests will be disregarded.

A more recent technique called TAG is proposed in [54]. In this case, the authors propose an SQL like language that is used for generating queries over the sensor network. The TAG approach is one of a general purpose aggregation. That is, it has not been designed with an application specific intent. It's operation is fairly simple, the base station defines a query using the SQL-like language designed for use in TAG. The sensors then route data back to the base station according to a routing tree. At each point in the tree, data is aggregated according to the routing tree and according to the particular aggregation function that is defined in the initial query [54].

More recently Shrivastava *et al.* propose a summary structure that is able to support fairly complex aggregate functions, such as median and range queries [75]. It's important to note that typical aggregate functions are capable of performing min/max, sum, and average. The more complex aggregates, such as finding the most frequent data values, are typically not supported. They note that the added aggregate functions are not exact. However, they prove strict guarantees on the approximation quality of the queries [75].

Wagner analyzes the resilience of all aggregation techniques in [82], and argues that current aggregation schemes were designed without security in mind and that there are easy attacks against them. Wagner proposes a mathematical framework for formally evaluating the security for aggregation, allowing them to quantify the robustness of an aggregation operator against malicious data. This seminal work opens the door to secure data aggregation in sensor networks; however, the one-level homogeneous aggregation model is too simple to represent real sensor network deployments. Extending the model to a more realistic model, e.g., multi-level and heterogeneous, is an interesting direction.

6.9.2 Secure Data Aggregation Techniques

As was shown above, the idea of information aggregation has been studied in reasonable depth. The problem with the standard information aggregation techniques, however, is that they assume that all nodes are trustworthy. Of course, this is not the case and secure data aggregation techniques will be necessary in many wireless sensor networks.

Przydatek *et al.* describe a secure information aggregation technique (SIA) [68]. They note that sensor networks and data aggregation techniques are vulnerable to a variety of attacks including denial of service attacks as described in 5.2. However, [68] focus their efforts on defending specifically against a type of attack called the stealthy attack. In a stealthy attack, the attacker seeks to provide incorrect aggregation results to the user without the user knowing that the results are incorrect. Therefore, the goal of [68] is to ensure that if a user accepts an aggregate value as correct, then there is a high probability that the value is close to the true aggregation value [68]. In the event that the aggregate value has been tampered with, the user should reject the incorrect results with high probability.

The approach that [68] provide is termed the aggregate-commit-prove technique. As the name would suggest, the technique is composed of three phases. In the first stage, aggregate, the aggregator collects data from the sensors and computes the aggregation result according to a specific aggregate function. Each sensor should share a key with the aggregator. This allows the aggregator to verify that the sensor reading is authentic. However, it is possible that a sensor has been compromised and possesses the key, or that the sensor is simply malfunctioning. The aggregate phase does not prevent such malfunctioning.

In the second phase, the commit phase, the aggregator is responsible for committing to the collected data. This commitment ensures that the aggregator actually uses the data collected from the sensors. One way to perform this commitment is to use a Merkle hash-tree construction [56]. Using this technique the aggregator computes a hash of each input value and the internal nodes are computed as the hash of their children concatenated. The commitment is the root value. The hashing is used to ensure that the aggregator cannot change any input values after having hashed them.

In the final phase, the aggregator is charged with proving the results to the user. The aggregator first communicates the aggregation result and the commitment. The aggregator then uses an interactive proof to prove the correctness of the results. This generally requires two steps. In the first, the user/home server checks to ensure that the committed data is a good representation of the data values in the sensor network. In the second step, the user/home server decides whether the aggregator is lying. This can be done by checking whether or not the aggregation result is close to the committed result [68]. The interactive proof differs depending on the aggregation function that is being used.

Hu and Evans propose a secure aggregation technique that uses the μTESLA protocol for security [33]. In this case, the nodes organize into a tree based hierarchy where the internal nodes act as aggregators. Recall that the μTESLA protocol achieves asymmetry through delayed discloser of symmetric keys. Therefore, a child's parent will be unable to immediately verify the authenticity of the child's data as the key used to generate the MAC will not have been revealed. This technique, however, does not guarantee that nodes and aggregators are providing correct values. To address this problem, the base station is responsible for distributing temporary keys to the network as well as the base station's current μTESLA key, used for validating MACs. Using the μTESLA key, nodes verify their children's

MAC and are responsible for ensuring that the MACs are consistent.

To this end, we argue that secure aggregation techniques play an important role in adopting wireless sensor networks, because of the large amount of raw data and the necessity of the localized in-network processing, and much more investigation is needed.

6.10 Defending Against Physical Attacks

Physical attacks, as we argued in the beginning of the chapter, pose a great threat to wireless sensor networks, because of it's unattended feature and limited resources. Sensor nodes may be equipped with physical hardware to enhance protection against various attacks. For example, to protect against tampering with the sensors, one defense involves tamper-proofing the node's physical package [88]. [3, 4, 43] focus on building tamper-resistant hardware in order to make the actual data and memory contents on the sensor chip inaccessible to attack. Another way is to employ special software and hardware outside the sensor to detect physical tampering.

As the price of the hardware itself gets cheaper, tamper-resistant hardware may become more appropriate in a variety of sensor network deployments. One possible approach to protect the sensors from physical attacks is self-termination. The basic idea is the sensor kills itself, including destroy all data and keys, when it senses a possible attack. This is particularly feasible in the large scale wireless sensor network which has enough redundancy of information, and the cost of a sensor is much cheaper than the lost of being broken (attacked). The key of this approach is detecting the physical attack. A simple solution is periodically conducting neighborhood checking in static deployment. For mobile sensor networks, this is still an open problem.

In [3, 4, 43], the authors describe techniques for extracting protected software and data from smartcard processors. This includes manual microprobing, laser cutting, focused ion-beam manipulation, glitch attacks, and power analysis, most of which are also possible physical attacks on the sensor. Based on an analysis of these attacks, Andersen *et al.* give examples of low-cost protection countermeasures that make such attacks considerably more difficult, including [4]:

- **Randomized Clock Signal** Inserting random-time delays between any observable reaction and critical operations that might be subject to an attack.

- **Randomized Multithreading** Designing a multithread processor architecture that schedules the processor by hardware between two or more threads of execution randomly at a per-instruction level

- **Robust Low-frequency Sensor** Building an intrinsic self-test into the detector. Any attempt to tamper with the sensor should result in the malfunction of the entire processor.
- **Destruction of Test Circuitry** Destroying or disabling the special test circuitry which is for the test engineers, closing the door to microprobing attackers.
- **Restricted Program Counter** Avoid providing a program counter that can run over the entire address space.
- **Top-layer Sensor Meshes** Introducing additional metal layers that form a sensor mesh above the actual circuit and that do not carry any critical signals to be effective annoyances to microprobing attackers.

For the deployment of components outside the sensor, various approaches have been proposed to protect the sensor, and are summarized in [17]. Sastry *et al.* [71] introduce the concept of secure location verification and propose a secure localization scheme, the ECHO protocol, to make sure the location claims are legitimate. In their work, the security rests on physical properties of sound and RF signal propagation. An adversary cannot cheat and claim a shorter distance by starting the ultra-sound response early, because it will not have the nonce. Hu *et al.* [34] introduce directional antennas to defend against wormhole attacks. In [85] the authors study the modeling and defense of sensor networks against *Search-based Physical Attacks*. They define a search-based physical attack model, where the attacker walks through the sensor network using signal detecting equipment to locate active sensors, and then destroys them. In a prior work, they have identified and modeled blind physical attacks [84]. The defense algorithm is executed by individual sensors in two phases: in the first phase, sensors detect the attacker and send out attack notification messages to other sensors; in the second phase, the recipient sensors of the notification message schedule their states to switch. A mechanism named SWATT to verify whether the memory of a sensor node has been changed [74] is proposed by Seshadri *et al.*

6.11 Trust Management

Trust is an old but important issue in any networked environment, whether social networking or computer networking. Trust can solve some problems beyond the power of the traditional cryptographic security. For example, judging the quality of the sensor nodes and the quality of their services, and providing the corresponding access control, e.g., does the data aggregator perform the aggregation correctly? Does the forwarder send out the packet in a timely fashion? These questions are important, but difficult, if not impossible, to answer using existing security mechanisms. We argue that trust management is the key to build trusted, dependable wireless sensor network applications. The trust issue is emerging as sensor networks thrive. However, it is not easy to build a good trust model within a sensor network given the resource limits. Furthermore, in order to keep the sensor nodes independent, we should not assume there is a trust among sensors in advance.

According to the small world principle in the context of social networks and peer-to-peer computing [60], one can employ a path-finder to find paths from a source node to a designated target node efficiently. Based on this observation, Zhu *et al.* [92] provide a practical approach to compute trust in wireless networks by viewing individual mobile devices as a node of a delegation graph G and mapping a delegation path from the source node S to the target node T into an edge in the correspondent transitive closure of the graph G, from which the trust value is computed. In this approach, an undirected transitive signature scheme is used within the authenticated transitive graphs.

In [90], a trust evaluation based security solution is proposed to provide effective security decisions on data protection, secure routing, and other network activities. Logical and computational trust analysis and evaluation are deployed among network nodes. Each node's evaluation of trust on other nodes is based on serious study and inference from trust factors such as experience statistics, data value, intrusion detection results, and references to other nodes, as well as a node owner's preference and policy. Ren *et al.* describe a technique to establish sufficient trust relationships in ad hoc networks with minimum local storage capacity requirements on the mobile nodes [70]. The authors propose a probabilistic solution based on a distributed trust model. A secret dealer is introduced only in the system bootstrapping phase to complement the assumption in trust initialization. With the help of the secret dealer, much shorter and more robust trust chains are able to be constructed with high probability. A fully self-organized trust establishment approach is then adopted to conform to the dynamic membership changes. But the shortcoming of this approach for the common sensor network is that it is not reasonable to introduce a dealer in a totally decentralized ad hoc environment.

The approaches described above are proposed in the context of ad hoc network. For the wireless sensor network, they can not be employed directly because of the capacity of the sensor. Some researchers specifically focus on the sensor networks that have been proposed recently. Ganeriwal and Srivastava propose a reputation-based framework for high integrity sensor networks [23]. Within this framework the authors employ a beta reputation system for reputation representation, updates, and integration. Tanachaiwiwat *et al.* [80] propose a mechanism of location-centric isolation of misbehavior and trust routing in sensor networks. In their trust model, the trustworthiness value is derived from the capacity of the cryptography, availability and packet forwarding. If the trust value is below a specific trust threshold, then this location is considered insecure and is avoided when forwarding packets.

Liang and Shi focus on trust model developing and the analysis of rating aggregation algorithms in the open untrusted environment [48, 49, 50]. Their findings and observations can be applied to wireless sensor networks directly, although the work is performed in the context of peer-to-peer settings. They propose a personalized trust model called PET in [50], which supports the customization of trustworthiness from the view of individual sensors. Regarding how to aggregate the ratings from referrals, they recently analyze the effect of ratings on the trust inference in a comprehensive way [48]. They find that the rating is not always helpful given the limitations of other factors. In the open environment with high dynamics the rating performance degrades and can produce negative effects. They observe that the storage space for saving self-knowledge is a potential bottleneck to the effect of ratings. Their recent simulation results show that it is better to treat the ratings from different evaluators equally given the dynamics of the open environment, and simply averaging ratings is appropriate considering the simplicity of the algorithm design and the low cost in running the system. They argue that the most important issue for building a trust model is adjusting parameters according to environment changes. These suggestions are quite useful for building trust models in the wireless sensor network given their simplicity and cost savings.

7 Conclusions

In this chapter we have described the four main aspects of wireless sensor network security: obstacles, requirements, attacks, and defenses. Within each of those categories we have also sub-categorized the major topics including routing, trust, denial of service, and so on. Our aim is to provide both a general overview of the rather broad area of wireless sensor network security, and give the main citations such that further review of the relevant literature can be completed by the interested researcher.

As wireless sensor networks continue to grow and become more common, we expect that further expectations of security will be required of these wireless sensor network applications. In particular, the addition of public-key cryptography and the addition of public-key based key management described in 6.1.3 will likely make strong security a more realistic expectation in the future. We also expect that the current and future work in privacy and trust will make wireless sensor networks a more attractive option in a variety of new arenas.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, August 2002.

[2] P. Albers and O. Camp. Security in ad hoc networks: A general intrusion detection architecture enhancing trust based approaches. In *First International Workshop on Wireless Information Systems, 4th International Conference on Enterprise Information Systems*, 2002.

[3] R. Anderson and M. Kuhn. Tamper resistance - a cautionary note. In *The Second USENIX Workshop on Electronic Commerce Proceedings*, Oakland, California, 1996.

[4] R. Anderson and M. Kuhn. Low cost attacks on tamper resistant devices. In *IWSP: International Workshop on Security Protocols, LNCS*, 1997.

[5] T. Aura, P. Nikander, and J. Leiwo. Dos-resistant authentication with client puzzles. In *Revised Papers from the 8th International Workshop on Security Protocols*, pages 170–177. Springer-Verlag, 2001.

[6] A. R. Beresford and F. Stajano. Location Privacy in Pervasive Computing. *IEEE Pervasive Computing*, 2(1):46–55, 2003.

[7] P. Bose, P. Morin, I. Stojmenović;, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. *Wirel. Netw.*, 7(6):609–616, 2001.

[8] D. Braginsky and D. Estrin. Rumor routing algorthim for sensor networks. In *WSNA '02: Proceedings of the 1st ACM international workshop on Wireless*

sensor networks and applications, pages 22–31, New York, NY, USA, 2002. ACM Press.

[9] P. Brutch and C. Ko. Challenges in intrusion detection for wireless ad-hoc networks. In *2003 Symposium on Applications and the Internet Workshops (SAINT'03 Workshops)*, 2003.

[10] D. W. Carman, P. S. Krus, and B. J. Matt. Constraints and approaches for distributed sensor network security. Technical Report 00-010, NAI Labs, Network Associates, Inc., Glenwood, MD, 2000.

[11] H. Chan and A. Perrig. Security and privacy in sensor networks. *IEEE Computer Magazine*, pages 103–105, 2003 2003.

[12] H. Chan and A. Perrig. Pike: Peer intermediaries for key establishment in sensor networks. In *IEEE Infocom 2005*, 2005.

[13] H. Chan, A. Perrig, and D. Song. Random key predistribution schemes for sensor networks. In *Proceedings of the 2003 IEEE Symposium on Security and Privacy*, page 197. IEEE Computer Society, 2003.

[14] http://www.xbow.com/wireless_home.aspx, 2006.

[15] J. Deng, R. Han, and S. Mishra. INSENS: intrusion-tolerant routing in wireless sensor networks. In *Technical Report CU-CS-939-02, Department of Computer Science, University of Colorado*, 2002.

[16] J. Deng, R. Han, and S. Mishra. Countermeasuers against traffic analysis in wireless sensor networks. Technical Report CU-CS-987-04, University of Colorado at Boulder, 2004.

[17] J. Deng, R. Han, and S. Mishra. *Security, privacy, and fault tolerance in wireless sensor networks.* Artech House, August 2005.

[18] J. Douceur. The sybil attack. In *Proc. of the 1st International Workshop on Peer-to-Peer Systems (IPTPS'02)*, February 2002.

[19] W. Du, J. Deng, Y. S. Han, and P. K. Varshney. A pairwise key pre-distribution scheme for wireless sensor networks. In *CCS '03: Proceedings of the 10th ACM conference on Computer and communications security*, pages 42–51, New York, NY, USA, 2003. ACM Press.

[20] S. Duri, M. Gruteser, X. Liu, P. Moskowitz, R. Perez, M. Singh, and J. Tang. Framework for security and privacy in automotive telematics. In *In 2nd ACM International Worksphop on Mobile Commerce*, 2000.

[21] L. Eschenauer and V. D. Gligor. A key-management scheme for distributed sensor networks. In *Proceedings of the 9th ACM conference on Computer and communications security*, pages 41–47. ACM Press, 2002.

[22] D. Estrin, R. Govindan, J. S. Heidemann, and S. Kumar. Next century challenges: Scalable coordination in sensor networks. In *Mobile Computing and Networking*, pages 263–270, 1999.

[23] S. Ganeriwal and M. Srivastava. Reputation-based framework for high integrity sensor networks. In *Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*, Washington DC, USA, 2004.

[24] S. Ganeriwal, S. Čapkun, C.-C. Han, and M. B. Srivastava. Secure time synchronization service for sensor networks. In *WiSe '05: Proceedings of the 4th ACM workshop on Wireless security*, pages 97–106, New York, NY, USA, 2005. ACM Press.

[25] G. Gaubatz, J.P. Kaps, and B. Sunar. Public key cryptography in sensor networks - revisited. In *1st European Workshop on Security in Ad-Hoc and Sensor Networks (ESAS 2004)*, 2004.

[26] M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proceedings of the First International Conference on Mobile Systems, Applications, and Services (MobiSys)*. USENIX, 2003.

[27] M. Gruteser and D. Grunwald. A methodological assessment of location privacy risks in wireless hotspot networks. In *First International Conference on Security in Pervasive Computing*, 2003.

[28] M. Gruteser, G. Schelle, A. Jain, R. Han, and D. Grunwald. Privacy-aware location sensor networks. In *9th USENIX Workshop on Hot Topics in Operating Systems (HotOS IX)*, 2003.

[29] N. Gura, A. Patel, A. Wander, H. Eberle, and S. Shantz. Comparing elliptic curve cryptography and rsa on 8-bit cpus. In *In 2004 workshop on Cryptographic Hardware and Embedded Systems*, August 2004.

[30] C. Hartung, J. Balasalle, and R. Han. Node compromise in sensor networks: The need for secure systems. Technical Report Technical Report CU-CS-988-04, Department of Computer Science, University of Colorado at Boulder, 2004.

[31] U. Hengartner and P. Steenkiste. Protecting Access to People Location Information. In *Proceedings of First International Conference on Security in Pervasive Computing (to appear)*, LNCS. Springer, Mar 2003.

[32] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. E. Culler, and K. Pister. System architecture directions for networked sensors. In *Architectural Support for Programming Languages and Operating Systems*, pages 93–104, 2000.

[33] L. Hu and D. Evans. Secure aggregation for wireless networks. In *SAINT-W '03: Proceedings of the 2003 Symposium on Applications and the Internet Workshops (SAINT'03 Workshops)*, page 384. IEEE Computer Society, 2003.

[34] L. Hu and D. Evans. Using directional antennas to prevent wormhole attacks. In *In 11th Annual Network and Distributed System Security Symposium*, February 2004.

[35] Y. Hu, A. Perrig, and D. B. Johnson. Packet leashes: a defense against wormhole attacks in wireless networks. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies.*, volume 3, pages 1976–1986, 2003.

[36] Q. Huang, J. Cukier, H. Kobayashi, B. Liu, and J. Zhang. Fast authenticated key establishment protocols for self-organizing sensor networks. In *Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications*, pages 141–150. ACM Press, 2003.

[37] J. Hwang and Y. Kim. Revisiting random key pre-distribution schemes for wireless sensor networks. In *Proceedings of the 2nd ACM workshop on Security of Ad hoc and Sensor Networks (SASN '04)*, pages 43–52, New York, NY, USA, 2004. ACM Press.

[38] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *Mobile Computing and Networking*, pages 56–67, 2000.

[39] C. Karlof, N. Sastry, and D. Wagner. Tinysec: A link layer security architecture for wireless sensor networks. In *Second ACM Conference on Embedded Networked Sensor Systems (SensSys 2004)*, pages 162–175, November 2004.

[40] C. Karlof and D. Wagner. Secure routing in wireless sensor networks: Attacks and countermeasures. *Elsevier's AdHoc Networks Journal, Special Issue on Sensor Network Applications and Protocols*, 1(2–3):293–315, September 2003.

[41] B. Karp and H. T. Kung. GPSR: greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 243–254. ACM Press, 2000.

[42] T. Kaya, G. Lin, G. Noubir, and A. Yilmaz. Secure multicast groups on ad hoc networks. In *Proceedings of the 1st ACM workshop on Security of Ad hoc and Sensor Networks (SASN '03)*, pages 94–102. ACM Press, 2003.

[43] O. Kömerling and M. G. Kuhn. Design principles for tamper-resistant smartcard processors. In *appeared in the USENIX Workshop on Smartcard Technology proceedings*, Chicago, Illinois, USA, May 1999.

[44] Y. W. Law, J. Doumen, and P. Hartel. Survey and benchmark of block ciphers for wireless sensor networks. *ACM Trans. Sen. Netw.*, 2(1):65–93, 2006.

[45] L. Lazos and R. Poovendran. Secure broadcast in energy-aware wireless sensor networks. In *IEEE International Symposium on Advances in Wireless Communications (ISWC'02)*, 2002.

[46] L. Lazos and R. Poovendran. Energy-aware secure multicast communication in ad-hoc networks using geographic location information. In *Proceedings of IEEE International Conference on Acoustics Speech and Signal Processing*, 2003.

[47] L. Lazos and R. Poovendran. Serloc: Robust localization for wireless sensor networks. *ACM Trans. Sen. Netw.*, 1(1):73–100, 2005.

[48] Z. Liang and W. Shi. Analysis of recommendations on trust inference in the open environment. Technical Report MIST-TR-2005-002, Department of Computer Science, Wayne State University, February 2005.

[49] Z. Liang and W. Shi. Enforcing cooperative resource sharing in untrusted peer-to-peer environment. *ACM Journal of Mobile Networks and Applications (MONET)*, 10(6):771–783, 2005.

[50] Z. Liang and W. Shi. PET: A PErsonalized Trust model with reputation and risk evaluation for P2P resource sharing. In *Proceedings of the HICSS-38*, Hilton Waikoloa Village Big Island, Hawaii, January 2005.

[51] D. Liu and P. Ning. Efficient distribution of key chain commitments for broadcast authentication in distributed sensor networks. In *Proceedings of the 10th Annual Network and Distributed System Security Symposium*, pages 263–276, 2003.

[52] D. Liu and P. Ning. Multilevel μTESLA: Broadcast authentication for distributed sensor networks. *Trans. on Embedded Computing Sys.*, 3(4):800–836, 2004.

[53] D. Liu, P. Ning, and R. Li. Establishing pairwise keys in distributed sensor networks. *ACM Trans. Inf. Syst. Secur.*, 8(1):41–77, 2005.

[54] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. Tag: a tiny aggregation service for ad-hoc sensor networks. *SIGOPS Oper. Syst. Rev.*, 36(SI):131–146, 2002.

[55] D. J. Malan, M. Welsh, and M. D. Smith. A public-key infrastructure for key distribution in tinyos based on elliptic curve cryptography. In *First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON*, 2004.

[56] R. C. Merkle. Protocols for public key cryptosystems. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*, April 1980.

[57] D. Molnar and D. Wagner. Privacy and security in library rfid : Issues, practices, and architectures. In *ACM CCS*, 2004.

[58] G. Myles, A. Friday, and N. Davies. Preserving Privacy in Environments with Location-Based Applications. *IEEE Pervasive Computing*, 2(1):56–64, 2003.

[59] J. Newsome, E. Shi, D. Song, and A. Perrig. The sybil attack in sensor networks: analysis & defenses. In *Proceedings of the third international symposium on Information processing in sensor networks*, pages 259–268. ACM Press, 2004.

[60] A. Oram. *Peer-to-Peer: Harnessing the Power of Disruptive Technologies.* O'Reilly & Associates, March 2001.

[61] C. Ozturk, Y. Zhang, and W. Trappe. Source-location privacy in energy-constrained sensor network routing. In *Proceedings of the 2nd ACM workshop on Security of Ad hoc and Sensor Networks*, 2004.

[62] P. Papadimitratos and Z. J. Haas. Secure routing for mobile ad hoc networks. In *Proceedings of the SCS Communication Networks and Distributed System Modeling and Simulation Conference (CNDS 2002)*, 2002.

[63] B. Parno, A. Perrig, and V. Gligor. Distributed detection of node replication attacks in sensor networks. In *Proceedings of IEEE Symposium on Security and Privacy*, May 2005.

[64] A. Perrig, J. Stankovic, and D. Wagner. Security in wireless sensor networks. *Commun. ACM*, 47(6):53–57, 2004.

[65] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler. Spins: security protocols for sensor networks. *Wireless Networking*, 8(5):521–534, 2002.

[66] R. Di Pietro, L. V. Mancini, Y. W. Law, S. Etalle, and P. Havinga. LKHW: A directed diffusion-based secure multicast scheme forwireless sensor networks. In *First International Workshop on Wireless Security and Privacy (WiSPr'03)*, 2003.

[67] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In *Proc. of the Sixth Annual ACM International Conference on Mobile Computing and Networking (MOBICOM)*, August 2000.

[68] B. Przydatek, D. Song, and A. Perrig. Sia: Secure information aggregation in sensor networks, 2003.

[69] S. Rafaeli and D. Hutchison. A survey of key management for secure group communication. *ACM Comput. Surv.*, 35(3):309–329, 2003.

[70] K. Ren, T. Li, Z. Wan, F. Bao, R. H. Deng, and K. Kim. Highly reliable trust establishment scheme in ad hoc networks. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 45:687–699, August 2004.

[71] N. Sastry, U. Shankar, and D. Wagner. Secure verification of location claims. In *ACM Workshop on Wireless Security*, September 2003.

[72] I. Sato, Y. Okazaki, and S. Goto. An improved intrusion detection method based on process profiling. *IPSJ Journal*, 43(11):3316–3326, 2002.

[73] B. Schneier. *Applied Cryptography.* Second Edition, John Wiley & Sons, 1996.

[74] A. Seshadri, A. Perrig, L. van Doorn, and P. Khosla. Swatt: Software-based attestation for embedded devices. In *In Proceedings of the IEEE Symposium on Security and Privacy*, May 2004.

[75] N. Shrivastava, C. Buragohain, D. Agrawal, and S. Suri. Medians and beyond: new aggregation techniques for sensor networks. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 239–249. ACM Press, 2004.

[76] A. Smailagic, D. P. Siewiorek, J. Anhalt, and Y. Wang D. Kogan. Location sensing and privacy in a context aware computing environment. In *Pervasive Computing*, 2001.

[77] E. Snekkenes. Concepts for personal location privacy policies. In *Proceedings of the 3rd ACM conference on Electronic Commerce*, pages 48–57. ACM Press, 2001.

[78] J. A. Stankovic et al. Real-time communication and coordination in embedded sensor networks. *Proceedings of the IEEE*, 91(7):1002–1022, July 2003.

[79] S. Tanachaiwiwat, P. Dave, R. Bhindwale, and A. Helmy. Poster abstract secure locations: routing on trust and isolating compromised sensors in location-aware sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 324–325. ACM Press, 2003.

[80] S. Tanachaiwiwat, P. Dave, R. Bhindwale, and A. Helmy. Location-centric isolation of misbehavior and trust routing in energy-constrained sensor networks, April 2004.

[81] S. Čapkun and J.-P. Hubaux. Secure positioning in wireless networks. *IEEE Journal on Selected Areas in Communications*, 24(2):221–232, 2006.

[82] D. Wagner. Resilient aggregation in sensor networks. In *Proceedings of the 2nd ACM workshop on Security of Ad hoc and Sensor Networks (SASN '04)*, pages 78–87, New York, NY, USA, 2004. ACM Press.

[83] W. Wang and B. Bhargava. Visualization of wormholes in sensor networks. In *WiSe '04: Proceedings of the 2004 ACM workshop on Wireless security*, pages 51–60, New York, NY, USA, 2004. ACM Press.

[84] X. Wang, W. Gu, S. Chellappan, K.t Schoseck, and Dong Xuan. Lifetime optimization of sensor networks under physical attacks. In *Proc. of IEEE Internationl Conference on Communications*, May 2005.

[85] X. Wang, W. Gu, S. Chellappan, Dong Xuan, and Ten H. Laii. Search-based physical attacks in sensor networks: Modeling and defense. Technical report, Dept. of Computer Science and Engineering, The Ohio-State University, February 2005.

[86] X. Wang, W. Gu, K. Schosek, S. Chellappan, and D. Xuan. Sensor network configuration under physical attacks. Technical Report Technical Report (OSU-CISRC-7/ 04-TR45), Dept. of Computer Science and Engineering, The Ohio-State University, July 2004.

[87] R. Watro, D. Kong, S. Cuti, C. Gardiner, C. Lynn, and P. Kruus. Tinypk: securing sensor networks with public key technology. In *Proceedings of the 2nd ACM workshop on Security of Ad hoc and Sensor Networks (SASN '04)*, pages 59–64, New York, NY, USA, 2004. ACM Press.

[88] A. D. Wood and J. A. Stankovic. Denial of service in sensor networks. *Computer*, 35(10):54–62, 2002.

[89] Y. Xi, L. Schwiebert, and W. Shi. Preserving privacy in monitoring-based wireless sensor networks. In *Proceedings of the 2nd International Workshop on Security in Systems and Networks (SSN '06),*. IEEE Computer Society, 2006.

[90] Z. Yan, P. Zhang, and T. Virtanen. Trust evaluation based security solution in ad hoc networks. In *NordSec 2003, Proceedings of the Seventh Nordic Workshop on Secure IT Systems*, 2003.

[91] F. Ye, H. Luo, S. Lu, and L. Zhang. Statistical en-route detection and filtering of injected fase data in sensor networks. In *IEEE INFOCOM 2004*, 2004.

[92] H. Zhu, F. Bao, R. H. Deng, and K. Kim. Computing of trust in wireless networks. In *Proceedings of 60th IEEE Vehicular Technology Conference*, Los Angles, California, September 2004.

[93] S. Zhu, S. Setia, and S. Jajodia. Leap: efficient security mechanisms for large-scale distributed sensor networks. In *CCS '03: Proceedings of the 10th ACM conference on Computer and communications security*, pages 62–72, New York, NY, USA, 2003. ACM Press.

[94] http://www.zigbee.org/, 2005.