Simulation and Evaluation of Security and Intrusion Detection in IEEE 802.15.4 Network

A thesis presented
by

Fereshteh Amini

to
The Department of Computer Science
in partial fulfillment of the requirements
for the degree of
Master of Science
in the subject of

Computer Science

The University of Manitoba
Winnipeg, Manitoba
June 2008

Thesis advisor

Dr. Jelena Mišić and Dr. Rasit Eskicioglu

Author

Fereshteh Amini

Simulation and Evaluation of Security and Intrusion Detection in IEEE 802.15.4 Network

Abstract

With the fast growth of wireless sensor technology, the idea of using Wireless Sensor Networks (WSNs) in more critical applications of real life is becoming popular. WSNs are restricted in energy, memory and bandwidth which makes them particularly vulnerable against intrusions. Security is important and needs to be considered for any WSN. The standard specification of IEEE 802.15.4, as one technology of such networks, introduces many constraints which makes the application of security even harder. In addition, WSNs are vulnerable to an intruder's malicious attacks because of their characteristics and the nature of applications they are used for. In this thesis, I describe a secure data exchange protocol including a key exchange mechanism based on the ZigBee specification and built on top of IEEE 802.15.4 link layer. All nodes will apply power management technique based on the constant event sensing reliability required by the coordinator. Power management generates random sleep times by every node which on average fairly distributes the sensing load among the nodes. Key exchange is initiated by cluster coordinator after some given number of sensing packets have been received by the coordinator. On the other hand cheap devices and accessibility of these sensors, encourage adversaries to launch physical

attacks. This can be done by compromising a node inside the network or introducing an extra malicious node to the network. I have simulated key exchange and power management technique into an IEEE 802.15.4 cluster's reliable sensing function. I evaluate the impact of security function and its periodicity on cluster performance. I have also simulated attacks in which sensor nodes are compromised and could be accessed by intruders to maliciously fool the network. I have proposed and simulated intrusion detection mechanisms utilized to detect such attacks. The results show high completeness of detection and suggest parameter selection values to keep high level of performance.

Contents

List of Figures

Acknowledgments

First and foremost I would like to thank my supervisors, Dr. Jelena Mišić and Dr. Rasit Eskicioglu, for all their support. My special thanks go to Dr. Jelena Mišić whose guidance and help I am truly grateful of and without her support this thesis would not have been possible. She was the one who made me interested in the field of networking because of her great lectures in the very first courses I took at the Univerity of Manitoba. I thank Dr. Rasit Eskicioglu for his invaluable advice during my MSc and specially during my first year at the University of Manitoba.

Thanks to Dr. Pourang Irani form department of computer science and Dr. Pradeepa Yahampath from department of electrical and computer engineering for accepting to be in my thesis defense committee, pointing out the mistakes and problems in the thesis, and their helpful comments to make the final version of the thesis better.

My sincerest thanks go to Hossein, my beloved husband, who contributed in countless ways to the successful completion of this work. This thesis is as much a product of his support and patience as it is of my efforts. I thank him for his love during past couple of years and his promise to do so for the rest of our lives.

Many thanks to Moazzam Khan for his insightful discussions and comments about my work and for being there for me when I needed help. Working with him in a team on the first steps of this research has made it so much easier.

I thank my Mom, Dad, and my brothers Reza and Morteza for their love and encouragement throughout all of my university studies.

My appreciation goes to my friends at the Wireless Sensor Network's Lab. for their support and for making my time here more enjoyable.

To Hossein
whose strong presence and love in decisive moments of my life
has been the source of motivation and inspiration.

Chapter 1

Introduction

Wireless Sensor Networks (WSNs), are becoming increasingly popular with many potential applications including general engineering, health, military, environment science, etc. They consist of large number of tiny sensor nodes with limited energy, memory, bandwidth, and processing power. Sensor nodes can be randomly deployed in inhospitable places. A WSN is self-organized with collaboration between nodes. Base station, which is usually a powerful computer with more computational resources, energy, and storage, is present and receives aggregated data from the sensors.

Low Rate Wireless Personal Area Networks (LR-WPAN) [3] are enabling technologies for WSN. IEEE 802.15.4 LR-WPANs were introduced to satisfy the need for a low cost, low power and short-range networks. WPANs might be involved in critical applications, such as health care, home security systems, and environmental monitoring (e.g. presence of bacteria in water/soil). In these applications, WPANs are of interest to adversaries. The nature of these networks (i.e. wireless communica-

tion, limited resources and physically accessible) makes them very vulnerable to an intruder's malicious attacks. WPANs are often unattended but physically reachable from the outside world and are comprised of cheap small devices. Therefore, they must be secured to prevent an intruder from obstructing the delivery of correct sensor data and from forging sensor data.

The 802.15.4 network specification [3] outlines some basic security services at the data link layer that can be combined with advanced techniques at the upper layers to implement a comprehensive security solution. For example, the recent ZigBee specification [6] implements a number of protocols, including security-related ones, that can be deployed in an IEEE 802.15.4 network. Given that the IEEE 802.15.4 devices are typically severely constrained in terms of their communication and computational resources, the implementation of such solutions is likely to impose a significant performance overhead. Currently, not many wireless sensor network overhead statistics are available when security is employed in such networks. Sensor network application developers and network administrators always need these overhead statistics in choosing the security option that best suites the security for a particular threat environment. For evaluating these security overheads on wireless sensor networks, we will simulate IEEE 802.15.4 media access control layer and secure data exchange once the devices exchange link keys with the PAN coordinator. We will measure communication costs that are incurred after employing these security features under different inputs to wireless sensor network model.

Reliable sensing requires guaranteed data integrity and in some cases data confidentiality. This means that each node must possess at least one secret symmetric

key by which it can provide Packet Authentication Code and (optionally) encrypt the data. This opens arena for key-compromise attacks where attacker tries to compute the key in order to listen or modify the data. Periodic key update can be used as good preventive methods against key-compromise attacks.

Key update provides an automated mechanism for restricting the amount of data which may be exposed when a link key is compromised. The key update frequency depends on the key update overheads and threat environment under which a network is working. Hence controlling the lifetime of keys and determination of how the key update occurs is a technical challenge. In [40] authors reported the activity management and network behavior without considering any security parameters. In this thesis, I develop a simulation model for the cluster behavior including periodic key exchange (with a variable update threshold), power management and sensing data application. For activity management, nodes in the cluster apply sleep technique in order to deliver only the required number of packets per second (which we will call event sensing reliability) to the coordinator. We obtained simulation results to evaluate the overhead of key exchange in terms of medium behavior, total number of delivered packets, nodes' utilization and its effect on node's lifetime.

The frequency of updating keys between nodes and the PAN Coordinator is reciprocally related to the power consumption of the nodes [28]. The tradeoff between efficiency and accuracy can keep doors open for an intruder's attack. Applying intrusion prevention techniques, such as encryption and authentication can protect WPANs from some types of attacks. However, they can not eliminate all attacks and there are some attacks for which there are no known prevention methods. Moreover,

there is no guarantee that the preventive security methods will be able to completely block the intruders. As a result, it is necessary to have some intrusion detection mechanism in place.

Amongst various threats to WPANs and to a IEEE 802.15.4 network in particular, the attacks which are developed as a result of node compromise have certain importance. Although having tamper-proof hardware for sensor nodes is possible, it is expensive and therefore usually the nodes' hardware is vulnerable to physical attacks and an adversary could capture sensor nodes within such a network and get access to the IDs and keys by reading the memory. These attacks usually target critical functions and mechanism exploited in the network and can cause a lot of damage if not detected. Therefore it is important to develop detection mechanisms to detect and fight back against these kinds of attacks.

In this thesis, I have simulated and evaluated security and intrusion detection in an IEEE 802.15.4 network. I have simulated key exchange and power management technique into an IEEE 802.15.4 cluster's reliable sensing function. I evaluate the impact of security function and its periodicity on cluster performance. I have minimized the energy consumption by incorporating an energy management technique in my simulation. I have also simulated attacks in which sensor nodes are compromised and could be accessed by intruders to maliciously fool the network. I have proposed and simulated intrusion detection mechanisms that can be used to detect such attacks. The results show high completeness of detection and suggest parameter selection values to keep high level of performance.

In the remainder of this thesis, I will give a background overview of wireless

sensor networks and IEEE 802.15.4 networks in particular in Sections 2.1 and 2.2 of Chapter 2. The background information and related work regarding security of wireless sensor networks and IEEE 802.15.4 is presented in details in Sections 2.3 and 2.4. Section 2.5.1 and Section 2.5.2 include some basic information and current research about intrusion detection in wireless sensor networks as parts Chapter 2. Chapter 3 contains the detailed discussion on security measures considered in my simulation model of the IEEE 802.15.4 cluster. Power management is also considered while security is provided and is described in Section 3.6. In Section 3.7, I provide some information about why interconnection of clusters in IEEE 802.15.4 is necessary. Chapter 4 is dedicated to description and related work of different attacks I have chosen to simulate in my model together with their detection algorithms. Later in Chapter 5, I will talk about the details, assumptions, and settings of every scenario in my simulation environment. This chapter also includes the results obtain from the experiments followed by evaluation and analysis of the findings. Finally, I will provide a summary as the conclusion of my thesis in Chapter 6 and suggest potential future work.

Chapter 2

Background and Related Work

2.1 Wireless Sensor Networks at a Glance

Sensor nodes are autonomous devices with limited power, computational resources, and memory. WSNs are used for monitoring various physical phenomena over sometimes large and geographically distributed areas. Applications of WSNs range from general engineering to health care [5]. For example, they can be used for automotive driving, sensing and maintenance in industrial plants. Agriculture and environment sciences can also benefit from utilizing WSNs.

Many factors need to be considered for WSNs' architecture. There is a large amount of cross-layer interconnection in protocol layers of sensor networks. For example, power management, which is one of the important functions to be considered in a WSN, naturally belongs to the application layer that "knows" which event sensing reliability (i.e. sensing throughput) is required per cluster. However, power consumption in each cluster depends on routing decisions and number of packet collisions.

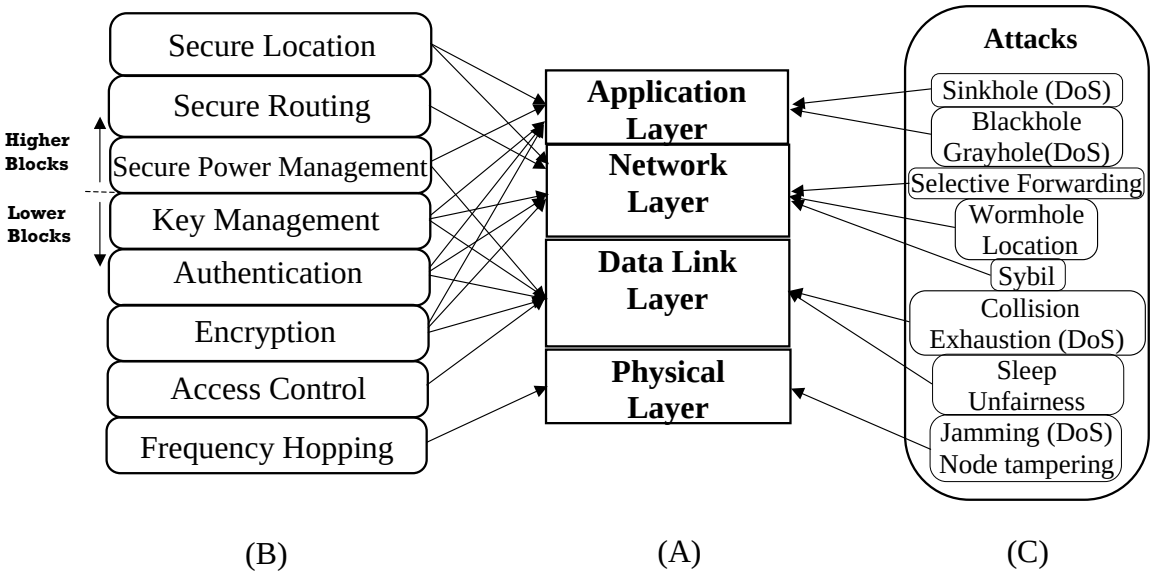


Figure 2.1: The anatomy of security in wireless sensor networks.

Therefore, all three layers must cooperate to determine the sleep period per node which will result in the required event sensing reliability. This leads us to consider a cross-layer architecture for wireless sensor networks (Figure 2.1(A)).

A unique feature of sensor networks is the cooperative effort of sensor nodes. Besides sensing, sensor nodes use their on-board processor to do simple computations locally and then transmit only the required and partially processed data. Therefore, one of the design issues of WSNs is routing. Sensor nodes have different traffic patterns compared to traditional networks and they involve data aggregation and different addressing. The important components of sensor network protocols are power management, location management and routing which should be designed by considering the previous two components.

2.2 A Quick Overview of IEEE 802.15.4 Networks as a Technology for Sensor Networks

As mentioned by the IEEE 802.15.4 standard [3], some of the goals in creating LR-WPANs are reliable data transfer, short-range operation, extremely low cost, and a reasonable battery life, while maintaining a simple and flexible protocol. It has been shown that the effective bandwidth to the application in an IEEE 802.15.4 network is less than 20 percent of the raw bandwidth due to the communication overhead [40]. The two topology options for the network are the peer-to-peer and the star topologies. The former applies to a network in which the nodes can directly communicate with one another and in the latter topology, nodes communicate via a single central controller called the Personal Area Network (PAN) coordinator. After deciding on a PAN identifier, the PAN coordinator may decide whether a device can join the PAN. In peer-to-peer topology, all the devices have the same initial energy, computational resources and link capacity, while in star topology, it is possible for the PAN coordinator to have higher energy and computational resources. As discussed earlier, sensor nodes may be deployed in hard to access areas where changing the batteries of sensor nodes is difficult and in some cases (e.g. sensor networks covering large geographic areas) impossible. Therefore, when in the peer-to-peer networks the sensors closer to the base station die [36], the whole network becomes unavailable. The choice of topology is a tradeoff between the node simplicity and homogeneity versus the duration of the network lifetime. It has been shown in [58] that networks, in which nodes with a lot of packet relaying have higher power resources than ordinary

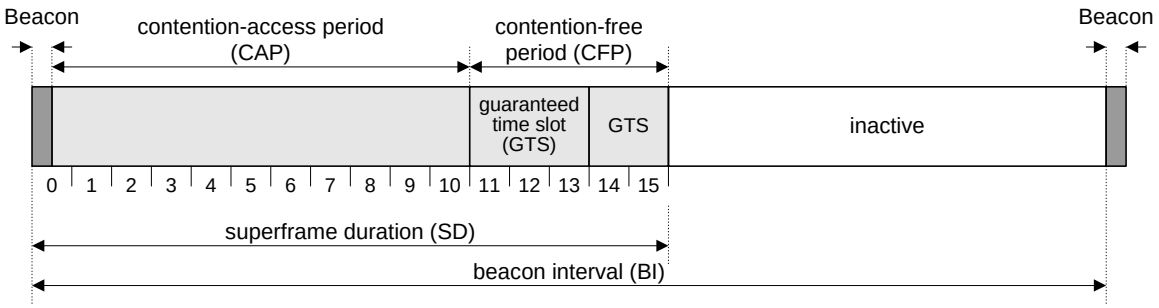


Figure 2.2: The superframe structure.

nodes, can achieve longer lifetimes. This motivates us toward the use of star topology, with the PAN coordinator having higher power resources.

When choosing the star topology, beacon-enabled communications are considered. In this form of communication, devices first listen for the network beacon. When the beacon is found, the device synchronizes to the superframe structure. Superframe in IEEE 802.15.4 is referred to cycles of time durations bounded by network beacons and includes a beacon frame, a contention access period, a contention free period, and an inactive portion (Figure 2.2). At the appropriate point, the device transmits its data packet, using a channel access mechanism namely slotted Carrier Sense Multiple Access with a Collision Avoidance (CSMA-CA) [3], to the coordinator(uplink). The coordinator sends an acknowledgment frame after it receives the data successfully.

When the PAN coordinator has something to send to a device (downlink), it informs the device by including in the network beacon that a data message is pending. The device periodically listens to network beacon and, if a message is pending, transmits a request frame to the coordinator using slotted CSMA-CA. The coordinator acknowledges the successful reception of the data request by transmitting an

acknowledgment frame. The pending data frame is then sent to the destination device. The device acknowledges the successful reception of the data by transmitting an acknowledgment frame.

2.3 Security in Wireless Sensor Networks

The security architecture of WSNs is different from other kinds of networks due to their special characteristics [20]. First, we have to make sensor networks technically and economically viable as sensor devices are limited in their energy, computation, and communication capabilities. Second, unlike traditional networks, sensors are often deployed in accessible areas, presenting the added risk of physical attack. Third, sensor networks interact closely with their physical environments and with people, posing new security problems. Consequently, existing security mechanisms are inadequate, and new ideas are needed [45].

The security of a network is determined by the security over all layers. For example, confidentiality, integrity, and availability typically address security of the link layer. Referring to Figure 2.1, we note that securing the link layer provides a certain level of security to the layers above; however, it does not address security problems in the physical layer below, most notably jamming. In general, an insecure physical layer may practically render the entire network insecure, even if the layers above are secure. This is especially true in the sensor network environment since basic wireless communication is inherently not secure. In the rest of this section, I will discuss some security considerations for WSNs followed by security consideration for IEEE 802.15.4 networks.

2.3.1 Security View through Basic Security Blocks

Service **integrity** is an important security requirement. Above the networking layer, the sensor network usually implements several application-level services. Important components of integrity are data authenticity and origin authenticity. Using origin authenticity we make sure that messages sent from an unauthorized sender will be discarded and using message authenticity, if an adversary modifies a message from an authorized sender while the message is in transit, the receiver should be able to detect this tampering. **Authentication** is necessary to enable sensor nodes to detect maliciously injected or spoofed packets. Almost all applications require data authentication. In military and safety-critical applications and even in civilian applications such as office or home applications, where we expect a relatively secure environment, the adversary may inject false data or malicious routing information. Although authentication tries to prevent outsiders from injecting or spoofing packets, it does not solve the problem of compromised nodes. Since a compromised node has the secret keys of a legitimate node, it can authenticate itself to the network. However, we may be able to use intrusion detection techniques to find the compromised nodes and revoke their cryptographic keys network-wide.

Providing **availability** requires that the sensor network be functional throughout its lifetime. Denial of Service (DoS) attacks result in loss of availability. In practice, loss of availability may have serious impacts. For example in a manufacturing monitoring application, loss of availability may cause failure to detect a potential accident and result in financial loss; in a battlefield surveillance application, loss of availability may open a back door for enemy invasion. When considering availability

in sensor networks, it is important to achieve graceful degradation in the presence of node compromise or benign node failures.

Ensuring the secrecy, also known as **confidentiality** of sensed data, is important for protecting data from eavesdroppers. We can use standard encryption functions and a shared secret key between the communicating parties to achieve secrecy. However, encryption itself is not sufficient for protecting the privacy of data, as an eavesdropper can perform traffic analysis on the overheard ciphertext, and this can release sensitive information about the data. In addition to encryption, privacy of sensed data also needs to be enforced through access control policies at the base station to prevent misuse of information.

2.3.2 Security View through Secure Blocks of Protocol Layer

To provide security services discussed above, WSNs utilize several security mechanisms. The "higher blocks" in Figure 2.1(B) use the functionalities presented by the "lower blocks". We can say that the two block sections are orthogonally related. *Secure Location*, which is a mechanism applied through application and network layers of protocol stack, will need to have authentication, encryption and access control to provide a secure location service on top of other services. Location service provides geographical location of the node which accompanies report of data. It is necessary to provide integrity of this service as well. The same goes for *Secure Routing* and *Secure Power Management*. Secure routing requires services from lower blocks such as authentication and access control since the integrity of routing request/reply and neighborhood information must be protected. Power management service guarantees

required event sensing reliability and network lifetime. Integrity of this service is important.

2.3.3 Security View through Possible Attacks on Security Blocks

As shown in Figure 2.1(B) different security mechanisms are applied to ensure security over all protocol layers in WSNs. Attacks in each layer are possible and can be categorized as insider and outsider attacks. In an outsider attack, the attack node is not an authorized participant of the sensor network. As the sensor network communicates over a wireless channel, a passive attacker can easily eavesdrop on the networks radio frequency range to steal private or sensitive information. The adversary can also alter or spoof packets to attack the authenticity of communication or inject interfering wireless signals to jam the network. Another form of outsider attack is to disable sensor nodes. An attacker can inject useless packets to drain the receivers battery, or he can capture and physically destroy nodes. A failed node is similar to a disabled node.

Unlike outsider attacks, insider attacks are performed by compromised nodes in the WSN. With node compromise, an adversary can perform an insider attack. In contrast to a disabled node, a compromised node seeks to disrupt or paralyze the network. A compromised node may be a subverted sensor node or a more powerful device, like a laptop, with more computational power, memory, and powerful radio. It may be running some malicious code and seek to steal secrets from the sensor network or disrupt its normal functions. It may have a radio compatible with sensor

nodes such that it can communicate with the sensor network.

The following describes the possible attacks that can happen in each layer and the countermeasures that can be taken to avoid the attacks (Figure 2.1):

- *Sinkhole Attack*: The malicious node tries to get all the transmitted traffic from a particular area. This attack can result in denial of service (DoS) attack. A *DoS* attack is any event that diminishes or eliminates a networks capability to perform its expected functions.

- *Blackhole and Grayhole Attacks*: These two attacks target the data traffic. In a Blackhole attack, which results from *Selective Forwarding*, the malicious node drops all the data packets which it receives from other nodes. Similarly in a Grayhole attack the malicious node selectively drops the data packets and therefore a Grayhole attack can be harder to notice. In both attacks the data packets can be further analyzed to obtain critical information.

- *Wormhole Attack*: A compromised node captures packets from one location in the network, and "tunnels" them to another malicious node at a distant location. This attack is a particularly severe control attack on the routing function of wireless sensor networks and usually happens with the sinkhole. Wormhole and sinkhole attacks can be hard to detect [27].

- *Sybil Attack*: The adversary presents multiple identities to other nodes in the network. If other nodes are fooled, the data flow will be transmitted through the adversary and the control of substantial fractions of the network system will be at risk [42]. Sinkhole, Blackhole/Grayhole, Wormhole and Sybil attacks

are enforced on application and network layers. In addition to encryption and authentication, secure location and routing countermeasures can be used to avoid these kinds of attacks.

- *Collision Attack*: When there is a packet being transmitted on a channel, adversaries can easily conduct attacks through sending out some packets to disrupt the ongoing transmission.

- *Battery Exhaustion Attack*: Adversaries can attempt to retransmit control packets to normal nodes repeatedly, forcing receivers to acknowledge them. These kinds of abnormal retransmissions could result in the exhaustion of battery resources of the receivers.

- *Sleep Attack*: When a node sleeps (i.e. becomes inactive) for less than it is needed for "required event sensing reliability", it may bias the sensing results. Additionally if a node sleeps more than required, the measurements will not be accurate [46]. Collision, Exhaustion and Sleep Attacks are enforced at the link layer and can be avoided using access control, encryption, authentication and secure power management.

- *Jamming attack*: The most common type of attack in the physical layer is jamming. The malicious node in this case will prevent or interfere with the signals sent and received by the nodes within a WSN. The countermeasure for this type of attack is by frequency hopping where nodes change their communication frequency in a predetermined sequence and the attacker will not be able to figure out this pattern [13].

2.4 Security Considerations in IEEE 802.15.4 Networks

As described in Section 2.3, security services need to be considered in any wireless sensor networks. The security protocol in IEEE 802.15.4 network, provides authentication, confidentiality, integrity, and freshness checks to avoid replay attacks.

To achieve authentication, any device can maintain an Access Control List (ACL) which is a list of trusted nodes from which the device wishes to receive data. Data encryption is done by encryption of Message Authentication Control (MAC) frame payload using the key shared between two devices (aka peers), or among a group of peers. If the key is to be shared between two peers, it is stored with each entry in the ACL list; otherwise, the key is stored as the default key. Thus, the device can make sure that its data can not be read by devices that do not possess the corresponding key. However, device addresses are always transmitted unencrypted, which makes attacks that rely on device identity somewhat easier to launch. Integrity service is applied by appending a Message Integrity Code (MIC) generated from blocks of encrypted message text. This ensures that a frame can not be modified by a receiver device that does not share a key with the sender. Finally, sequential freshness uses a frame counter and key sequence counter to ensure the freshness of the incoming frame and guard against replay attacks.

An application implemented using IEEE 802.15.4 has a choice of different security suites that control the type of security protection by setting appropriate control parameters in the link layer security suite stack. These security suites provide a

range of security options starting from no security to providing encryption and authentication alone and also both together on the communicating traffic. In an IEEE 802.15.4 network, if privacy of data is the only desired requirement, it is achieved by employing Advanced Encryption Standard (AES) in CounTeR (CTR) mode of operation . In CTR mode, counters are encrypted using a block cipher. The output sequence of blocks are then XORed with the plain text to produce the cipher text. If only authentication is desired, it is achieved by employing a Cipher Block Chaining Message Authentication Code (CBC-MAC) mode of authentication. The CBC-MAC mode uses a block cipher with a key K to encrypt input vectors of the block size to output vectors of the block size. In applications where both encryption and authentication are desired, IEEE 802.15.4 networks will employ counter with CBC-MAC (CCM) [57] mode of operation of AES with optional block sizes of 32, 64 and 128 bits. CCM mode combines the CTR mode of encryption with the CBC-MAC mode of authentication. In IEEE 802.15.4 networks, CCM mode uses counter mode by first applying integrity protection both on the message header and data payload and then encrypting the data payload and MAC using AES. The receiver gets the packet and applies decryption using parameters based on the sender's address from its ACL. The insight of combining both methods of encryption and authentication is that the same encryption key can be used for both, provided that the counter values used in the encryption do not collide with the input vectors used in authentication.

A cryptographic MAC is used to authenticate messages. While longer MACs lead to improved resiliency of the code [9], they also make packet size larger and thus take up bandwidth in the network. Applications that require continuous data flow will be

affected more than those that are event-driven, but in either case an optimum balance between security requirements and network throughput must be found.

2.4.1 Attacks on IEEE 802.15.4 Networks

Two very important applications of IEEE 802.15.4 networks are in health care and agriculture. Even with security prevention techniques, some attacks are still possible and they can be critical in some cases. For instance, the application of wireless sensor networks in health care involves patient's security and potentially his/her life. Ignoring little mistakes may result in tragic situations so applying tough security considerations becomes a serious affair. Analyzing attacks in detail will help to create a desired security and intrusion detection mechanism which fits the needs of this kind of networks.

We can classify the threats to an IEEE 802.15.4 network into attacks on ordinary nodes and attacks on the PAN coordinator.

Attacks on nodes: Since a node's hardware is not tamper-proof, it is possible to read the ID and master secret from its ROM and wait for the time when PAN coordinator initiates a new link key update.

Even with small number of corrupted nodes it is possible to attack the fair bandwidth allocation in a WSN due to CSMA-CA type of access. Malicious node(s) can try to access the medium more frequently which will increase the access delay and packet loss for other nodes. This attack can be prevented if the PAN coordinator keeps the track of the number of packets received from each node during some time period and avoid receiving more packets when the number of received packets exceeds

a specified threshold.

Another case is when a node applies some power saving technique and sleeps for random periods of time [41]. Such techniques are applied to extend the lifetime of the network while achieving constant sensing reliability. In that case an attacker can appear as a legitimate node which has woken up while the real node is sleeping. To deal with this attack, the PAN coordinator has to keep track of the average of the sleep periods and isolate the node which wakes up more often than the others.

Attacks on the PAN coordinator: Attacks on the PAN coordinator can be much more harmful than attacks on ordinary nodes since the PAN coordinator receives and forwards data packets and in overall has the control over security and management of the PAN. Some attack scenarios can be described as follows:

1. An adversary may read the contents of the beacon frame as well as the contents of all data frames if they are not encrypted. From the sequence of beacon frames, an adversary may learn node IDs and requested event sensing reliability. It can also learn the period of key exchange. One way of dealing with this problem is to encrypt the data in the beacon with the group key which will prevent passive listening.

2. If the PAN coordinator's hardware is not tamper free, then it can be temporarily stolen and master secrets for the nodes can be read.

3. The location of the PAN coordinator has to be securely reported to the central database. The location of the PAN coordinator can be determined with the collaboration of surrounding PAN coordinators which have Global Positioning

System (GPS) receivers or by using the node's own GPS. The first approach is similar to the location discovery algorithms [33] where the PAN coordinator does not have GPS but it can hear beacon frames from other PAN coordinators with GPSs and determine its relative location with respect to its neighbors based on the power of the received signals. This approach can be attacked by an adversary which has higher powered transmitter. As a result, the PAN coordinator will report measurements from the wrong location which will not be accepted by the central database. Recent work [34, 32] attempts to detect malicious nodes using location detection algorithms, but there is an issue of how harmful this attack can be if it is not detected. If the PAN coordinator has its own GPS but it does not participate in localization algorithms, it is possible that adversary attacks the PAN coordinator and forges location and data.

4. It is possible that two corrupted distant PAN coordinators establish a channel (due to the issue of physical security we assume that this will be wireless channel also). Then, one PAN coordinator can forward the data to the other which can report them as measurements. This is similar to a wormhole attack which was first introduced as an attack on routing algorithms [21, 26]. Since no other PAN coordinators are affected it is very difficult to detect this attack using collaborative algorithms among PAN coordinators. Instead, it is necessary to check the activities on the wireless channels which are not in use by the neighboring PAN coordinators.

2.5 Intrusion Detection

Since network-based computer systems are used in various facets of our lives, they have become the targets of intruders. An adversary's malicious attack may violate integrity, availability and confidentiality of an information system or its data. Intrusion prevention techniques, such as encryption and authentication (e.g., using passwords and biometrics) which are the first line of defense, are usually not sufficient because of the scale and complexity of network systems [20]. Further, preventing attacks from insider nodes in a WSN is very difficult. Therefore, intrusion detection mechanisms are necessary to detect malicious nodes. Intrusion detection and response provide a second line of defense. Given that new vulnerabilities will be discovered and the fact that our adversaries will continue to invent new attack methods, especially for a relatively new technology such as WSNs, we have to use effective detection approaches [22].

2.5.1 Intrusion Detection Techniques

An Intrusion Detection System (IDS) may be classified based on the detection technique used [10]:

- **Misuse or Signature-Based:** A potential intrusion is reported if a sequence of events within a system matches a set of known security policy violations. To detect an intrusion by Misuse a knowledge of potential vulnerabilities of the system should be available. The intrusion detection system then applies this rule set to the sequences of data to determine a possible intrusion. This

technique may exhibit low false positives[1], but does not perform well at detecting previously unknown attacks. Subhadrabandhu et al. [54] present a robust intrusion detection system using misuse detection technique. Anjum et al. [8] deal with the ability of various routing protocols to facilitate intrusion detection techniques when the attack signatures are completely known in the network.

- **Anomaly:** A set of expected values are used to compare with the system's behavior. If the computed statistics do not match the expected values, an anomaly is reported. Anomaly-based detection defines a profile of normal behavior and classifies any deviation of that profile as an intrusion. The normal profile is updated as the system learns the system's behavior. This technique may detect previously unknown attacks but may exhibit high false positives. Zhang et al. [62], present an anomaly detection model. They use trace data which describes the normal updates of network routing information since the main concern is that false routing will be used by other nodes. The generated trace data will then bear evidence of normality or anomaly. High false positive rates are reported based on their simulation results.

  Anomaly detection may be used to detect attacks against a network daemon or a SetUID program by building a normal profile of the system calls made during program execution. If the process execution deviates significantly from the established profile, an intrusion is assumed. Okazaki et al. [43] have proposed a lightweight approach using profiles consisting of the type of system call and its frequency of occurrence, in which speech recognition methods are used to

[1] A false positive is when an event is mistakenly marked as intrusion.

calculate the optimal match between a normal profile and a sample profile.

- **Specification:** Compared to the Misuse modeling, specification modeling takes the opposite approach; it looks for a specification of how a system or program executes and marks a sequence of instructions as a potential intrusion if it violates the specification. This technique may provide the capability to detect previously unknown attacks, while exhibiting a low false positive rate. For example, Snort [50] is an open source network intrusion prevention and detection system utilizing a rule-driven language, which combines the benefits of signature-based and anomaly-based detection methods.

A second distinction can be made in terms of the placement of the IDS. In this respect IDSs are usually divided into host-based and network-based systems and once again, both systems offer the advantages and disadvantages:

- Host-based systems are present on each host that requires monitoring, and collect data concerning the operation of this host, usually log files, network traffic to and from the host, or information on processes running on the host. Host-based systems are able to determine if an attempted attack was indeed successful, and can detect local attacks, privilege escalation attacks and attacks which are encrypted. However, such systems can be difficult to deploy and manage, especially when the number of hosts needing protection is large. Furthermore, these systems are unable to detect attacks against multiple targets within the network.

- Network-based IDSs monitor the network traffic on the network containing the

hosts to be protected, and are usually run on a separate machine termed a sensor. Network-based systems are able to monitor a large number of hosts with relatively little deployment costs, and are able to identify attacks on and from multiple hosts. However, they are unable to detect whether an attempted attack was indeed successful, and are unable to deal with local or encrypted attacks.

Hybrid systems, which incorporate host- and network-based elements can offer the best protective capabilities, and systems to protect against attacks from multiple sources are also under development.

2.5.2 Current Intrusion Detection Solutions for WSNs

In this section, I review some of the systems and algorithms which have been proposed for intrusion detection in WSNs. Some general approaches are presented as well as the algorithms which are based on Markov models. Some other solutions utilize mobile agents in order to detect and respond to intrusions.

General Approaches Silva et al. [12] propose a decentralized intrusion detection system for WSNs. The IDS component is loaded into some nodes called "monitor" nodes. The detection system is specification-based, since the WSN's behavior may vary depending on the application goal. When deploying the sensor network, monitor nodes are distributed all over the network so that every node is covered by at least one monitor node. Their algorithm consists of three phases: in phase one which is the data acquisition phase, messages are collected in a promiscuous mode and the

important information is filtered before being stored for subsequent analysis. In the processing phase, the intrusion detection rules are applied to the stored data. Finally the last phase or intrusion detection phase will determine if an intrusion detection is raised.

Mittal and Vigne [38] describe a signature-based intrusion detection technique which is for detecting routing-based attacks. Detecting these kinds of attacks is difficult because malicious routing behavior can be identified only in specific network locations. They use the characteristics of the Routing Information Protocol (RIP) [18], the network topology, and the positioning of the intrusion detection sensors to automatically determine both the signature configuration of the sensors and the messages that the sensors have to exchange to detect attacks against the routing infrastructure. The approach uses a set of sensors that analyze routing traffic in different locations within a network. An algorithm to automatically generate both the detection signatures and the inter-sensor messages needed to verify the state of the routing infrastructure has been devised for the case of the RIP distance-vector routing protocol.

Du et al. [16] propose a general Localization Anomaly Detection (LAD) scheme. They consider the fact that some anomalies happen in the process of location discovery (localization). For instance deploying GPS in every sensor to determine the location of the sensors, is costly. A number of solutions consider deploying GPS to just a few numbers of nodes (aka, beacon nodes) in the network. The remaining nodes will verify their location using the location of sensors with GPS. We can see that this approach may result in localization anomalies if adversaries compromise beacon

nodes and send incorrect reference information about locations. The proposed scheme (LAD), takes advantage of the deployment knowledge and the group membership of its neighbors, and uses such knowledge to find out whether the estimated location is consistent with observations. If they are inconsistent, LAD will report an anomaly. Du et al. formulate the problem as an anomaly intrusion detection problem, and introduce a localization anomaly detection phase after the localization phase. In the localization phase, sensors derive their locations. Then in the detection phase, sensors verify whether the derived locations are correct or not. A failure of the verification indicates an anomaly.

Although the idea of using monitor nodes, GPS on some nodes, and intrusion detection sensors is appealing, the energy consumption of these nodes will be considerably more in comparison to that of ordinary sensing nodes in the network. Therefore, the batteries in the monitor and intrusion detection nodes will die faster and need to be changed. Since the majority of the nodes will still have battery left, changing the batteries of only a few nodes is costly or even impossible because of deployment of these nodes in hostile environments and all the sensor nodes need to be redistributed. The network will also be vulnerable to battery attack.

Onat and Miri [44] introduce a novel anomaly-based intrusion detection method for wireless sensor networks suited to their simple and resource-limited nature. Intrusion detection functions are distributed to all the nodes in the network. This detection-based security scheme, which is for large scale sensor networks, exploits network stability in its neighborhood information. In many attacks against sensor networks, the first step for an attacker is to establish itself as a legitimate node within the

network. If each node can build a simple statistical model of its neighbors behavior, these statistics can later be used to detect changes in them. The authors have shown that, by looking at a relatively small number of received packet features, a node can effectively identify an intruder impersonating a legitimate neighbor.

Markov-Based Approaches Doumit and Agrawal [15] propose an anomaly detection approach based on Self-Organized Criticality (SOC) which is meant to link the multitude of complex phenomena observed in nature to simplistic physical laws and or one underlying process. Hidden Markov models are used to detect data inconsistencies. This approach is developed based on the structure of naturally occurring events. With the acquired knowledge derived from the self-organized criticality aspect of the deployment region, a hidden Markov model is then applied. The proposed approach lets a sensor network adapt to the norm of the dynamics in its natural surrounding so that any unusual activities can be singled out. The work is focused on the fact that sensor nodes in WSNs are limited in resources and tries to minimize the resource consumption.

A new technique for handling security in WSNs is presented by Agah et al. [4]. They formulate the attack-defense problem by game theory and use a Markov Decision Process to predict the most vulnerable sensor nodes. Their approach formulates attack-defense problem as a two-player, non-zero-sum, non-cooperative game between an attacker and a sensor network. In a non cooperative game unlike cooperative ones, no outside authority assures that players stick to the same predetermined rules, and binding agreements are not feasible. Each player (attacker and sensor network) tries to maximize its own payoff. The sensor network tries to defend the sensor nodes

against intrusions. The algorithm is nonzero-sum in the sense that the increase in one players payoff implies the decrease in the other players payoff. The work by Agah et al. shows that this game achieves Nash equilibrium and thus leads to a defense strategy for the network. Then, it uses Markov Decision Process to predict the most vulnerable sensor node.

The anomaly-based detection algorithms using Markov processes and game theory is promising and a wide variety of attacks can be detected using these techniques but the false positive rate is high.

Mobile Agents One solution to perform distributed intrusion detection is by using mobile agent technology [30]. Agents can be seen as guards which protect a network by moving from host to host performing random sampling to be used for detection of anomalous activities or attacks. Instead of monitoring each host at all times, agents only visit machines from time to time to conduct their examinations. When any anomaly is detected, a more comprehensive search is initiated. Although the idea of patrolling guards seems appealing at first, this approach has the disadvantage of leaving hosts vulnerable while no agents are present. On the other hand, random sampling definitely reduces the average computational load at each machine.

Kachirski and Guha [25] have proposed a distributed intrusion detection system based on mobile agent technology. By efficiently merging audit data from multiple network sensors, their scheme analyzes the entire network for intrusions at multiple levels. There are three major agent categories: monitoring, decision-making, and action agents. Some are present on all mobile hosts, while others are distributed to only selected groups of nodes. The monitoring agents look for suspicious activities on

the host node they are currently executing on. If some anomalous activity is detected, the node is reported to the decision agent of the same cluster. The decision agent, then based on these reports, will decide whether the node has been compromised. When a certain level of threat is reached for the node in question, the decision agent dispatches a command that an action must be undertaken by local agents on the node to respond to the intrusion. The response can include logging of the intrusion, real-time notification, disabling the compromised node or even disabling the entire network.

Mobile agents introduce some advantages such as, reducing network load and overcoming network latency and scalability. On the other hand they may also result in some problems like, securing the agent itself and large code size.

In my work, attack signatures are used to detect attacks against the network to evaluate the performance of the network with presence of intrusion detection system. Therefore the detection technique will be signature-based. Also the intrusion detection functionality will be distributed all over the network to avoid uneven consumption of energy so the intrusion detection system is distributed in this sense.

Chapter 3

IEEE 802.15.4 Cluster: Security, Power Management, and Cluster Interconnection

The IEEE 802.15.4 specification provides basic security mechanisms but these security features can not work at their own. The level of security in any network revolves around the keys that are shared among devices. Different approaches have been suggested to distribute and manage these keys. Since IEEE 802.15.4 does not suggest any keying mechanism, in this thesis, I will follow the keying mechanism from Zigbee alliance specifications [6]. In this chapter, I will first introduce the key establishment mechanisms and later explain how this is handled in Zigbee specification by taking advantage of the inherent security mechanisms already provided by IEEE 802.15.4. Later, I will give some detailed description of activity management in an IEEE 802.15.4 cluster followed by a section on methods of cluster interconnection in

such cluster. I will explain why it is important to use cluster interconnection.

3.1 Key Establishment

As explained above, the IEEE 802.15.4 addresses good security mechanisms but it still does not address what type of key establishment mechanism will be used to employ above techniques.

Zigbee alliance [6] is an association of companies working together to enable wireless networked monitoring and control products based on IEEE 802.15.4 standard. After the acceptance of 802.15.4 as IEEE standard, Zigbee alliance is mainly focused on developing network and Application layer issues. Zigbee alliance is also working on Application Programming Interfaces (API) at network and link layer of IEEE 802.15.4. Alliance also introduces secure data transmission in wireless sensor network that are based on IEEE 802.15.4 specification but most of this work is in general theoretical descriptions of security protocol at network layer. There is no specific study or results published by Zigbee alliance in regards to which security suite perform better in different application overheads.

Zigbee alliance has also recommended both symmetric and asymmetric key exchange protocols for different networking layers. Asymmetric key exchange protocols that mainly rely on public key cryptography are computationally intensive and their feasibility in wireless sensor networks is only possible with devices that are resource rich both in computation and power.

Application support sub-layer of ZigBee specification provides the mechanism by which a Zigbee device may derive a shared secret key (Link Key) with another ZigBee

device. Key establishment involves two entities, an initiator device and a responder device and is prefaced by a trust provisioning step. Trust information (e.g. MASTER key) provides a starting point for establishing a link key and can be provisioned in-band or out-band.

Zigbee alliance uses Symmetric-Key Key Establishment (SKKE) protocol for link key establishment. In SKKE an initiator device establishes a link key with a responder device using a master key. This master key, for example, may be pre-installed during manufacturing, may be installed by a trust center, or may be based on user-entered data (PIN, password). In current study we assume that all the devices and PAN coordinator have pre-installed Master keys and we will focus mainly on Link key establishment.

3.2 Keyed Hash function for Message Authentication

A hash function is a way of creating a small digital fingerprint of any data. Cryptographic hash function is a one-way operation and there is no practical way to calculate a particular data input that will result in a desired hash value thus is difficult to forge. A practical motivation for constructing hash functions from block ciphers is that if an efficient implementation of block cipher is already available within a system (either in hardware or in software), then using it as the central component for a hash function may provide latter functionality at little additional cost. IEEE 802.15.4 protocol supports a well known block cipher AES and hence Zigbee Alliance specification also

relied on AES. Zigbee alliance suggested the use of Matyas-Meyer-Oseas [35] as the cryptographic hash function that will be based on AES with a block size of 128 bits.

Mechanisms that provide integrity checks based on a secret key are usually called Message Authentication Codes (MACs). Typically, message authentication codes are used between two parties that share a secret key in order to authenticate information transmitted between these parties. Zigbee alliance specification suggests the keyed hash message authentication code (HMAC) as specified in the FIPS Pub 198 [2].

A Message Authentication code or MAC takes a message and a secret key and generates a $MACtag$, such that it is difficult for an attacker to generate a valid (message, tag) pair and are used to prevent attackers forging messages. In this thesis, the calculation of $MACTag$ (i.e HMAC) of data $MACData$ under key $MACKey$ will be shown as follows

$$MACTag = MAC_{MACKey}MACData$$

3.3 Symmetric-Key Key Establishment (SKKE) Protocol

Key establishment involves two entities, an initiator device and a responder device, and is prefaced by a trust-provisioning step. Trust information (e.g., a master key) provides a starting point for establishing a link key and can be provisioned in-band or out-band. In the following explanation of the protocol we assume unique identifiers for initiator device's as U and for Responder Device (PAN Coordinator) as V. The

master key shared among both devices is represented as $Mkey$.

We will divide Symmetric-Key Key Establishment Protocol (SKKE) between initiator and responder in following major steps.

3.3.1 Exchange of ephemeral data

Figure 3.1 illustrates the exchange of the ephemeral data where the initiator device U will generate the Challenge QEU. QEU is a statistically unique and unpredictable bit string of length *challengelen* by either using a random or pseudorandom string for a challenge *Domain* D. The challenge domain D defines the minimum and maximum length of the Challenge.

$$D = (minchallengeLen, maxchallengeLen)$$

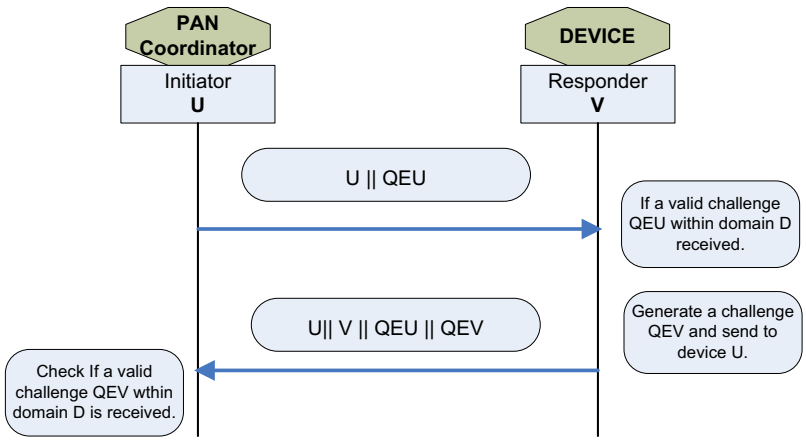


Figure 3.1: Exchange of ephemeral data. Adopted from [29]

Initiator device U will send the Challenge QEU to responder device which upon

receipt will validate the Challenge QEU by computing the bit-length of bit string Challenge QEU as $Challengelen$ and verify that

$$Challengelen \in [minchallengelen, maxchallengelen]$$

Once the validation is successful the Responder device will also generate a Challenge QEV and send it to initiator device U. The initiator will also validate the Challenge QEV as described above.

3.3.2 Generation of shared secret

Both parties involved in the protocol will generate a shared secret based on unique identifiers (i.e. distinguished names for each parties involved), symmetric master keys and Challenges received and owned by each party (Figure 3.2).

1. Each party will generate a $MACData$ by appending their identifiers and respective valid $Challenges$ together as follows

   $$MACData = U||V||QEU||QEV$$

2. Each party will calculate the $MACTag$ (i.e Keyed hash) for $MACData$ using $Mkey$ (Master Key for the device) as the key for keyed hash function as follows.

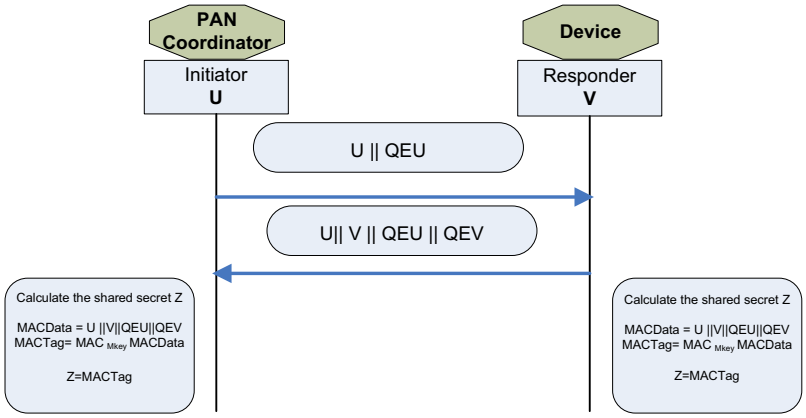


Figure 3.2: Generation of shared Secret. Adopted from [29]

$$MACTag = MAC_{Mkey}MACData$$

3. Now both parties involved have derived same secret Z

   (note: This is just a shared secret not the Link key. This Shared secret will be involved in deriving the link key but is not the link key itself.)

$$Z = MACTag$$

3.3.3 Derivation of link key

Each party involved will generate two cryptographic hashes (this is not keyed hash) of the shared secret as described in ANSI X9.63-2001 [1].

$$Hash_1 = H(Z||01)$$

$$Hash_2 = H(Z||02)$$

The hash value $Hash_2$ will be Link key among two devices (Figure 3.3). Now for confirming that both parties have reached on same Link key ($KeyData = Hash_2$) we will use value $Hash_1$, as key for generating Keyed hash values for confirming stage of the protocol.

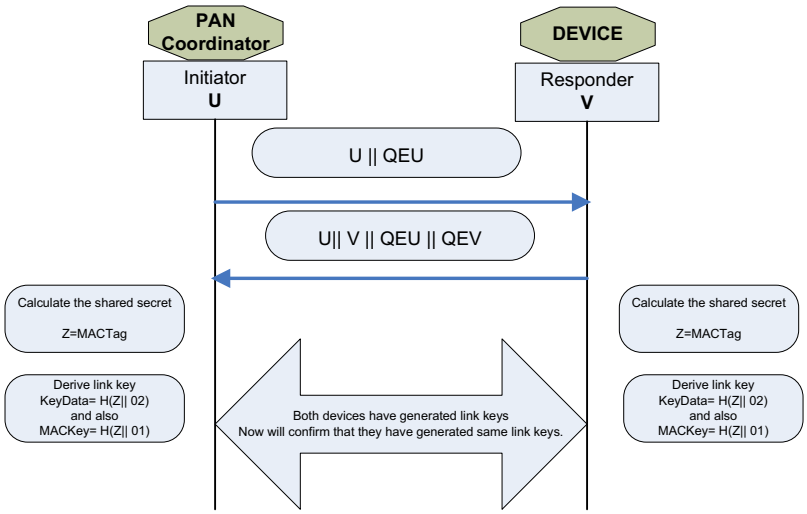


Figure 3.3: Generation of Link Key. Adopted from [39]

$$MACKey = Hash_1 \tag{3.1}$$

$$KeyData = Hash_2 \tag{3.2}$$

$$KKeyData = Hash_1||Hash_2 \tag{3.3}$$

3.3.4 Confirming Link key

Till this stage of protocol both parties are generating the same values and now they want to make sure that they reached on same Link key values but they do not want to exchange the actual key at all. For this they will once again rely on keyed hash functions and now both devices will generate different $MACTags$ based on different Data values but will use same key (i.e. $MACKey$) for generating the keyed hashes ($MACTags$).

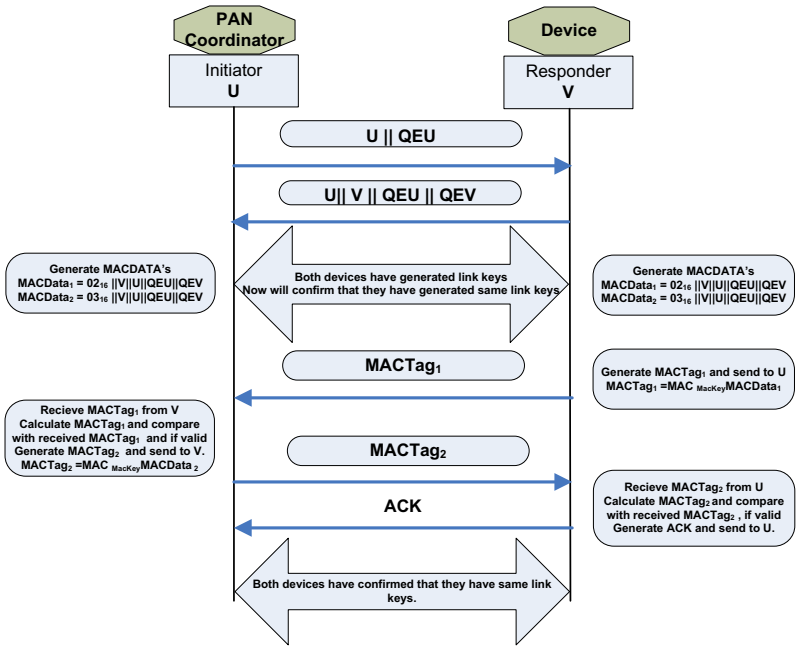


Figure 3.4: Confirmation of Link Keys. Adopted from [29]

1. Generation of MACTags

   Initiator and responder devices will first generate $MACData$ values and based on these values will generate $MACTags$. Initiator device D will receive the

$MACTag_1$ from the responder device V and generate $MACTag_2$ and send to device V.

We explain the generation of both $MACData$ values and $MACTags$ as follows

First both devices will calculate $MACData$ values

$$MACData_1 = 02_{16}||V||U||QEU||QEV$$

$$MACData_2 = 03_{16}||V||U||QEU||QEV$$

From the above $MACData$ values both devices will generate the $MACTags$ using the key $MACkey$ (Equation 3.1) as follows

$$MACTag_1 = MAC_{MacKey}MACData_1$$

$$MACTag_2 = MAC_{MacKey}MACData_2$$

2. Confirmation of MACTags

Now the initiator device D will receive $MACTag_1$ from responder and Responder device V will receive $MACTag_2$ from device D and both will verify that the received $MACTags$ are equal to corresponding calculated $MACTags$ by each device. Now if this verification is successful each device knows that the other device has computed the correct link key (Figure 3.4).

3.4 Simulating SKKE

I will incorporate SKKE's four major communication steps in my simulations as are described in ZibBee specification [6] (Figure 3.5).

SKKE-1

Initiator U will send the Challenge QEU and wait for the Challenge QEV from responder V.

SKKE-2

Responder V will receive the Challenge QEU from initiator U, calculates its QEV and in the same data packet will send the $MACTag_1$.

SKKE-3

Initiator will verify the $MACTag_1$ and if it is verified successfully, will send its $MACTag_2$. Now the initiator has a Link key but will wait for an acknowledgment that its $MACTag_2$ has been validated by the Responder V.

SKKE-4

Responder will receive and validate the $MACTag_2$ from the Initiator. If $MACTag_2$ validated successfully, the responder will send an acknowledgment and now both Initiator and Responder have Link keys. Once initiator receives this SKKE-4 message, keys establishment is complete and now regular secure communication can proceed using Link key among the initiator and the responder.

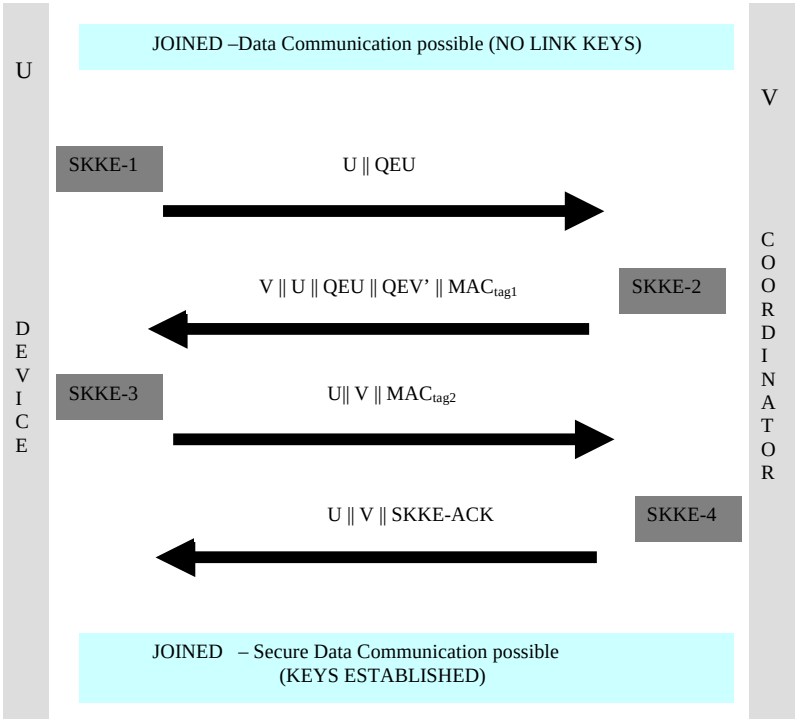


Figure 3.5: SKKE protocol. Adopted from [29]

3.5 Link Key Update

Key management is the set of techniques and procedures supporting the establishment and maintenance of keying relations between authorized parties. Key management is simplest when all keys are fixed for all time. The time period over which these keys are valid for use is limited because use of same key may result in giving enough information relating to a specific key for cryptanalysis and also may expose network traffic in case of compromise of single key.

Depending on the severity of the threat environment, it is possible that a node or Link key is some how compromised by an adversary and can send false data to the

PAN coordinator. Key update provides an automated mechanism for restricting the amount of data which may be exposed when a Link key is compromised. Sound security policies regarding transparent key updates is fundamental component of sound security practices. But key updates protocol depends on the key update overheads and threat environment under which network is working. Hence controlling the life time of keys and determination of how the key update occurs is a challenging task in any network. Approaches for key updates in general wireless networks mainly target network that have group key structures and have high communication bandwidth [56, 61]. For resource scarce IEEE 802.15.4 networks these key updates will effect the performance adversely.

In this work we assume that PAN coordinator maintains a counter for each node that keeps track of the number of packets exchanged under the same key (Figure 3.5). When the threshold value of the counter is reached for any device, the PAN coordinator will initiate the key exchange with all the devices in the cluster. During the key exchange, all devices will temporarily stop the data transmission and resume it when they acknowledge the new key. Alternative approach will be to use the single counter for all the devices. However, this approach may open the security hole for denial of service attack by single corrupted device.

3.6 Power Management

Power management consists of adjusting the frequency and ratio of active and inactive periods of sensor nodes [53, 48]. For IEEE 802.15.4 nodes it can be implemented in two ways. In the first one, supported by the standard [6], the interval

between the two beacons is divided into active and inactive parts, and the sensors can switch to low-power mode during the inactive period. Activity management for individual nodes can be accomplished through scheduling of their active and inactive periods.

Let us consider a sensing application in which redundant sensors are used to achieve the desired value of event sensing reliability (number of packets persecond needed for reliable event detection) [48]. We assume that individual nodes sleep for a random time interval, the duration of which is a geometrically distributed random variable regulated with probability P_{sleep}. When a node wakes up, it waits for the beacon from the coordinator before it attempts to transmit the packet. We have used Bernoulli scheduling for the packet scheduling during the active period of the node. In this approach, at the end of each packet transmission the node checks its uplink buffer. If it is empty, the node immediately goes to sleep; if there are packets to send, the node transmits the next packet from the buffer with the probability P_{active}, or goes to sleep with the probability $1 - P_{active}$. Therefore, two control parameters are needed: one, P_{sleep}, regulates the duration of the inactive period; the other, P_{active}, regulates the duration of the active period. when individual nodes begin to cease functioning, either because of battery exhaustion or for other reasons, the remaining nodes will have to extend their activity to achieve satisfactory reliability, and the importance of the Bernoulli mechanism will increase.

Depending on whether we split the computational load of activity management, we can have two approaches of centralized and distributed controls. By choosing the later approach to distribute the computational load more evenly, we assume that

the network coordinator is aware of the number of sensor nodes (which have to be explicitly admitted to the network [3]) and their packet arrival rates (which may be obtained as simple long-term averages, as packet headers contain the source node address). The coordinator first determines node utilization based on the number of live nodes and then calculates the individual reliability r per node (by dividing the required collective reliability R by the number of live nodes n) and sends this information within the beacon frame. Over time some sensors die, and the coordinator has to broadcast updated values of individual reliability, which grow whenever one of the sensors die. Note that the sleep time is geometrically distributed, and the mean sleep time is $t_{boff}/(1 - P_{sleep}) = 1/r$ where $t_{boff} = 0.32ms$ corresponds to the duration of one backoff period. Therefore, each sensor node starts with $Psleep = 1 - rt_{boff}$ and $Pactive = 0$. It then monitors the utilization of its radio transmitter/receiver subsystem, using a monitoring window of specified size. Utilization is simply calculated as the count of backoff periods in which the node was active during the recent window divided by the total size of the window.

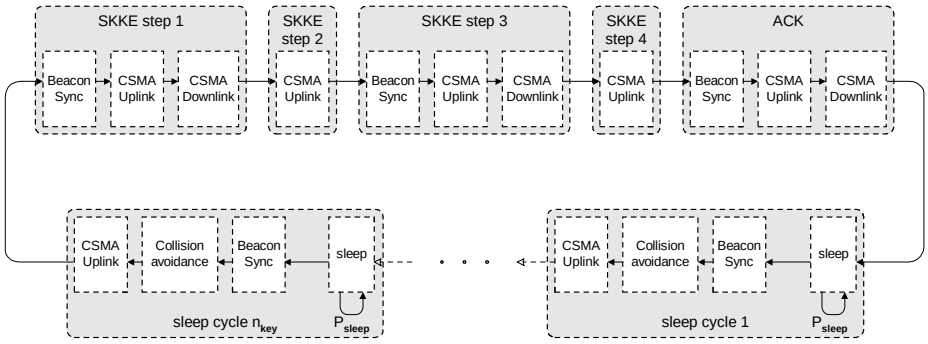


Figure 3.6: Markov chain for the node behavior under threshold triggered key exchange. Adopted from [29]

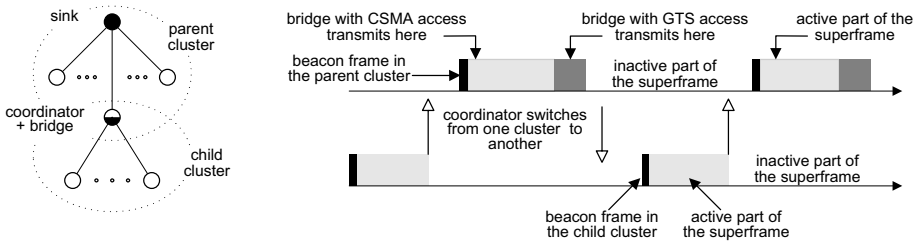
In [7], we have developed Markov chain model for node behavior which includes all

phases of SKKE protocol and subsequent sleep and transmission phases. We assume that PAN coordinator maintains a separate counter for the number of transmissions by each node. When the counter value reaches threshold n_k, key update protocol is triggered. Updated keys are used to generate Message Authentication Code. The high level Markov chain which includes key update, sleep periods followed by the transmissions is presented in Figure 3.6.

3.7 Cluster Interconnection

To provide scalability, clustering can be used. By creating multiclusters network, different services can be distributed to ease the management and control within each cluster. For instance the performance of CSMA-CA algorithm used in IEEE 802.15.4 networks is shown to deteriorate by the rapid increase of the number of nodes. We define a cluster as collection of several nodes/devices and a PAN coordinator. As a solution to the problem of providing interconnection between different clusters, bridging scheme has been introduced[37]. A *bridge* shares its time between the child cluster (where it acts as the coordinator) and the parent cluster (where it is just an ordinary node).

As mentioned before the superframe may also contain an optional inactive period, in which individual nodes are free to engage in other activities or simply shut down in order to conserve energy. The presence of an inactive period facilitates the creation of multicluster networks, since a bridge may use the inactive period to switch to the other cluster. Figure 3.7 shows the network topology of master-slave bridging scheme and bridge operation for two clusters with the same superframe duration.



(a) Network topology.

(b) Bridge operation.

Figure 3.7: Bridging scheme for two clusters. Adopted from [37]

In this scheme, the coordinator/bridge has to buffer all the data it receives from the sensor nodes in the child cluster (i.e., source cluster) before it can upload it to the coordinator of the parent cluster (i.e., the network sink). The size of that buffer is a critical parameter since small buffers may lead to excessive packet blocking, which will in turn affect the quality of information received by the sensing application.

It should be noted that the bridge acts just like an ordinary node in the parent cluster. This means that a countdown that exceeds the active portion of the parent cluster superframe will be frozen during the inactive portion of the said superframe, and resumed in the next one. It also means that the bridge will compete for access to the medium with ordinary nodes in the parent cluster. As the bridge traffic originates from the entire child cluster, there is potentially significant contention in the parent cluster.

Chapter 4

Intrusion Detection in IEEE 802.15.4 Cluster

As explained in Section 2.4.1, attacks to an IEEE 802.15.4 cluster can be classified into attacks on ordinary nodes and attacks the PAN coordinator. The PAN coordinator usually has high computational and communication power. Also in most applications tough security policies are in place to protect the PAN coordinator. The sensor devices, on the other hand, are restricted in such resources and therefore more likely to be under attack by the adversaries.

Since sensor nodes' hardware is not completely tamper proof, an intruder is able to capture a device and get access to the ID and keys stored in the memory. This gives control to the intruder over the sensor and opens numerous threarts to the network as a result. By targeting essential functions and mechanisms exploited by an IEEE 802.15.4 network such as sensing, forwarding critical information packets, and activity management, these kinds of attacks can be harmful if they are not detected

and responded to. In this chapter, I will describe three major attacks threating an IEEE 802.15.4 network: Sleep Deprivation attack, False Data Injection attack, Sybil, and node displacement attack. I will also propose possible detection methods and explain them in detail.

4.1 Sleep Deprivation Attack and Detection

4.1.1 Sleep Deprivation Attack

Although utilization of power management mechanism while dealing with limited battery supplied sensors helps to extend their lifetime ([7]), it also introduces a novel malicious activity. The adversaries can capture nodes since they are not completely tamper resistant in most cases and access the data, and the code inside compromised nodes. By reprogramming the compromised nodes, they can launch attacks in different layers (Section 2.3.3) against the network. Focusing on the approach taken in this work for power management (i.e. the ability of sensor nodes to enter a low power mode - sleeping), adversaries can conduct "Sleep Deprivation" attack first introduced by F. Stajano [52, 51].

Sensor node under sleep deprivation attack will stay out of the sleep mode and this will lead to the exhaustion of its battery much sooner than expected. Decrease in the lifetime of some sensors under this attack effects the lifetime of the network as whole. Apart from the effects of this attack on network's lifetime, it may also lead to biased average sensed results because the captured sensor nodes will have more chances of transmitting data packets.

Sleep deprivation attack can be avoided using access control, encryption, authentication, and secure power management. However, as discussed earlier, no security measure can make a sensor network completely safe. The adversary can capture keys after compromising the sensor node, wait for the next key update, and start communicating with the PAN coordinator as a legitimate node. Detection methods are needed to avoid further damage to the network and keep the network working for as long as possible [46].

4.1.2 Proposed Detection Solutions

Considering the details of network on which we have applied our security measures as well as the power management method, three solutions can be considered for the detection of sleep deprivation attack.

- **Lifetime Observation**: By keeping the track of sensor nodes' power consumption, we are able to develop a detection method based on the remaining lifetime of the network in overall. The sensor node can be in one of the four states: *transmit*, *receive*, *idle*, and *sleep*. When node wakes up and has a packet to transmit it turns its receiver on in order to synchronize to the beacon, node turns the transmitter on and starts backoff count in order to transmit the packet. After packet transmission, node turns the receiver on in order to receive acknowledgment and then starts the new sleep period. Most of the power management schemes try to minimize the idle state in which the sensor node does not receive or transmit.

  Table 4.1 illustrates the power consumption in each state derived from typi-

Table 4.1: Sensor's Power Consumption per one backoff period derived from specification of tmote_sky module.

Mode	**Param**	**value**	**CPU State**	**Radio State**
Transmit	ω_t	$15.8\mu J$	On	Transmitting
Receive	ω_r	$17.9\mu J$	On	Receiving
Sleep	ω_s	$18.2nJ$	Sleep Mode	Off

cal operating conditions reported in documentation for Ultra low power IEEE 802.15.4 sensor module tmote_sky [11] operating in ISM band with raw rate 250kbps. According to the specification of tmote_sky module, two AA batteries are needed in order to supply voltage between 2.1-3.6 V.

Choosing to trigger Sleep deprivation attack by tracking the changes in the remaining lifetime of sensor nodes is definitely a good approach in detecting the attacks that cause huge power consumption and lead to considerable degradation of lifetime. However, some adversaries mostly concentrate on biasing the sensed data and not necessarily take a lot of CPU time with large computation jobs. These attacks are harder and in some cases impossible to detect using lifetime observation approach.

- **Data Packet Count**: The number of packets sent by sensors can be another parameter by which the observation shows a distinguished pattern of change while there are some captured nodes under the sleep deprivation attack in the network. In the case of biasing the results using this attack, the adversary may try to send more packets and satisfy the requested event sensing reliability from the PAN coordinator. The number of data packets sent by adversary sensor nodes will get noticeably higher in most cases.

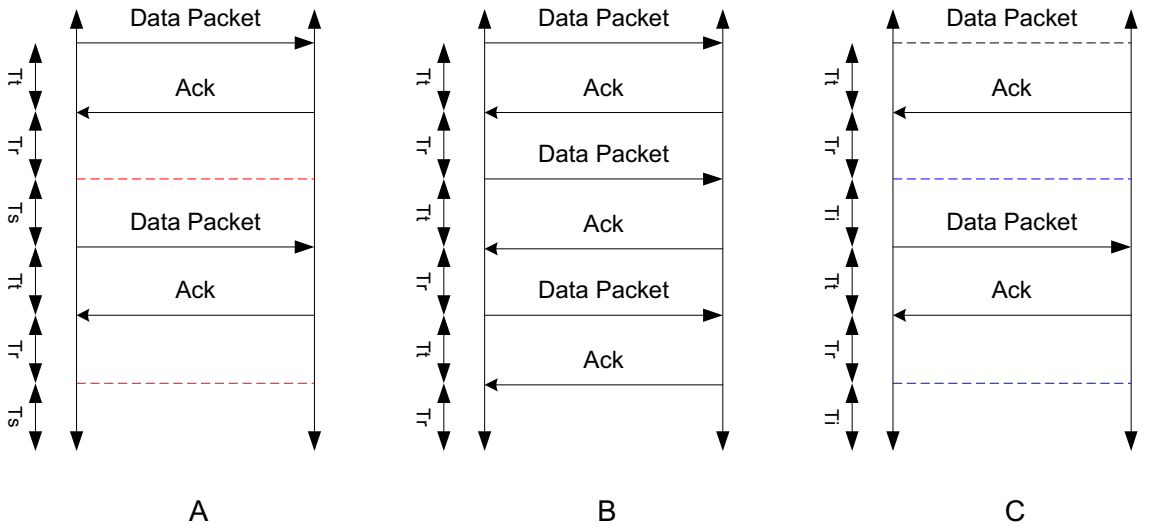


Figure 4.1: Time spent on transmitting (Tt), receiving (Tr), sleeping (Ts), and being idle (Ti) by A: a non-adversary sensor node, B: An adversary sensor node with the intension of biasing sensing data received by PAN coordinator, C: An adversary sensor node with the intension of shortening network's lifetime.

A detection system can be put in place to trace the number of packets received by the PAN coordinator from each sensor node and compare them with a predefined threshold of expected number. The downside of this method is that attacks are harder or sometimes impossible to detect when the adversary node does not send suspicious number of data packets and only eats up the power supply of the sensor by using too much of CPU time.

- **Sleep Observation**: No matter what kind of sleep deprivation attack we deal with, one thing they all have in common is that by all means they make the adversary sensor node kept out of the low power mode. Figure 4.1 shows the difference between the activities of an adversary sensor node under sleep deprivation attack and a non-adversary sensor node.

The common characteristic between the two different versions of the sleep deprivation attack is that they both eliminate the sleep state from the sensor node's activity. Consequently sleep observation method remains the best option to build the detection system upon since it can detect both of the aforementioned attacks (i.e. biasing sensing data and shortening network's lifetime by eating up battery) successfully. In this work, I will focus on this approach for intrusion detection purposes but will consider other parameters as well.

4.1.3 Detection Algorithm

As mentioned earlier, I will use sleep observation method to create the detection algorithm. In my simulated model, each device stays active (i.e. not in sleep mode) if there is a packet to send or receive. Thus, each time the device wakes up, checks its buffer and if the buffer has queued a packet, it will turn its transmitter on and send the packet. Devices will also remain active or if they were sleep, wake up in case they see there is a packet waiting for them and they need to send a request.

When the devices need to go to the low power mode, probability of sleep (p_sleep) will be calculated based on the required event sensing reliability announced by the PAN coordinator in each beacon. The resulted p_sleep will then be used as an input to a Geometric Distribution generator to calculate the sleep period for which the device will be in low power mode. Obviously a compromised device will refuse to go to sleep at this point. In the simulated model, the compromised nodes will go to transmit/receive mode if there is a packet to send/receive and if the buffer is empty, they will go to idle mode (i.e. not sending or receiving and not sleeping either).

I have developed a detection algorithm by considering the fact that Geometric distribution is used to calculate the sleep period:

$Mean\ Sleep\ Time\ =\ 1/p\_sleep$

$Standard\ Deviation\ =\ \sqrt{(1-p\_sleep)/p\_sleep^2}$

Since the sleep period will fluctuate from node to node due to the randomness of Geometric distribution, we need a mechanism to estimate average sleep period correctly. To estimate average sleep period we will use the following equation upon receiving a sleep period from a node [31]:

$$EstimatedSleepPeriod = (1-\alpha) * EstimatedSleepPeriod + \alpha * SleepPeriod$$

This equation is recursive and the initial value of $EstimatedSleepPeriod$ is selected as $1/p\_sleep$ (i.e. mean of the Geometric distribution). $EstimatedSleepPeriod$ is a weighted average of sample sleep period values. This weighted average puts more weight on recent samples rather than old samples since the more recent samples better reflect the current state in the network. The degree of weighing decrease is expressed as a constant smoothing factor α, a number between 0 and 1. α may be expressed as a percentage, so a smoothing factor of 10 percent is equivalent to $\alpha = 0.1$. Calculating variability of sleep periods, as well as the estimate of sleep period is useful to find an interval in which most of sleep periods for an individual node, fall. We use the following equation to calculate such a deviation:

$$DevSleepPeriod = (1-\beta) * DevSleepPeriod + \beta * \mid SleepPeriod - EstimatedSleepPeriod \mid$$

Similar to the previous equation, this is recursive too and the initial value of $DevSleepPeriod$ is calculated as $\sigma\ =\ \sqrt{(1-p\_sleep)/p\_sleep^2}$ (i.e. the standard

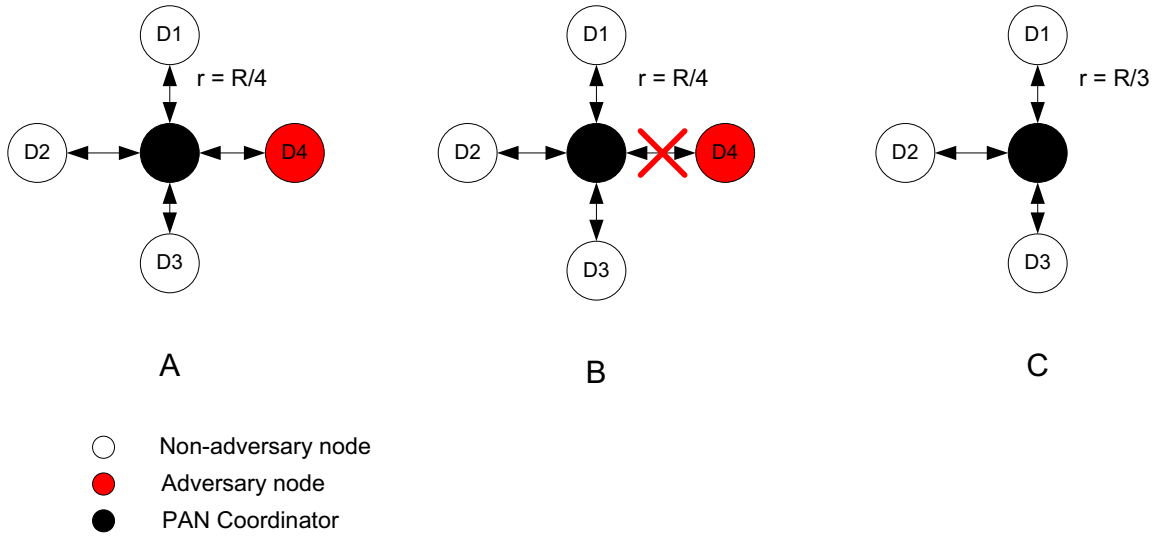


Figure 4.2: Illustration of the attack and response system (A→B). Individual sensing reliability (r) is calculated based on Total required sensing reliability (R) and the number of alive nodes in the network.

deviation of the Geometric distribution). Finally the SleepPeriod interval can be calculated using the following equation:

$$SleepPeriodInterval = SleepPeriodInterval + n * DevSleepPeriod$$

The calculated interval is used as a threshold with which we can compare device's actual sleep period. Any period that deviates with the calculated threshold will be deemed abnormal and the device showing this behavior will be reported as malicious. Further decisions can be made on how to deal with this situation and respond to the attack depending on the specific needs of the network. Currently, in our simulation model, the device will be removed from the network by the PAN Coordinator and new sensing reliability will be announced to the devices (Figure 4.2).

4.2 False Data Injection Attack and Detection

4.2.1 False Data Injection Attack

Wireless sensor networks are prone to node compromise attack where intruders gain the control of sensor nodes. One of the most common consequences of node compromise attack is known to be false data injection attack in which malicious nodes inject false data into the network and distort data integrity (Figure 4.3). This will mislead the base station, cause false alarms, increase battery consumption of sensor nodes and reduce computational and communication recourses. This puts an emphasis on the importance of false data prevention and detection to minimize the resource consumption and to gather correct information at the base station.

Most of the suggested preventive solutions provide node and message authentication to avoid false data injection in sensor networks. However, they all concentrate on outside attacks and do not consider node compromisation within the network. An intruder can easily access all the security information of compromised sensor node. There should be a way to differentiate between correct and false data once it has been delivered.

In [23] authors suggest Commutative Cipher-Based En-Route Filtering scheme (CCEF) based on public key cryptography. In this mechanism, there are two keys, session (between base station and the source node) and witness (for intermediate forwarding nodes) keys per session. Source and intermediate nodes use commutative cipher to verify data packets using witness key. Employing public-key based authentication mechanism for sensor networks to revoke the key of any compromised node

may sound possible but in reality the computation and storage constraints of small sensor nodes make mechanisms based on asymmetric cryptography infeasible.

The mechanism presented in [59] by F. Ye et al is a Statistical En-route Filtering scheme (SDF) in which relaying nodes and base station are able to detect false data with a certain probability. SEF relies on the collective decisions of multiple sensor nodes and their generated report. Again by relying on a collection of nodes, S. Zhu et al [63] introduce the interleaved hop-by-hop authentication scheme that guarantees false data detection by the base station if at least $t + 1$ sensor nodes agree upon a report where t represents the maximum number of compromised nodes. A collection of nodes are responsible for authenticating the report in an interleaved hop-by-hop fashion and forward it to the base station. Authors did not provide any results in terms of actual implementation or simulation of their proposed solution. In both of these works, a collection of nodes in the network are involved in decision making and compromised forwarding nodes are not considered.

Probabilistic Nested Marking (PNM) [60] scheme is a method to fight back against false data injection attack by locating compromised nodes. The authors use packet marking in which a node marks its identity in the packets it forwards. By collecting such marks, the sink can find the route and the origin location of the traffic.

4.2.2 Detection of False Data Injection Attack

In false data injection attack, the intruder has full control over the sensor node under attack and this makes it hard to detect the attack specially in early stages. It is usually the case that data packets reach to the base station before being detected.

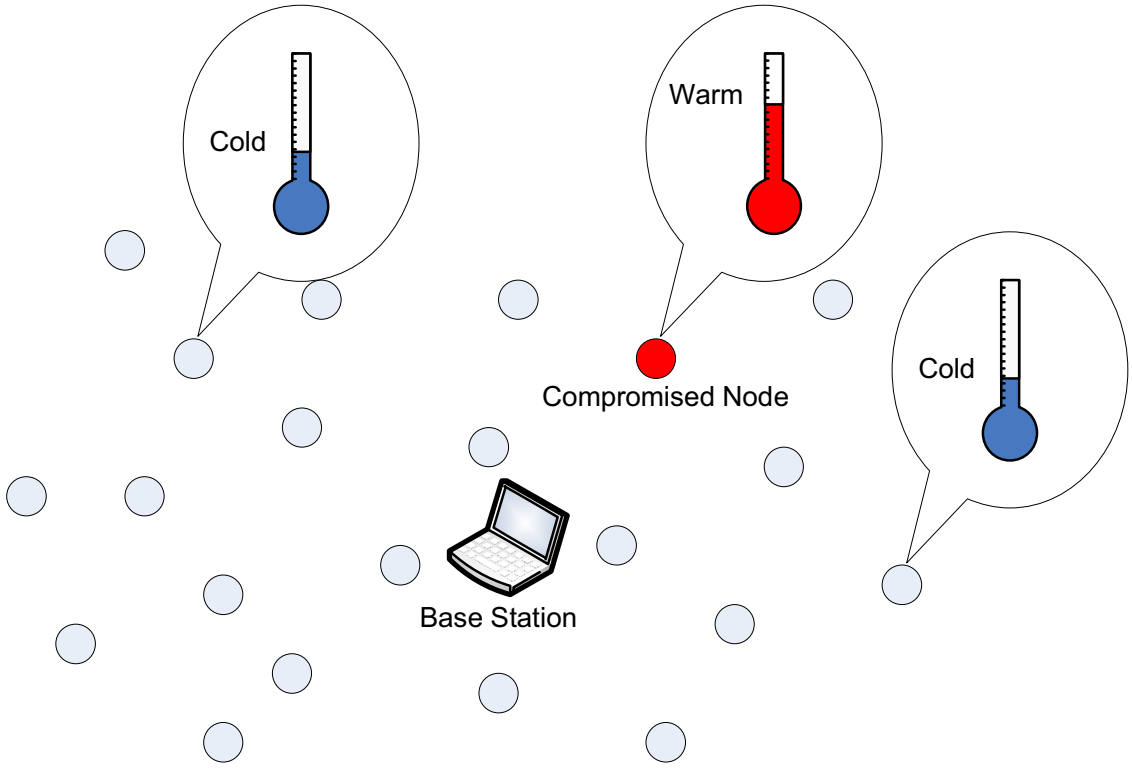


Figure 4.3: A wireless sensor network under False Data Injection attack.

This is specially true when the the network is based on star topology where sensor nodes are 1-hop away from the base station. Figure 4.3 shows a network under false data injection attack.

In this work, I have based my simulation on an IEEE 802.15.4 that follows star topology and hence devices are connected to the PAN coordinator directly. The devices in such a network are responsible for forwarding sensed data packets to the PAN Coordinator using their pairwise keys. An attacker can gain access to the keys of the compromised device and start sending false reports to the PAN Coordinator in a similar fashion. The only trustworthy approach to detect such a false report is

by analyzing the information sent in the data packets. This can be done by the PAN Coordinator since it has enough battery and computational power.

Detection Algorithm

In order to be able to develop a detection mechanism by analyzing the information within data packets, we need to use healthy nodes' (i.e. devices that are not compromised) packets and generate an estimation of what kind of information is to expected and what information is being received. For example referring to Figure 4.3, the estimation of the base station will say that warm temperature is expected and should a compromised device report cold temperature, the base station can catch the intruder and ignore the data packets sent by the compromised device.

To estimate the average, we will use the following equation upon receiving a data packet from a node [31]:

$$Estimate = (1 - \alpha) * Estimate + \alpha * Measurement$$

Estimate is a weighted average of sample measurement values sensed by the devices. This weighted average puts more weight on recent samples rather than old samples since the more recent samples better reflect the current state in the network. The degree of weighing decrease is expressed as a constant smoothing factor α, a number between 0 and 1. α may be expressed as a percentage, so a smoothing factor of 10 percent is equivalent to $\alpha = 0.1$. Calculating variability of measured data, as well as the estimate is useful to find an interval in which most of measurements for an individual node, fall. We use the following equation to calculate such a deviation:

$$Deviation = (1 - \beta) * Deviation + \beta * \mid Measurement - Estimate \mid$$

Finally the measured interval can be calculated using the following equation:

$$Interval = Interval + n * Deviation$$

The calculated interval is used as a threshold with which we can compare sensed measurement values. Any data that deviates with the calculated threshold will be deemed abnormal and the device showing this behavior will be reported as malicious.

4.3 Sybil and Sensor Displacement Attacks and Detection

J. Newsome et al. [42] discuss Sybil attack and possible defense techniques against it. There are few works done to suggest a solution for Sybil attack detection or evaluate a detection mechanism for Sybil attack in WSN and non of them consider star topology. In an star topology all the communications go through the PAN coordinator and therefore, use of neighborhood information is impossible. In [14], authors present a method to detect Sybil attack based on Received Signal's Strength to detect Sybil attack but they do not consider the path loss and fading effects. In this thesis I have simulated an IEEE 802.15.4 network under these two attacks together with a detection mechanism based on Received Signal Strength Indication (RSSI).

4.3.1 Sybil Attack

It is possible for an adversary to present multiple identities to other nodes in the network. The adversary can either fabricate a new identity or steal the identity of a

legitimate node. If the Sybil nodes succeed in fooling the other nodes/PAN Coordinator and start getting data flow, they could cause several malicious behaviors [42]:

- The popular distributed storage system used in sensor networks can be defeated by this attack. For example data fragmentation and replication that are done across several nodes, will include Sybil nodes as a subgroup. This can be specially threatening in applications involving critical data.

- Some network activities (e.g. routing) may need the location of nodes as part of their process. In this case a Sybil node that is capable of being in two locations at the same time, will disrupt this function. This will be more severe in cases where a single Sybil node generates multiple identities.

- The PAN Coordinator in an IEEE 802.15.4 network, performs data aggregation and stores data packets sensed by the devices to be read later. Sufficient number of Sybil nodes could change the average value of sensing data.

- Sybil nodes can disturb power management regime of sensor networks. The required sensing reliability calculated by the PAN coordinator is based on the number of nodes alive (i.e. individual IDs with battery power). This calculation will also consider Sybil nodes with forged IDs and let the non-malicious nodes sleep more than required.

- Some detection mechanisms (e.g. Sleep deprivation detector and false data injection detector) take into account the majority of healthy non-malicious nodes, which send correct sensing data to the PAN coordinator. The Sybil nodes with

several IDs are able to alter the average value of the receiving data and consequently trick the detection system.

4.3.2 Sybil Attack Detection

Sybil nodes present fake identities to the network and try to send and receive packets and jeopardies the secrecy within the network or in some cases simply drop or modify information packets. In order to detect such attack, we need to use some localization method to track good nodes and differentiate between them and Sybil nodes. Using GPSs and guard nodes with correct trusted location is expensive in compares to sensor characteristics in such a network in terms of cost. One of the most effective and feasible solutions in current literature is to detect Sybil nodes using Received Signal Strength Indication (RSSI) [55].

RSSI

Position verification (Localization) is one of the most interesting detection techniques when it comes to Sybil attack. Most of the recent commercially available transceivers use Received Signal Strength Indication (RSSI) to estimate signal power at the receiving antenna [47]. It is generally used for Energy Detection (ED), Clear Channel Assessment (CCA: by comparing the calculated ED parameter over an observation interval with a predefined input power threshold to indicate whether or not the channel is free), and link quality indication (RSSI register values indicate whether the channel is quiet or busy).

The well known IEEE 802.15.4 standard [3] suggests the use of RSSI for the pur-

pose of CSMA/CA performance and link quality estimation. The standard states a receiver sensitivity of -85dBm or better, where 1dBm is equal to 10LOG(Power/1mW), 40dB as the minimum range and 6dB as the linearity within this range.

Different transmitters/receivers in the same environment, may behave differently. Antenna orientation also greatly effects RSSI calculations since different antennas have their own radiation patterns. All these parameters can make RSSI based systems weak but its characteristics such as no need for additional hardware, little power consumption, and low cost makes its use appealing when dealing with wireless sensor networks.

One other important parameter effecting the performance of RSSI is propagation loss (i.e. multi-path fading and shadowing), which is the change in channel behavior because of environmental characteristics (e.g. obstacles). In order to consider the effect of propagation loss, we use log-normal shadowing model to estimate the distance between transmitter and receiver. This model employs the signal power measurements at the receiver and the known transmit power [17]:

$$RSSI(d) = P_T(d) - \bar{PL}(d0) - 10\eta log_{10}(d/d0) + X_\sigma$$

Where $RSSI(d)$ is the RSSI value recorded at the distance d, $P_T(d)$ is device's initial transmit power, $\bar{PL}(d0)$ is the average path loss for reference distances, η is the path loss exponent and depends on the specific propagation environment, and X_σ is the zero mean Gaussian distribution random variable (in dB) with standard deviation of σ (also in dB), that models the random variation of the RSSI value.

By applying RSSI technique to our detection mechanism, we are able to detect different scenarios of attacks including Sybil attack as follows:

- **Different IDs, The same RSSIs:** This indicates the situation in which an outsider tries to listen to the beacons and after getting appropriate keys, send and receive packets. The malicious node could be either inside the network region with the same transmit power or outside the network region with stronger transmit power. It can also be the case where an adversary captures one of the nodes within the network (i.e. physical attack) and reads its keys and starts to interfere with the network by presenting multiple IDs. This scenario can fool the receiver since the PAN coordinator will have the same RSSI readings. However the PAN coordinator will be able to check the initial list of registered nodes and their assigned disc numbers and compare this information with the calculated location of the node based on its RSSI to detect the attack.

- **The same IDs, Different RSSIs:** Using RSSI readings, we can also detect displacement of sensor nodes. Since in our system, the nodes are of the static type, any major relocation of the nodes is considered an abnormal behavior and will trigger the detection system. The physical displacement attack is easy to launch by attackers and can be the start of more severe attacks. Therefore, detecting this attack is important in wireless sensor networks.

Detection Algorithm

RSSI-based localization schemes can be used to detect Sybil attack in an IEEE 802.15.4 network in which devices are connected to a central coordinator and all the communications will be through the coordinator. We can assume the network region a collection of concentric circles creating same size discs with the PAN coordinator

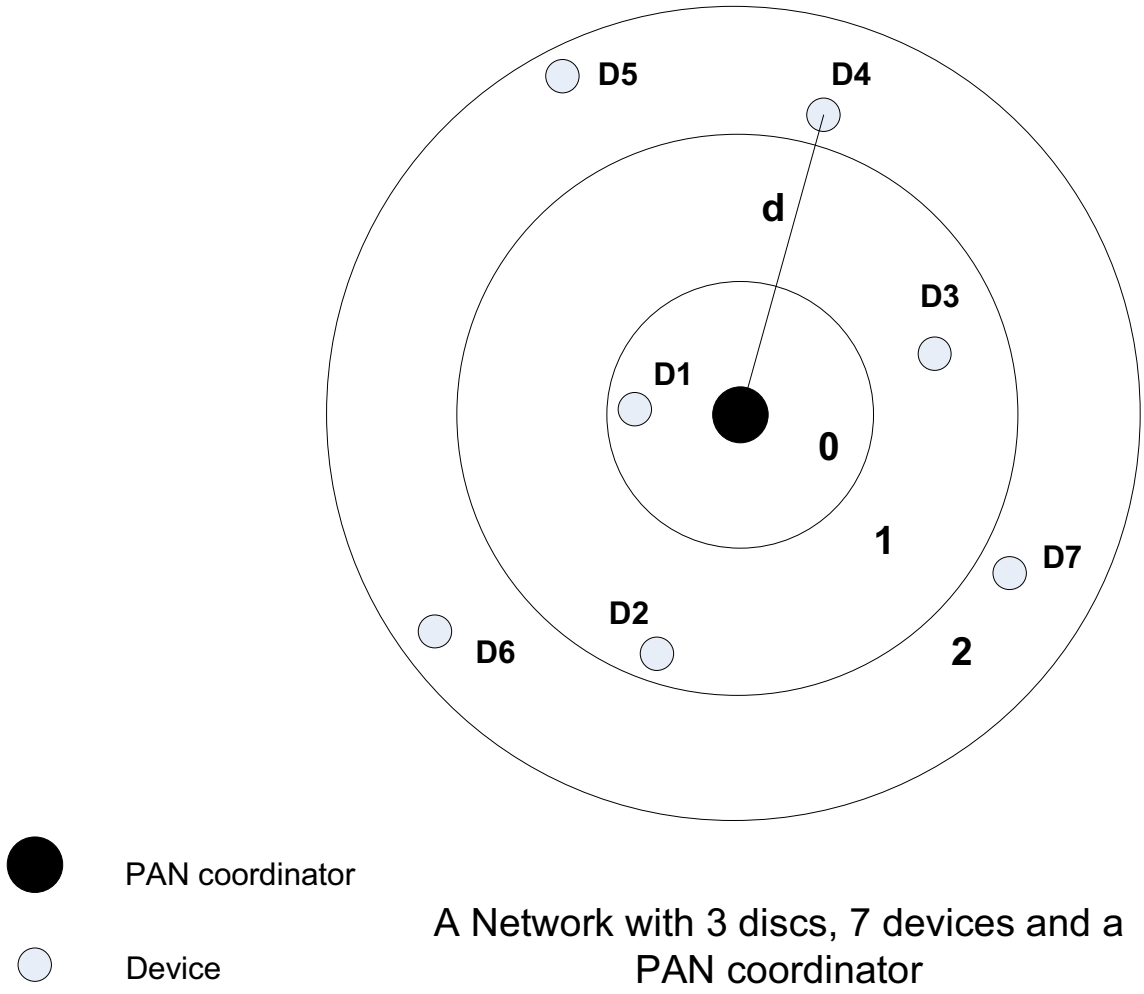


Figure 4.4: An Example of network layout.

in the center as shown in Figure 4.4. At the time of sensor deployment, devices are assigned a disc number [1] and a device ID. The PAN coordinator stores this information in a table.

Upon receiving a packet, the PAN coordinator estimates location of the sender device (the distance between the device and the PAN coordinator) using equation

[1]Depending on the type of application the assignment of disc numbers could be random or based on a design for the specific application needs

in 4.3.2 and associates this value to the device ID embedded in the packet header. The computed distance will determine a disc number in which the sending device is located and the PAN coordinator compares this information with the list stored at the deployment time. A Sybil attack and/or node displacement attack will be detected if the computed (disc number,ID) set differ from the stored information.

In practice, RSSI fluctuates and does not stay the same even if the location of the two transceivers are fixed [14]. Thus we need a mechanism to estimate average RSSI precisely. To estimate average RSSI (i.e. mean), we will use the following equation upon receiving a packet from a node [31]:

$$EstimatedRSSI = (1 - \alpha) * EstimatedRSSI + \alpha * RSSI$$

This equation is recursive and the initial value of $EstimatedRSSI$ is selected as the mean of Gaussian distribution. $EstimateRSSI$ is a weighted average of sample RSSI values. This weighted average puts more weight on recent samples than old samples since the more recent samples better reflect the current state in the network. The degree of weighing decrease is expressed as a constant smoothing factor α, a number between 0 and 1. α may be expressed as a percentage, so a smoothing factor of 10 percent is equivalent to $\alpha = 0.1$. Calculating variability of RSSIs, as well as the estimate of RSSI is useful to find an interval in which most of RSSIs for an individual node, fall. We use the following equation to calculate such a deviation:

$$DevRSSI = (1 - \beta) * DevRSSI + \beta * \mid RSSI - EstimatedRSSI \mid$$

Similar to the previous equation, this is recursive too and the initial value of $DevRSSI$ is calculated as σ (i.e. the standard deviation of the Gaussian distribution).

Finally the RSSI interval can be calculated using the following equation:

$$RSSIInterval = RSSIInterval + n * DevRSSI$$

Chapter 5

Simulation Model and Evaluation Results for an IEEE 802.15.4 Network

This chapter describes the model that simulates security and intrusion detection mechanisms in an IEEE 802.15.4 network. All the simulations are done using Artifex [24] a general development platform for discrete event simulations. I have run the simulation to measure the performance. The values of MAC parameters (e.g. buffer size, packet size) are selected according to the IEEE 802.15.4 standard specifications. The performance parameters are:

- **Number of nodes:** The number of nodes represents the total number of devices within a cluster. In this simulation, the value of this parameter will range from 3 to maximum of 60 devices.

- **Simulation runtime:** The simulation runtime represents the time duration in which the simulation will run.
- **Packet size:** The size of packet to be transmitted from device to PAN coordinator and vise versa.
- **Power resource:** The available power resource for a node from its creation to its death.
- **Number of malicious nodes:** The number of malicious nodes represents the total number of devices under attack.

The simulation model is set to measure both effectiveness and performance of the network under three main settings. First, the basic simulation model of the IEEE 802.15.4 network without any security and intrusion detection mechanisms will be evaluated. Second, the network's simulation model with security considerations will be assessed. For both of these settings the following performance measures will be observed:

- **Throughput:** the fraction of time used to transmit data payload (bits) to total time needed to access the medium and transmit a complete packet successfully.
- **Access probability:** The probability of accessing the medium by a device in order to send a packet without collision.
- **Blocking probability:** The probability of a packet being dropped at the device's buffer.

- **Key cost:** The duration it takes for a device to establish a key with the PAN coordinator.

- **Utilization:** The proportion of time during the simulation that devices are sending and receiving packets.

- **Sleep Probability:** The probability of going to the sleep (i.e. low power mode) for a device.

The third main setting will include the attacks and intrusion detection mechanisms to detect them. The effectiveness and performance of the network is measured and compared to the previous settings. For the case when the system carries an intrusion detection mechanism, the additional performance measures are:

- **Probability of detection:** This measurement determines the rate of attacks detected correctly by an IDS in a given environment during a particular time frame. The difficulty in measuring the detection rate is that the success of an IDS is largely dependent upon the set of attacks used during the test. Also, the probability of detection is reversely dependent with the false positive rate. To have a higher probability of detection, a tougher detection system needs to be implemented. For example, in the case of "signature-based" detection mechanism, higher probability of detection means wider set of attack signatures. As the number of signatures in a detection system grows, there will be more chances of marking an event as intrusion when it is really just a normal behavior of the network. Depending on the desired security level of the application an IDS can be tunned (i.e. different configurations can be used by setting the level

of detection) to favor either the ability to detect attacks or to minimize false positives.

- **Probability of false alarm:** This measurement determines the rate of false positives produced by an IDS in a given environment during a particular time frame. A false positive or false alarm is an alert caused by normal non-malicious background traffic.

- **Detection duration:** This measurement specifies the duration of detection of an attack which is from when an attack occurs till the response module is triggered.

The simulation model settings will also include power management policies and the performance of the network will be measured in the extended lifetime mode. To make sure the performance measures after the system reach a steady state, one tenth of the simulation time is defined as the run-up period of the simulation.

5.1 Key Exchange and Power Management

For the remaining of this section I will first give a quick introduction of beacon-enabled simulation model of 802.15.4 [49, 40] and later explain the simulated key exchange process simulated in this work.

5.1.1 Beacon-Enabled IEEE 802.15.4 Simulation Model

The network communication model of this simulation is based on star topology. The model is built on three primary objects :PAN coordinator, Device and Medium.

The device and PAN coordinator objects are inter-connected via medium object in our simulation model.

Two different Token types are defined that play the role of packet and backoff. Packets can be any of beacon, MAC request, data and acknowledgment (ack) types. The communication is initiated when PAN coordinator first sends beacon to medium (beacons are sent after every 48t where t is duration of one backoff period). After receiving the beacon the medium starts a clock and sends pulse to all devices every t time.

Data packets are generated by device object following exponential distribution and are destined to a randomly chosen device. The packet is then sent to the medium and a copy of it is kept for retransmission if needed. Data packets are then received by the medium. If the number of received packets in medium is greater than 1, collision occurs. If there are no collisions, data packets are sent successfully to the PAN coordinator and the medium status is set to busy.

PAN coordinator is the next stop for data packets and is responsible for sending ack type packets to corresponding device after a specified delay. As of every packet, ack will be received first by the medium and then sent to corresponding device. When PAN coordinator is sending data to a device it keeps finite buffer for each device in the PAN. If the buffer of the device which the data packet is destined for is full, the packet will be discarded. In the case that there is still room in that device's buffer, the coordinator adds the destination ID of packet to the pending devices list and advertises the ID in the beacon. The device will notice that there is packet waiting for it and will initiate a MAC request packet to be sent to the coordinator. The PAN

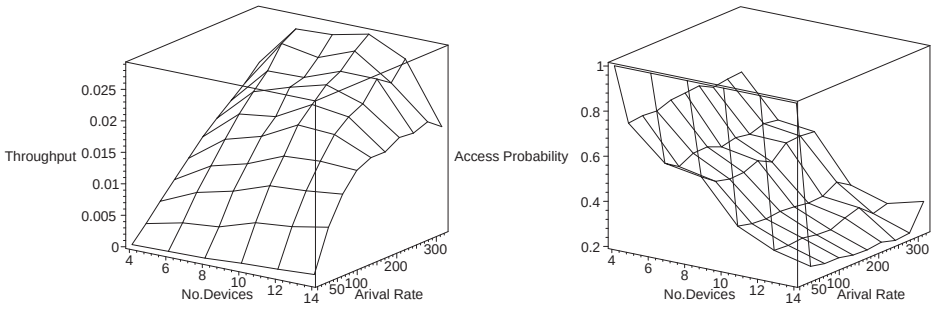
coordinator after receiving the request will perform round robin scheduling algorithm and choose the device to send the packet from its corresponding downlink buffer.

5.1.2 Adding Key Exchange Mechanism to the Simulation Model of IEEE 802.15.4 Network

In this section I describe the communication between the ordinary nodes and PAN coordinator which occurs as result from the link key exchange. We assume that devices are attached to the cluster and the formation of the piconet is finalized. Also, we assume the master keys are established, so that there is no threat of eavesdropping during exchange of master keys. The next step is generating link keys between each device and PAN coordinator. For the exchange of link keys, we will follow SKKE protocol as describe in Section 3.4.

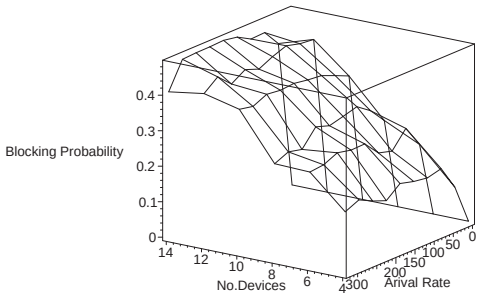
The process of key generation starts by PAN coordinator's advertisement for the first phase of key generation packets. Depending on which stage of generation we are in, the corresponding SKKE type of data packet (ranging from 1 to 4) will be processed (e.g the first data packet has the type of SKKE-1 and so on). According to the standard specification at most 7 devices can be advertised in each beacon. Therefore the PAN coordinator will advertise 7 devices in each beacon. According to the standard, each device listens to each beacon and if its ID has being advertised the device will send a request packet. Request packet is transmitted in CSMA-CA mode and can collide with other packets. If it is received successfully by the PAN coordinator it will be acknowledged and downlink packet transmission carrying the SKKE protocol data will follow in the downlink transmission.

In our model, key exchange packets have non-preemptive priority over data packets. If the node has started backoff process for data packet and it hears its ID in the beacon it will finish the current packet transmission before sending the request packet. However, if data packet arrives to the device's buffer while the key exchange is going on, its transmission will be postponed until device receives the new link key. PAN coordinator will first check key for the destination device from its access control list and no packet will be sent to the specific destination until the corresponding link key is already exchanged between PAN coordinator and the node. From this point on regular secure data packets will be immediately sent to the destination.



(a) Throughput

(b) Access probability



(c) Blocking Probability

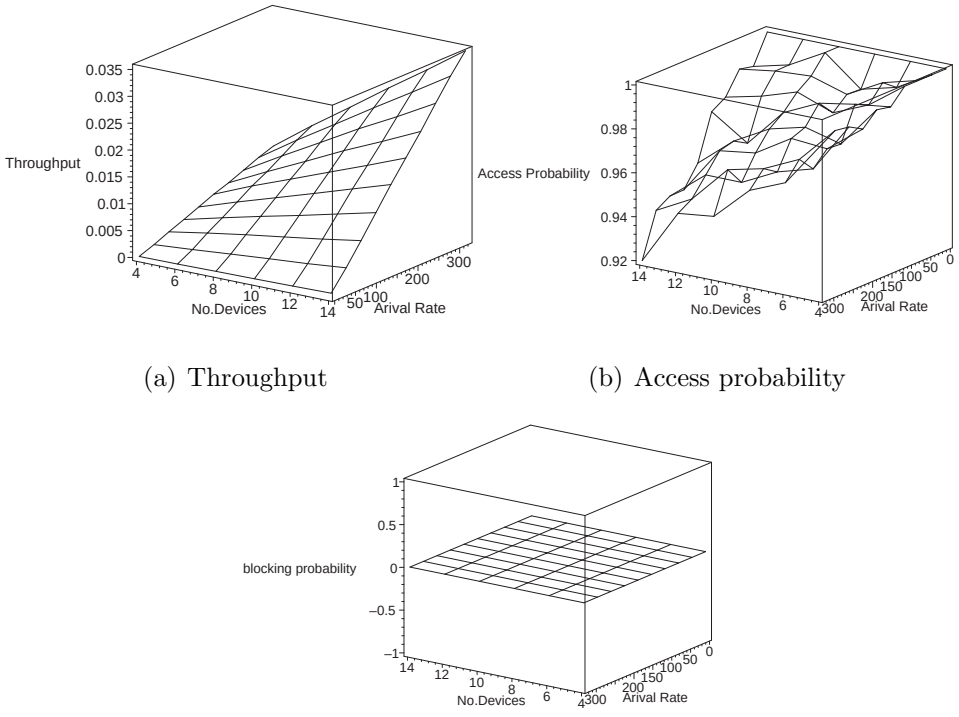
Figure 5.1: Throughput, Access Probability and Blocking Probability as the function of simulation time (backoffs) for the case when security is employed and all devices stop their communications to update their keys

5.1.3 Simulation Run and Analysis

I have implemented the physical, data link and security layer of an IEEE 802.15.4 cluster operating in beacon enabled, slotted CSMA-CA mode. The packet size without security overheads includes all physical layer and Medium Access control layer headers, and it is set to 30 bytes i.e. to three backoff periods. When packet signature (message authentication code) of 16 bytes is added to the total packet size had to be rounded to 5 backoff periods (the largest packet size could be set to 13 backoff periods).

The cluster under consideration contains 14 devices, each having buffer capacity for three packets. Packet arrival per device followed the Poisson process with average rate of 90.5 packets per minute. When the coordinator announces key exchange in the beacon, all nodes had to temporarily stop uplink data transmissions until they receive new key initialization values from the coordinator in the downlink packets. Due to complex downlink data-link transmission algorithm it is expected that key exchanges will adversely affect the regular sensing traffic.

The impact of the increase of packet size due to addition of Message Authentication Code, increased processing time needed for encryption in AES with CBC-MAC, and key exchange between the nodes over various packet arrival rates and cluster sizes is considered. Figure 5.1 presents throughput, access probability (probability of no packet collision) and blocking probability at the node's buffer when all security overhead is included. Results were taken for varying number of nodes and varying packet arrival rate per node. Figure 5.2 presents the same parameters (except the key exchange cost since it does not exist) when no security measures are deployed



(a) Throughput

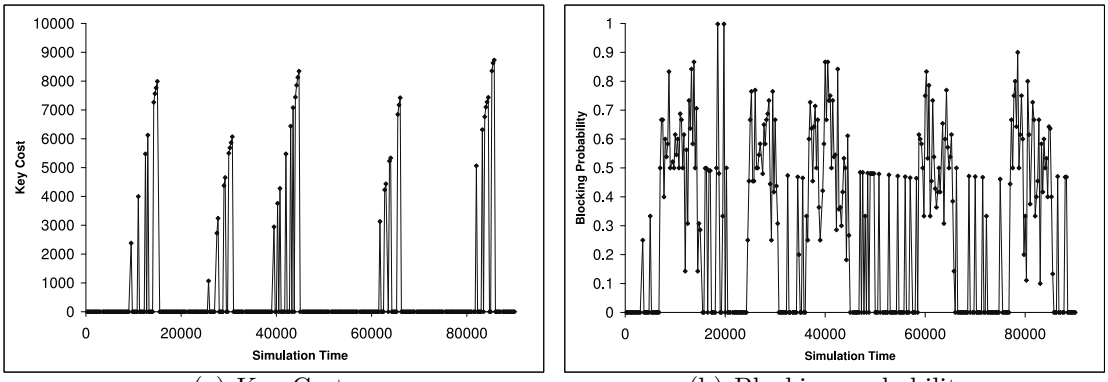
(b) Access probability

(c) Blocking probability

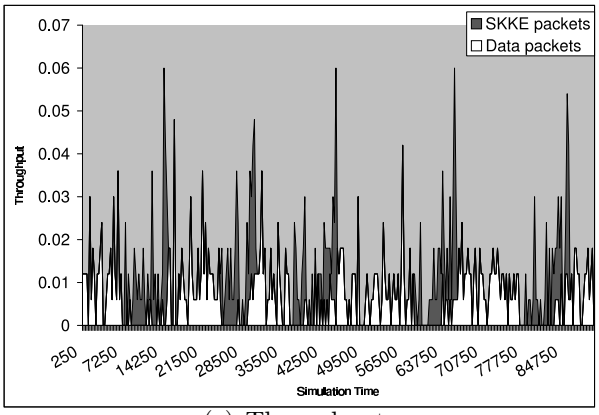
Figure 5.2: Throughput, Access Probability and Blocking Probability as the function of simulation time(backoffs) when no security technique is employeed.

in the network. We observe that without security measures, blocking probability is equal to zero i.e. that network works without losses.

The experiment to measure the cost of key update in the cluster contains seven devices only, and therefore it was possible to advertise the keys for devices in a single beacon. All devices temporarily stopped their data transmission during the key exchange. The behaviour of the cluster over time is presented in Fig. 5.3. Fig. 5.3(a) shows number of backoff periods spent in key exchange. We notice that average cost of key exchange is slightly below 2000 backoff periods, which gives 250-270 backoff periods per device. Knowing that the key exchange involves a total of two downlink



(a) Key Cost (b) Blocking probability



(c) Throughput

Figure 5.3: Key Cost, Blocking Probability and Throughput as the function of simulation time(backoffs) when devices stop their communications for key updates

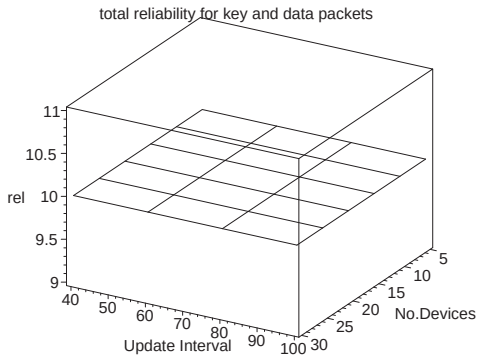
(uplink request + downlink data) transmissions and three uplink transmissions, we can conclude that one CSMA-CA access takes approximately 40 backoff periods. Given the backoff window sizes of (8, 16, 32) we can conclude that transmission commences in average after third backoff attempt which indicates moderate to large activity over the medium. The blocking probability at individual sensor node buffer over the snapshot periods is shown in Fig. 5.3(b). Due to large periods when device transmission is prevented during key exchange (well over 1500 backoff periods), the blocking probability skyrockets to values between 0.7 and 1. When the key exchange is

finished, normal data communications resume. As a result, the blocking probability drops abruptly to values around 0.3 and slowly declines further as the backlogged packets clear.

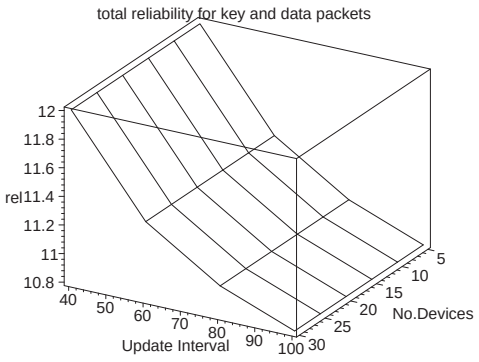
Fig 5.3(c) shows the throughput values measured during snapshot intervals of 250 backoff periods. The throughput of data packets is shown in white, while the throughput of key-exchange packets is shown in black. According to the throughput results reported in [40], the observed network regime without key exchange is slightly below the saturation condition (in saturation condition, all data transmissions end up in collisions).

I have also implemented distributed activity management in the simulator, assuming that the battery for each node has a fixed capacity. Battery capacity, which is expressed in backoff periods, is decremented by one for each backoff period in which the radio subsystem is active.

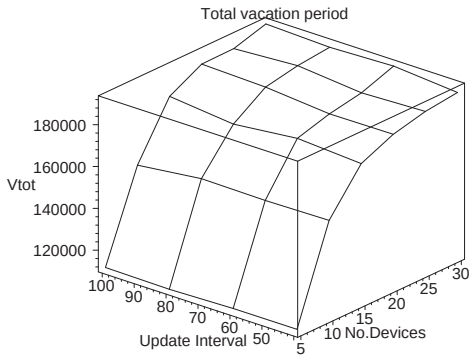
The key exchange threshold (n_k) is varied between 40 and 100 packets while the requested event sensing reliability was kept at $R = 10$ packets per second. Cluster size (n) was varied between 5 and 30 nodes. We assume that the network operates in the ISM band at 2.45GHz, with raw data rate 250kbps. The packet size was fixed at twelve backoff periods, and the device buffers have a fixed size of three packets. The packet size includes Message Authentication Code and all physical layer and Medium Access Control protocol sublayer headers, and is expressed as the multiple of the backoff period [6]. We also assume that the physical layer header has 6 bytes, and that the Medium Access Control sublayer header and Frame Check Sequence fields have a total of 9 bytes.



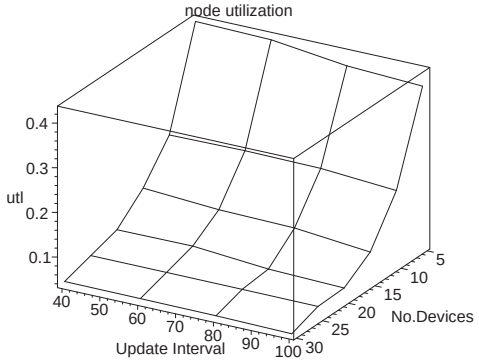
(a) Number of data packets transmitted per second.



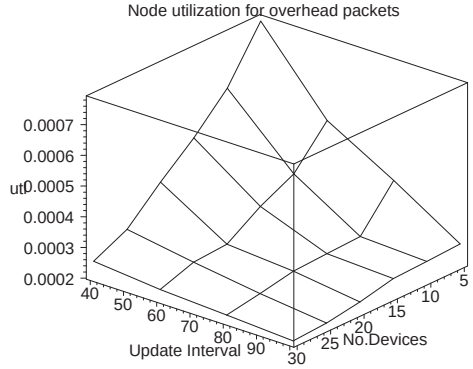
(b) Number of key and data packets transmitted per second.



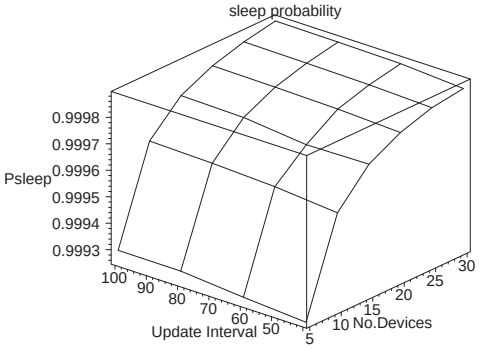
(c) Inactive period out of total simulation run = 200,000.



(d) Total utilization



(e) Utilization due to key exchange.



(f) Sleep probability.

Figure 5.4: Event sensing reliability for data and key+data, inactive period, total utilization and utilization for key packets, average number of active devices and sleep probability for a node.

Figure 5.4(b) shows total number of successfully transmitted packets (including key and data information) transmitted per second for requested data reliability of $R = 10$ packets per second (which is shown on Figure 5.4(a)). We note that the total number of packets hyperbolically grows when the key exchange threshold decreases linearly Figure 5.4(b). This is intuitive since the frequency of key updates is R/n_k per second and number of overhead packets with key information per second is equal to $8R/n_k$. We note that key exchange overhead becomes negligible only for $n_k \geq 90$. Probability that packet will not suffer from collision or noise error sharply drops when threshold for key exchange drops below 40 packets. Both the reliability overhead and success probability depend only on the requested event sensing reliability except for very small key update threshold. Sleep period, on the other hand, depends mostly on the number of alive nodes and impact of key exchange overhead is barely noticeable.

Total node utilization shown in Figure 5.4(d) depends mostly on the number of alive nodes, but it also increases with increase of the number of key exchanges per second and exact impact of the key exchange overhead is shown in Figure 5.4(e). Finally, sleep probability for each node is shown in Figure 5.4(f). Sleep probability dominantly changes with n, while the changes with n_k are much milder.

5.2 Interconnection of the IEEE 802.15.4 Clusters

I have simulated a two cluster network in which the clusters are interconnected by a Master-Slave bridge. Each cluster follows the start topology as its network communication model. The objects: device and PAN coordinator/bridge are interconnected via medium object. The PAN coordinator at the child cluster also acts as a bridge

and switches to the other cluster (i.e. parent cluster) when it has some data in its buffer.

We assume that all the sensing traffic from the child cluster occurs in the uplink direction with the parent cluster's coordinator as the ultimate destination except the traffic required for exchanging the link key; the parent cluster has local traffic (originating at ordinary nodes and PAN coordinator) as well. This assumption is valid in sensor networks where most, of the traffic will be directed toward the network sink. We assume that both clusters operate in beacon enabled, slotted CSMA-CA mode under the control of their respective cluster (PAN) coordinators.

Figure 5.5 shows the communications between and within the two clusters. The communication is initiated when PAN coordinator first sends beacon to the medium (beacons are sent after every $48t$ where t is duration of one backoff period). We assume that devices are attached to the cluster and the formation of the piconet is finalized. Also, we assume the master keys are established, so that there is no threat of eavesdropping during exchange of master keys. For the exchange of link keys, we will follow SKKE protocol as described in Section 3.4. The exchange of SKKE packets happens first between local devices and the PAN coordinator and also between the bridge after switching to the parent cluster as an ordinary device. For each downlink packet advertised by the PAN coordinator in the beacon, devices need to generate a MAC request packet and send it to the corresponding PAN coordinator. All of the communications need to be acknowledged by the recipients at each step.

After the link key exchange, data packet transmissions start in uplink direction originating from the devices. The key exchange packets have non-preemptive priority

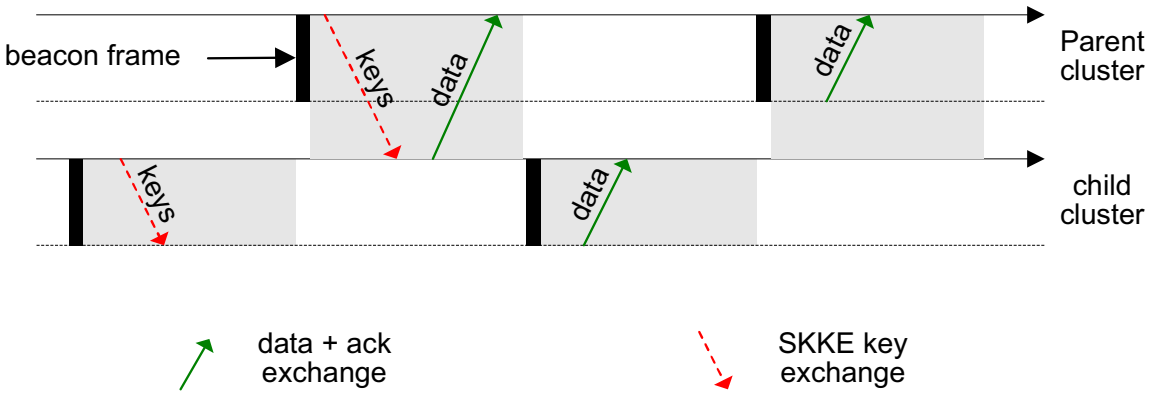


Figure 5.5: Communications between two clusters.

over data packets and if a data packet arrives to the device's buffer while the key exchange is going on, its transmission will be postponed until device receives the new link key. Of course the operation of the bridge is similar to what was given in Section 3.7 and data packets received from the local nodes will be stored in the buffer and are sent to the parent cluster each time the bridge switched to the other cluster.

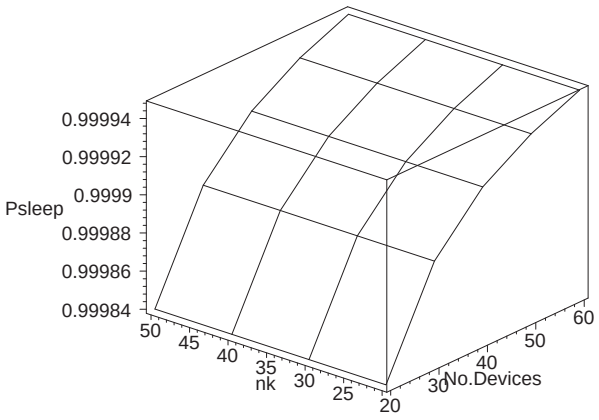
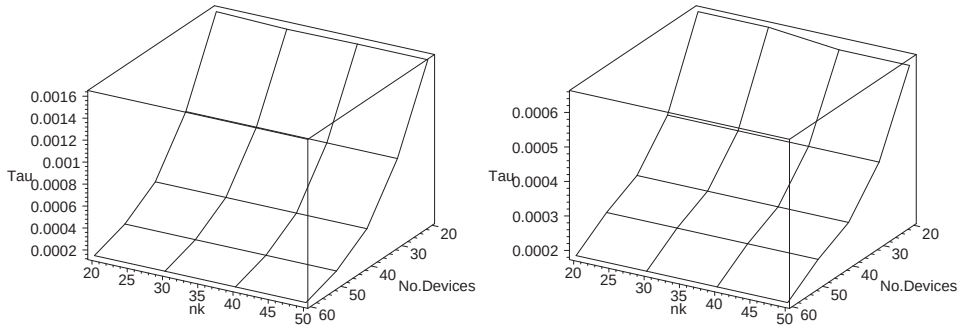


Figure 5.6: The probability of sleep for devices.



(a) Access Probability for devices in the parent cluster

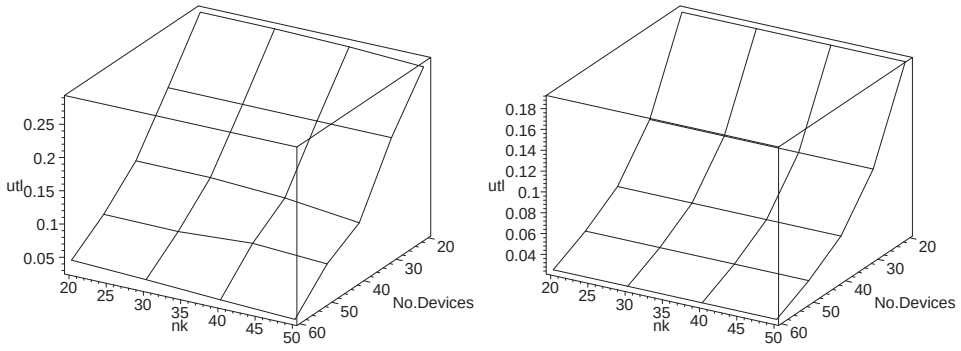
(b) Access probability for devices in the child cluster

Figure 5.7: Access Probability.

5.2.1 Simulation Run and Analysis

I have created the simulation model for the physical, data link and security layer of an IEEE 802.15.4 cluster. The packet size includes all physical layer, Medium Access control layer headers and security overheads, and it is set to 120 bytes i.e. to 12 backoff periods.

The number of devices (n) in both clusters is the same ranging from 20 to 60 devices each having buffer capacity for three packets. Packet arrival per device followed the Poisson process with one packet per second and the buffer size of 20 bytes is allocated for each device. If the number of transmitted packets to the medium is greater than 1, collision occurs. If there are no collisions, data packets are sent successfully to the PAN coordinator and the medium status is set to busy. Upon the successful receipt of packets at the coordinator/Bridge, they will be stored in the buffer of size 30 bytes (leading to a small blocking of packets at the bridge) and



(a) Utilization for devices in the parent cluster

(b) Utilization for devices in the child cluster

Figure 5.8: Node Utilization.

wait to be forwarded to the parent cluster. The newly received packets will replace the old ones and the old ones will be dropped (We assume that the parent cluster's coordinator, acting as the network parent, can accept all packets it receives).

The key exchange threshold (n_k) varies between 20 to 50 packets and indicates the maximum number of packets sent by any device in the cluster before the new link key establishment. This counter can be controlled at the coordinator side.

I have also considered distributed activity management and we assumed that the network operates in the ISM band at 2.45GHz, with raw data rate 250kbps. The requested event sensing reliability is kept at $R = 10$ packets per second and the PAN coordinator advertises this value in each beacon. Devices then calculate the probability of sleep and will decide whether or not to go to the inactive mode. Figure 5.6 presents the calculated value of P_{sleep} measured by running the simulator. Devices in the more populated cluster need less activity to match the fixed required event sensing reliability.

I have measured the probability of access to the medium at both clusters separately and the results are shown in Figure 5.7. It can be seen from the diagrams that the access probability decreases by the increase of the number of devices trying to send and receive packets. Since the devices in the parent cluster need to compete with the bridge, the probability of access is higher at the parent cluster due to higher contention at the medium.

Total node utilization shown in Fig. 5.8 depends mostly on the number of alive nodes, but it also slightly decreases with increase of the number of key exchanges per second. Again utilization is higher in the parent cluster due to more contention at the medium.

5.3 Sleep Deprivation Attack and Detection Mechanism

5.3.1 Model Details

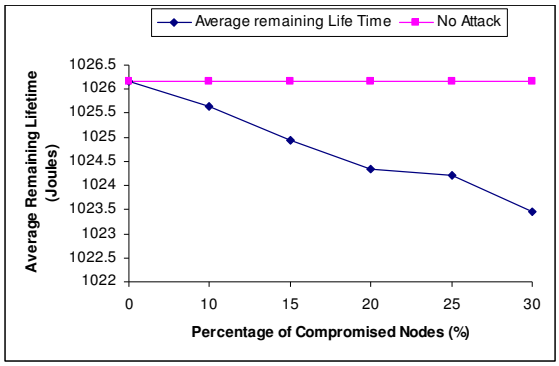
We assume that once sensor nodes exhaust their batteries, they become dead and no exchange of the power supply will happen. This assumption is realistic since sensor nodes are usually deployed in hardly reachable areas where renewing/recharching their batteries is difficult and in some cases impossible. The sensors work under a power management mechanism and switch to a low power mode also known as sleep mode when there are no packets to send or receive. We also assume that sensor nodes are not completely tamper resistant and an adversary could capture a node and manipulate the code inside, however, we will only focus on sleep deprivation attack.

The placement of adversary nodes is considered to be uniformly random to reflect the adversaries desire to maximize the damage by spreading all over the network.

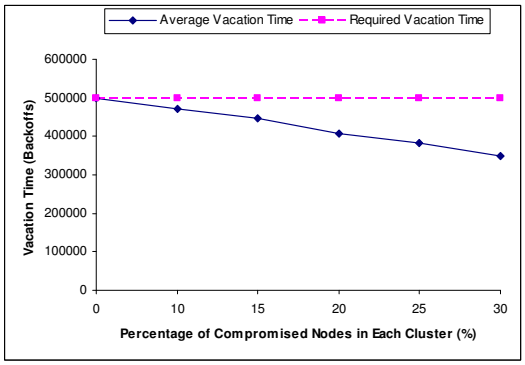
I have simulated an IEEE 802.15.4 network with presence of secure data exchange protocol- SKKE. The nodes are connected to the PAN coordinator via medium following star topology and clusters are interconnected via maser/slave bridge. We assume that the PAN coordinator has a large power supply and is highly secure so that it can not be compromised. Malicious nodes will be introduced to the network and they will launch sleep deprivation attack. The attack will prevent the compromised nodes from going to sleep. An intrusion detection algorithm is in place to detect and report the suspected nodes.

5.3.2 Simulation Design and Analysis

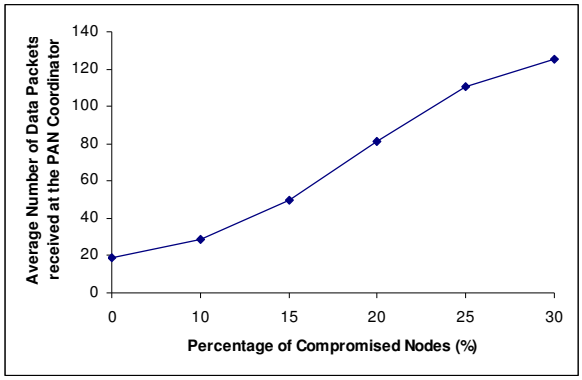
The network model is under sleep deprivation attack only and contains 60 devices in each cluster each having buffer capacity for 2 packets. The simulation run times are in backoffs (0.00032 sec) and have been set for 1 million backoffs which is almost 5 minutes. Figure 5.9 shows network's status under attack. The average remaining lifetime of sensor nodes in the network is definitely affected by the attack. Although these results are taken after performance of the network for a small amount of time, we can still see a considerable decrease in the remaining lifetime of the nodes. Also interesting to notice the direct effect of the attack on the average sleeping period of the nodes. With 30 percent of the nodes under attack, the sleeping time is decreased by 50 percent. Figure 5.9(c) shows the changes in the average number of data packets received at the PAN coordinator from all the nodes in the network. The compromised



(a) Life Time



(b) Vacation Time

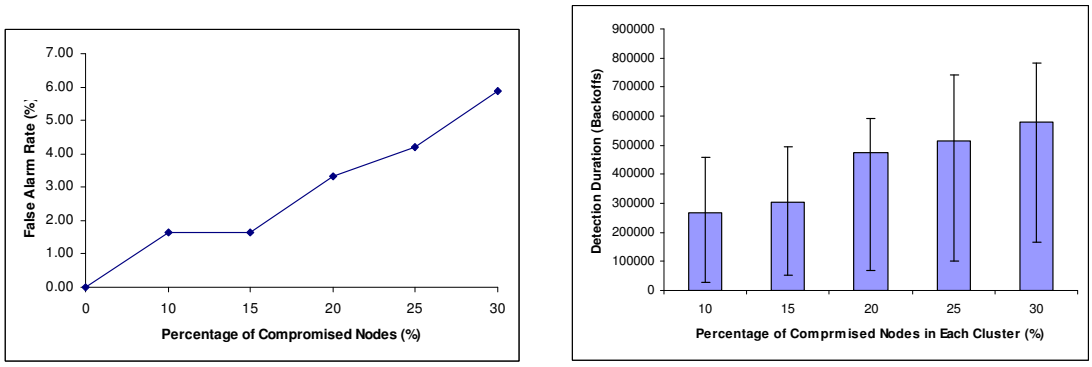


(c) Number of Data Packets

Figure 5.9: Lifetime (joules), Sleep period (backoffs), and the average number of data packets received at the PAN coordinator for the network under sleep deprivation attack.

nodes under sleep deprivation attack seem to have sent more number of data packets in average. This is caused by the deprived sleep of the nodes because of their malicious behavior.

In the second phase of the simulation, I have simulated our proposed detection mechanism to detect compromised nodes. Most of the processing for this purpose takes place at the PAN coordinator side with high processing power. Figure 5.10(a)



(a) False Alarm

(b) Detection Duration

Figure 5.10: The false alarm rate and detection duration (backoffs) of the detection system for sleep deprivation attack.

shows the false positive rate of the detection system. With have of the nodes in the network being compromised, the detection algorithm has a low false positive rate of under 20 percent. I have also measures the period of time it takes for the compromised node to be detected since getting compromised and the results are shown in Figure 5.10(b).

5.4 False Data Injection Attack and Detection Mechanism

5.4.1 Model Assumptions and Details

To examine the detection mechanism that is intended to detect malicious nodes which inject false date, I have simulated a network in which the nodes send sensed data packets to the PAN coordinator. R. A. Hellstrom and B. G. Mark [19] present

measurement data using wireless sensors for district wet and dry periods within an Alpine Valley. The measurements are taken from the weather station installed in the Llanganuco valley of Cordillera Balanca. The present several measurements such as air temperature, Wind, Soil moisture, and soil temperature. I have used the sample data for soil temperature for both dry and wet periods.

The simulated cluster contains 60 devices. Each device calculates the corresponding measurements for the specific time of the day. I have used curve fitting for the sample data to obtain the formula that gives values of soil temperature for every hour in the day. We assume that healthy nodes (i.e. the nodes which are not compromised and report correct data) report only the measurements for the wet period. This means that the reported state of the network in case of no attack should be the measurements of wet period (presented in [19]). We also assume that intruders want to reverse the measurements and instead report the measurements of the dry period (i.e. false report). A compromised devise uses the formula obtained for dry period from sample data. We assume that the network starts in complete health state and will not get compromised for a while. This assumption will give the PAN coordinator chance to collect enough healthy reports to base its decision on. Compromised nodes will then be introduced to the cluster and start sending false reports.

We assume that sensor nodes are not completely tamper resistant and an adversary could capture a node and manipulate the code inside, however, we will only focus on false data injection attack assuming the network will be under only one attack. The placement of adversary nodes is considered to be uniformly random to reflect the adversaries desire to maximize the damage by spreading all over the network.

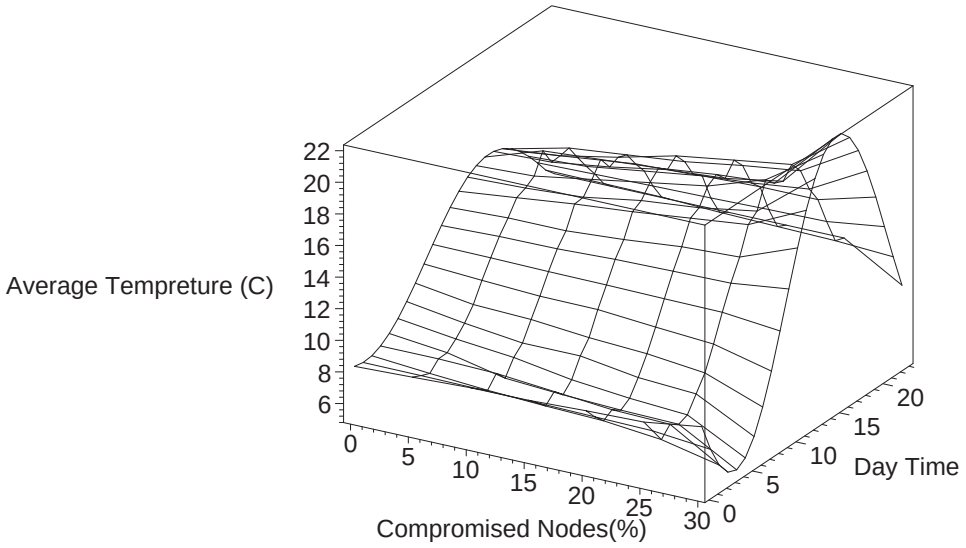
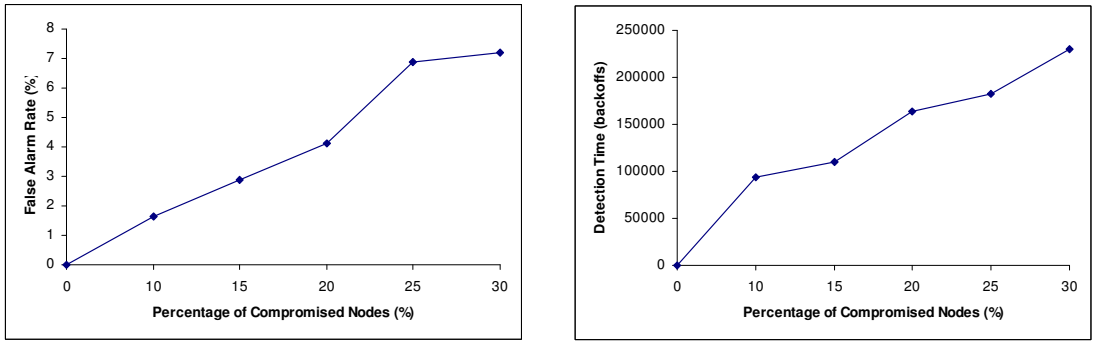


Figure 5.11: Soil temperature values throughout the day reported by sensor nodes.

5.4.2 Evaluation and Results

I have simulated a network with assumptions mentioned to evaluate the effects of false data injection attack and its detection mechanism. The simulation run times are in backoffs (0.00032 sec) and have been set for 1 million backoffs which is almost 5 minutes. In the first phase of the simulation sensor nodes will be captured and start maliciously sending false measurements (i.e. reporting soil temperatures for dry condition throughout the day). Figure 5.11 shows average measurements (C) received at the PAN coordinator for different times in the day. The percentage of compromised nodes varies between 0 to 30 percent. If the adversaries could compromise more than 30 percent of the nodes within the network, they can successfully full the PAN coordinator and report dry condition.

Next we simulate the detection mechanism discussed previously to check the accu-



(a) False Alarm

(b) Detection Duration

Figure 5.12: The false alarm rate and detection duration (backoffs) of the detection system for false data injection attack.

racy of the system. In this phase of the simulation again nodes will get compromised but this time there is a detection mechanism in place by the PAN coordinator. Figure 5.12(a) shows the false alarm rates for different percentages of compromised nodes. When less than 30 percent of the nodes are sending false report, the false alarm rate is under 10 percent which is considered low and means the compromised nodes can be detected with high probability. The detection durations (the duration between node capture and detection) are presented in Figure 5.12(b) in backoffs. As expected, the time it takes for the PAN coordinator to detect all the malicious nodes increased when there are more compromised nodes in the network.

5.5 Sybil Attack and Detection Mechanism

5.5.1 Model Assumptions and Details

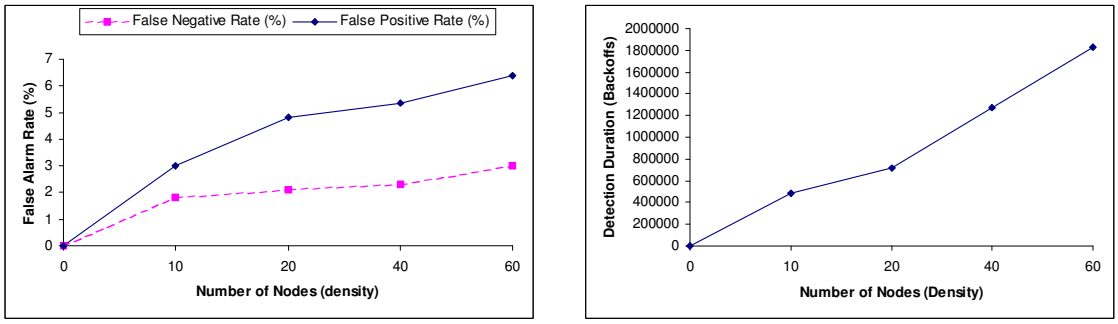
Our simulated model consists of a set of static nodes, such as the tmote-sky [11] with CC2420 chip radio, which is an IEEE 802.15.4 compliant radio and provides a RSSI output that is sampled by an 8-bit ADC. The RSSI is mainly used for RF signals, and the estimate unit is dBm.

In the simulated network, each device is given a unique ID at the time of deployment. The nodes are connected to the PAN coordinator following star topology. We assume that the PAN coordinator has a large power supply and is highly secure so that it can not be compromised. We assume an initial set of good nodes (i.e. non-malicious) and later malicious nodes will be introduced to the network and they will launch Sybil [1] and/or node displacement [2] attack. The new nodes with malicious behavior are placed randomly within the network region with the corresponding disc number. We assume that malicious nodes are able to get the necessary keys and start sending and receiving packets. The network model is assumed to be under Sybil/node displacement attack only. An intrusion detection algorithm is in place to detect and report the suspected nodes.

5.5.2 Evaluation and Results

I have simulated an IEEE 802.15.4 network with presence of secure data exchange protocol (i.e. SKKE) to study the performance of the system. The network operates

[1] Present different or multiple identities
[2] Physically displace or remove some of the sensor nodes from their original position



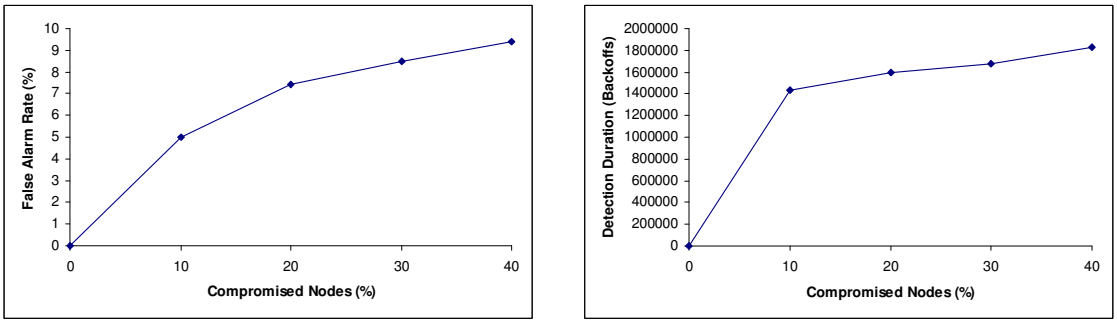
(a) False Alarm Rate

(b) Detection Duration

Figure 5.13: False positive, false negative rates (%), and detection duration (backoffs) changes for different network densities.

at 2.4 GHz with raw data rate of 250 kbps in an obstructed indoor environment and is formed in a circle planar. The number of nodes varies between 10 to 60 nodes, changing the level of density in the network. At the beginning all the nodes have identical transmit power of 100 dB. The simulation runtimes are in backoffs (0.00032 sec) and have been set for 2 million backoffs which is almost 10 minutes.

I have examined both completeness (i.e. probability of detecting the attack) and accuracy (i.e. probability of not reporting non-malicious nodes as malicious). Figure 5.13(a) illustrates both false positives and false negatives while 20 percent of the nodes are malicious. We notice that as the network gets more crowded, false alarm rate increases. The interesting point here is that the false negative rate is much smaller than false positive rate. Specially in the case of Sybil attack, it is very important to detect the attack and prevent further damage to the network which can be serious but the slightly higher false positive rate can be ignored since it only affects the performance of the system. Figure 5.13(b) shows the changes in the detection latency as the network grows. This is because of the bigger sampling size needed in



(a) False Alarm Rate

(b) detection Duration

Figure 5.14: False alarm rate (%) and detection duration (backoffs) changes for different percentage of compromised nodes.

order to get the desired false alarm rate. The trade-off between detection duration and false alarm rate is visible and network administrators need to consider the specific application needs and tune the detection thresholds accordingly.

Figure 5.14 shows the direct relationship between the percentage of malicious nodes in the network and false alarm rate and also detection duration while having a network with high level of density (60 nodes). If 40 percent or more of the nodes in the network be malicious/compromised, the false alarm rate will exceed 10 percent, which is considered high in such a network. We can say that our detection mechanism works acceptable when having 30 percent or less number of malicious/compromised nodes.

Finally, the standard deviation of the Gaussian distribution has a direct effect on false alarm rate as shown in Figure 5.15. Again to keep a high level of performance for the system, the value of standard deviation needs to be lower than 5.

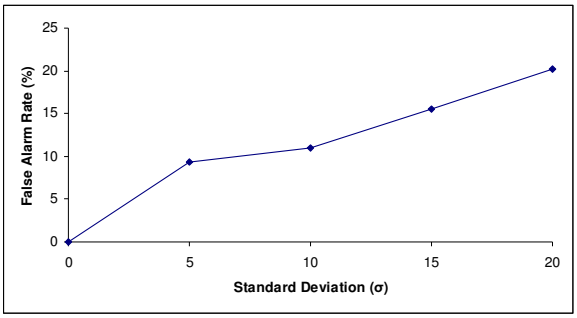


Figure 5.15: False alarm rate (%) changes for different standard deviation values.

Chapter 6

Conclusion and Future Work

The need for security in WSNs is being sensed more than ever because of the popularity of these networks. However, there are not enough experiments on the performance of security and intrusion detection in WSNs or particularly in IEEE 802.15.4 networks as one technology of WSNs.

I have simulated and studied key exchange process integrated with reliable sensing and power management in beacon enabled IEEE 802.15.4 cluster and the results confirm our expectations. Data encryption is provided by exchanging link keys between each device and PAN coordinator. The signature payload plays a big role on performance of the network. I have developed model of key exchange integrated into the sensing function of beacon enabled IEEE 802.15.4 cluster. The results show important impact of the ratio of the event sensing reliability and key update threshold on the clusters energy consumption. We have evaluated the impact of the threshold for key update on the clusters descriptors. The results can give useful hints for the choice of frequency of key updates for required event sensing reliability. I have also

simulated major attacks resulting from node compromise within an IEEE 802.15.4 cluster in spite of having security measures in place. I have evaluated the effects of such attacks on the network performance and proposed lightweight detection mechanisms to detect them. The results give an overview on parameter selection for the detection system and prove that utilization intrusion detection system in an IEEE 802.15.4 is possible while keeping a high level of accuracy and completeness. In my simulated model, the false positive rates are slightly higher in compare to the false negative rate. This is negligible since complete detection of compromised nodes is rather important.

As part of the future work, we can consider several other attacks on different layers of the network and expose our simulation model to them. Finding lightweight detection mechanisms for such attacks is another possible research idea. After collecting a comprehensive amount of data, these attacks and their corresponding detection methods can be compared. Providing a complete comparison of the severity and different detection parameters of attacks will help us make decisions regarding parameter selection for different aspects of security and intrusion detection (e.g. frequency of key update and completeness degree of detection algorithm) when it comes to the actual implementation of an IEEE 802.15.5 network.

.1 Acronyms

ACL	Access Control List
AES	Advanced Encryption Standard
API	Application Programming Interface
CAMA-CA	Carrier Sense Multiple Access with a Collision Avoidance
CBC-MAC	Cipher Block Chaining Message Authentication Code
CCA	Clear Channel Assessment
CCEF	Commutative Cipher-Based En-Route Filtering
CCM	Counter with CBC-MAC
CPU	Central Processing Unit
CTR	CounTeR
ED	Energy Detection
GPS	Global Positioning System
IDS	Intrusion Detection System
LAD	Localization Anomaly Detection
LR-WPAN	Low Rate Wireless Personal Area Network
MAC	Message Authentication Control
MIC	Message Integrity Code
PAN	Personal Area Network
PIN	Personal Identification Number
PNM	Probabilistic Nested Marking
RIP	Routing Information Protocol
ROM	Read Only Memory
RSSI	Received Signal Strength Indication
SDF	Statistical En-route Filtering
SKKE	Symmetric-Key Key Exchange
SOC	Self-Organized Criticality
WSN	Wireless Sensor Network

Bibliography

[1] ANSI X9.63-2001,Public Key Cryptography for the Financial Services Industry-Key Agreement and Key Transport Using Elliptic Curve Cryptography. American Bankers Association, 2001.

[2] FIPS Pub 198, The Keyed-Hash Message Authentication Code (HMAC). Federal Information Processing Standards Publication 198,US Department of Commerce/N.I.S.T., 2002.

[3] Standard for part 15.4: Wireless medium access control (MAC) and physical layer (PHY) specifications for low rate wireless personal area networks (WPAN). IEEE Std 802.15.4, IEEE, 2003.

[4] A. Agah, S. K. Das, K. Basu, and M. Asadi. A non-cooperative game approach for intrusion detection in sensor networks. In *Third IEEE International Symposium on Network Computing and Applications*, pages 343–346, September 2004.

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

[6] ZigBee Alliance. ZigBee specification (ZigBee document 053474r06, version 1.0), December 2004.

[7] F. Amini, J. Mišić, M. Khan, and H. Pourreza. Performance of IEEE 802.15.4 clusters with power management and key exchange. *Journal of Computer Science and Technology, JCST*, 23(3):377–388, May 2008.

[8] F. Anjum, D. Subhadrabandhu, and S. Sarkar. Signature based intrusion detection for wireless ad-hoc networks: A comparative study of various routing protocols. In *Vehicular Technology Conference, Wireless Security Symposium, Orlando, Florida*, volume 3, pages 2152–2156, October 2003.

[9] M. Bellare, J. Kilian, and P. Rogaway. The security of the cipher block chaining message authentication code. *Computer and System Sciences*, 61(3):362–399, 2000.

[10] M. Bishop. *Computer Security: Art and Science*. Addison Wesley, Pearson Education, Inc., Boston, 2004.

[11] Moteiv Corporation. Tmote sky: Ultra low power IEEE 802.15.4 compliant wireless sensor module data sheet, 6 February 2006.

[12] A. P. R. da Silva, M. H. T. Martins, B. P. S. Rocha, A. A. F. Loureiro, L. B. Ruiz, and H. C. Wong. Decentralized intrusion detection in wireless sensor networks. In *Proceedings of the 1st ACM international workshop on Quality of service & security in wireless and mobile networks*, pages 16–23, 2005.

[13] S. Datema. Master's thesis in computer science: A case study of wireless sensor

network attacks. Faculty of Electrical Engineering, Mathematics, and Computer Science, Delft University of Technology, September, 2005.

[14] M. Demirbas and Y. Song. An RSSI-based scheme for sybil attack detection in wireless sensor networks. In *WOWMOM '06: Proceedings of the 2006 International Symposium on on World of Wireless, Mobile and Multimedia Networks*, pages 564–570, Washington, DC, USA, 2006. IEEE Computer Society.

[15] S. S. Doumit and D. P. Agrawal. Self-organized criticality and stochastic learning based intrusion detection system for wireless sensor networks. In *MILCOM: IEEE Military Communications Conference*, volume 1, pages 609–614, October 2003.

[16] W. Du, L. Fang, and P. Ning. Lad: Localization anomaly detection for wireless sensor networks. In *IPDPS: 19th IEEE International Parallel and Distributed Processing Symposium*, pages 41a – 41a, April 2005.

[17] A. Flammin, D. Marioli, G. Mazzoleni, E. Sisinni, and A. Taroni. Received signal strength characterization for wireless sensor networking. In *IMTC '06: Proceedings of the IEEE Instrumentation and Measurement Technology Conference, 2006*, pages 207–211, Sorrento, 24-27 April 2006.

[18] C. Hedrick. Request for comments: Routing Information Protocol(RIP), Network Working Group, Rutgers University. http://www.ietf.org/rfc/rfc1058.txt, June 1988.

[19] R. A. Hellstrom and B. G. Mark. An embedded sensor network for measur-

ing elevation effects on temperature, humidity, and evapotranspiration within a tropical alpine valley. *63rd Eastern Snow Conference*, pages C1280+, December 2006.

[20] F. Hu and N. K. Sharma. Security considerations in ad hoc sensor networks. *Ad Hoc Networks*, 3(1):69–89, January 2005.

[21] Y. C. Hu, A. Perrig, and D.B. Johnson. Ariadne: A secure on-demand routing protocol for Ad Hoc networks. In *Proc. ACM Mobicom 2002*, pages 12–23, September 2002.

[22] Y. Huang and W. Lee. A cooperative intrusion detection system for ad hoc networks. In *SASN: Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks*, pages 135–147, January 2003.

[23] H.Yang and S. Lu. Commutative cipher based en-route filtering in wireless sensor networks. *IEEE 60th Vehicular Technology Conference, VTC2004-Fall*, 2:1223–1227, 26-29 Sept. 2004.

[24] RSoft Design Inc. Artifex v.4.4.2, 2003.

[25] O. Kachirski and R. K. Guha. Effective intrusion detection using multiple sensors in wireless ad hoc networks. In *Proceedings of the 36th Annual Hawaii International Conference on System Sciences*, pages 57 – 65, 2003.

[26] C. Karlof and D. Wagner. Secure routing in sensor networks. In *Proc. 1st IEEE Interantional Workshop on Sensor Network Protocols and Applications 2003.*

[27] I. Khalil, S. Bagchi, and N. B. Shroff. Liteworp: A lightweight countermeasure for the wormhole attack in multihop wireless networks. In *Proc. of the International Conference on Dependable Systems and Networks DSN , 28 June - 1 July 2005, Yokohama, Japan*, pages 612–621, 2005.

[28] M. Khan, F. Amini, J. Mišić, and V.B. Mišić. The cost of security: Performance of Zigbee key exchange mechanism in an 802.15.4 beacon enabled cluster. In *Proc. WSNS'06, held in conjunction with IEEE Mobile Adhoc and Sensor Systems (MASS06) 2006*, pages 876–881, Vancouver, CA, October 2006.

[29] M. Khan, F. Amini, J. Mišić, and V.B. Mišić. Key exchange in 802.15.4 networks and its performance implications. In *Proc. The Second International Conference on Mobile Ad Hoc and Sensor Networks (MSN 2006) 2006*, pages 497–508, Hong Kong, China, December 2006.

[30] C. Kruegel. Applying mobile agent technology to intrusion detection. In *Distributed Systems Group, Technical University of Vienna*, 2002.

[31] J. F. Kurose and K. W. Ross. *Computer Networking: A Top-Down Approach Featuring the Internet*. Addison-Wesley, 2005.

[32] L. Lazos and R. Poovendran. Serloc: Robust loacalization for wireless sensor networks. *ACM Transacations on Sensor Networks*, 1(1):73–100, 2005.

[33] J. Li, J. Jannotti, D.S.J. De Couto, D.R. Karger, and R. Morris. A scalable location service for geographic Ad Hoc routing. In *Proc. ACM Mobicom 2000*, pages 120–130, 2000.

[34] D. Liu, Ning P., and W. Du. Detecting malicious beacon nodes for secure loaction discovery in wireless sensor networks. In *Proc. International Conference on Distributed Computer Systems, ICDCS 2005*, pages 609–619, 2005.

[35] A.J. Menezes, P.C. van Oorschot, and S.A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1997.

[36] V. P. Mhatre, C. Rosenberg, D. Kofman, R. Mazumdar, and N. Shroff. A minimum cost heterogeneous sensor network with a lifetime constraint. *IEEE Transactions on Mobile Computing*, 4(1):4–15, Jan-Feb 2005.

[37] J. Misic, J. Fung, and V. B. Misic. Interconnecting 802.15.4 clusters in master-slave mode: queueing theoretic analysis. In *ISPAN '05: Proceedings of the 8th International Symposium on Parallel Architectures,Algorithms and Networks*, pages 378–385, Washington, DC, USA, 2005. IEEE Computer Society.

[38] V. Mittal and G. Vigna. Sensor-based intrusion detection for intra-domain distance-vector routing. In *ACM Conference on Computer and Communications Security*, pages 127–137, 2002.

[39] J. Mišić, F. Amini, and M. Khan. Performance implications of periodic key exchanges and packet integrity overhead in an 802.15.4 beacon enabled cluster. *To appear in Special Issue: "Large-Scale Wireless Sensor Networks: Challenges and Applications", International Journal of Sensor Networks, IJSNet*, 3(1), 2006.

[40] J. Mišić, S. Shafi, and V. B. Mišić. Performance of a beacon enabled IEEE

802.15.4 cluster with downlink and uplink traffic. *IEEE Transactions on Parallel and Distributed Systems*, 17(4):1–16, April 2006.

[41] J Mišić, S. Shafi, and V.B. Mišić. Cross-layer activity management in a 802.15.4 sensor network. *IEEE Communications Magazine*, 44(1):131 – 136, October 2006.

[42] J. Newsome, E. Shi, D. X. Song, and A. Perrig. The sybil attack in sensor networks: analysis & defenses. In *Proc. of the Third International Symposium on Information Processing in Sensor Networks, IPSN, Berkeley, California, USA, April 26-27*, pages 259–268, 2004.

[43] Y. Okazaki, I. Sato, and S. Goto. A new intrusion detection method based on process profiling. In *SAINT: Symposium on Applications and the Internet*, pages 82–91, 2002.

[44] I. Onat and A. Miri. An intrusion detection system for wireless sensor networks. In *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, pages 253–259, 2005.

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

[46] M. Pirretti, S. Zhu, N. Vijaykrishnan, P. McDoniel, and M. Kandmir. The sleep deprivation attack in sensor networks: Analysis and methods of defense. *International Journal of Distributed Sensor Networks*, 2:267–287, 2006.

[47] Th. S. Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall Communications Engineering And emerging Technologies Series, 2002.

[48] Yogesh Sankarasubramaniam, Özgür B. Akan, and Ian F. Akyildiz. ESRT: event-to-sink reliable transport in wireless sensor networks. In *Proc. 4th ACM MobiHoc*, pages 177–188, Annapolis, MD, June 2003.

[49] S. Shafi. Performance of a beacon enabled IEEE 802.15.4-compliant network. Master's thesis, Department of Computer Science, University of Manitoba, Winnipeg, Canada, 2005.

[50] Snort.org Web Site. http://www.snort.org, Date visited: April,17,2006.

[51] F. Stajano. *Security for Ubiquitous Computing*. John Wiley and Sons, February 2002.

[52] F. Stajano and R. Anderson. The resurrecting duckling: Security issues for Ad-hoc wireless networks. In *Security Protocols, 7th International Workshop Proceedings, Lecture Notes in Computer Science*, pages 172–194, Berlin Heidelberg, 1999. Springer-Verlag.

[53] Ivan Stojmenović, editor. *Handbook of Sensor Networks: Algorithms and Architectures*. John Wiley & Sons, New York, NY, 2005.

[54] D. Subhadrabandhu, S. Sarkar, and F. Anjum. Rida: Robust intrusion detection in ad hoc networks. In *Proc. of the 4th International IFIP-TC6 Networking Conference*, pages 1069 – 1082, 2005.

[55] J. Wang, G. Yang, Y. Sun, and Sh. Chen. Sybil attack detection based on RSSI for wireless sensor network. *WiCom '07: International Conference on Wireless Communications, Networking and Mobile Computing*, pages 2684–2687, 21-25 September 2007.

[56] Weichao Wang and Bharat Bhargava. Key distribution and update for secure inter-group multicast communication. In *SASN '05: Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks*, pages 43–52, New York, NY, USA, 2005. ACM Press.

[57] D. Whiting, R. Housley, and N. Ferguson. Counter with CBC-MAC (CCM). http://www.rfc-archive.org/getrfc.php?rfc=3610, 2003.

[58] Mark Yarvis, Nandakishore Kushalnagar, Harkirat Singh, Anand Rangarajan, York Liu, and Suresh Singh. Exploiting heterogeneity in sensor networks. In *Proc. INFOCOM05*, volume 2, pages 878–890, Miami, FL, March 2005.

[59] F. Ye, H. Luo, S. Lu, and L. Zhang. Statistical en-route filtering of injected false data in sensor networks. *IEEE Journal on Selected Areas in Communications*, 23(4):839–850, April 2005.

[60] F. Ye, H. Yang, and Z. Liu. Catching moles in sensor networks. *27th International Conference on Distributed Computing Systems, ICDCS '07*, pages 69–69, 25-27 June 2007.

[61] X. Brian Zhang, Simon S. Lam, Dong-Young Lee, and Y. Richard Yang. Pro-

tocol design for scalable and reliable group rekeying. *IEEE/ACM Trans. Netw.*, 11(6):908–922, 2003.

[62] Y. Zhang and W. Lee. Intrusion detection in wireless ad-hoc networks. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 275–283. ACM Press, 2000.

[63] S. Zhu, S. Setia, S. Jajodia, and P. Ning. An interleaved Hop-by-Hop authentication scheme for filtering false data injection in sensor networks, May 2004.