Edith Cowan University
Research Online

Australian Information Warfare and Security Conference

Security Research Institute Conferences

2011

Penetration of ZigBee-based wireless sensor networks

Michael N. Johnstone
Edith Cowan University

Jeremy A. Jarvis
Edith Cowan University

Originally published in the proceedings of the 12th Australian Information Warfare and Security Conference, Edith Cowan University, Perth Western Australia, 5th -7th December, 2011
This Conference Proceeding is posted at Research Online.
http://ro.ecu.edu.au/isw/45

PENETRATION OF ZIGBEE-BASED WIRELESS SENSOR NETWORKS

Michael N. Johnstone and Jererny A. Jarvis
School of Computer and Security Science
Edith Cowan University, Perth, Western Australia
m.johnstone@ecu.edu.au; j.jarvis@our.ecu.edu.au

Abstract

Wireless Sensor Networks are becoming popular as a simple means of collecting data by public utilities, motor vehicle manufacturers and other organisations. Unfortunately the devices on such networks are often insecure by default, which presents problems in terms of the integrity of the data provided across those networks. This paper explores a range of attacks that were successful on a network consisting of nodes using the ZigBee protocol stack and proposes defences that can be put in place to circumvent these attacks thus leading to more secure systems and increasing user confidence.

Keywords

Wireless Sensor Network, Vulnerability, ZigBee, 802.15.4 Standard.

INTRODUCTION

The Internet Engineering Task Force considers a vulnerability to be "A flaw or weakness in a system's design, implementation, or operation and management that could be exploited to violate the system's security policy." (RFC 2828, 2000, p190). A threat is "a potential violation of security.", according to Bishop (2005, p4). Therefore an exploit is an attack that takes advantage of a vulnerability therefore realising a threat. The purpose of this paper is to test threats by initiating attacks on vulnerabilities in a wireless sensor network system, specifically ChipCon CC24XX series sensors produced by Texas Instruments.

The prime tenets of information security are confidentiality, integrity and availability. Confidentiality means that only authorised users have correct access to assets (assets in this case means the data transmitted over a wireless network). Integrity can be described as ensuring that only authorised users can alter data in defined ways. Availability is a guarantee that authorised users are able to access data in a timely fashion.

There are well-known classes of attack for each of these tenets. For example, brute force code-breaking could be an attack on confidentiality; the "man in the middle" attack is an attack on integrity; and denial-of-service is an attack on availability. This paper tests the first two classes of attack by experiment. Attacks based on denial-of-service are not covered here because they can usually be detected more easily than the other classes of attack.

The structure of this paper is as follows: First, wireless sensor networks as a means to collect information are introduced. Next, the experimental methods and tools used to craft the attacks are described. Following this is an analysis of the results. The paper concludes by providing a mitigation strategy for those specific attacks that were successful.

WIRELESS SENSOR NETWORKS

Wireless networks are different from their wired counterparts in that wireless system often have dynamic topologies, are unprotected from other signals sharing the medium and communicate over a medium that is significantly less reliable than wired networks (IEEE, 2007). A wireless sensor node consists of a microprocessor, a radio frequency transmitter/receiver, a power supply (a battery or sometimes a solar cell) and a sensor of some type.

Sensors can be microphones, still cameras, video cameras, pressure, temperature and movement sensors. Such sensors could also be coupled together in that a movement sensor might trigger a dormant camera. This would have benefits in terms of power consumption as the lowest power device (the movement sensor) is on continuously, but the high power consumption/high bandwidth device is only activated when there is something of interest to detect.

Two important properties of a wireless network are the data transfer rate and the maximum distance between transmit/receive nodes. IEEE 802.11n has a transfer rate of approximately 248Mbps and a range of 250m under ideal conditions. This is perhaps no surprise as 802.11n was designed to address the speed limitations of prior

802.11 standards. By comparison, 802.15.4 has a transfer rate of between 40-250Kbps and a range of 75m, again under ideal conditions. It would appear that 802.15.4 is at a significant disadvantage compared to 802.11.x, but this is not necessarily the case as the shorter range requires less power.

Following the development of the 802.15.4 standard, several bodies such as the ZigBee Alliance, were formed to promote the development of low-power networks in various application domains. The Zigbee standard is not the same as 802.15.4 (with which it is often confused), but is a protocol stack built on top of the IEEE standard (see figure 1). Strictly speaking, Zigbee now uses the 802.14.5 standard, but the distinction is not significant for this discussion. Zigbee-based devices offer a potential solution where many sensors will be deployed in sub-optimal conditions, however they suffer from two drawbacks. First, power consumption is an issue as the devices are often small and powered by batteries and second, security is a potential issue.

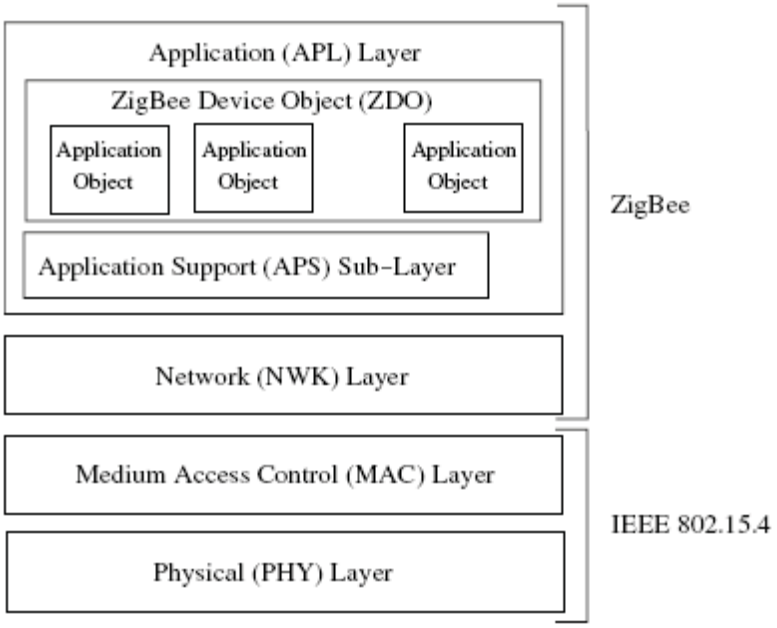

Figure 1: The Relationship between 802.15.4 and Zigbee (Akyildiz and Vuran, 2010, p6).

Most wireless sensors operating in idle mode consume power at a rate approximately equal to the power consumed in receive mode due to increased communication with a base station therefore power consumption is a significant issue for these low-power devices.

Shutting down a sensor may optimise power use compared to leaving it in idle mode when it is not transmitting or receiving. Unfortunately, once a sensor is turned off, it cannot receive any messages from its neighbour sensors (because it is effectively disconnected from the network). Further, continually turning the transceiver on and off also has an energy overhead. It might be more efficient for a sensor to be able to adjust its sampling rate and communications frequency to save power.

The 802.15.4 standard handles security at the MAC layer (see figure 1). This in itself is not necessarily a problem, however it must be enabled and is disabled by default. Figure 2 shows that two bytes are reserved for a CRC which is computed on the data bytes. This suggests that the datagrams are safe from tampering because the CRC would not match when recomputed at the endpoint. This is not necessarily so. Sastry and Wagner (2004) point out that 802.15.4 networks can suffer from security breaches in three areas, viz: IV (nonce) management, key management and integrity protection. Considering just the last area for brevity's sake, Sastry and Wagner claim that CRCs are insufficient for integrity protection and favour strong encryption instead (which is available on 802.15.4 devices). The main issue, according to Sastry and Wagner (2004, p39) is that all "*...of the attacks center on the fact that in the course of modifying the ciphertext, the adversary can construct appropriate modifications to the CRC so that the receiver accepts the packet. Researchers have discovered unauthenticated encryption vulnerabilities in…802.11…that compromise not only integrity but also confidentiality.*". This is still a problem as noted by Aggélou (2009) and Boudhir et al. (2010).

1 byte	2 bytes	1 byte	0-10 bytes	0-10 bytes	variable	2 bytes
Len.	Flags	Seq. #	Dest. Address	Source Address	Data	CRC

Figure 2: An IEEE 802.15.4 Datagram (Adapted from Sastry and Wagner, 2002).

Wireless sensor networks appear to show some promise but there exist some security problems to be solved in terms of power consumption of the network nodes (availability) and the integrity and potentially confidentiality of the data sent across the network. The next section describes the attacks crafted to expose the vulnerabilities of a specific wireless sensor network.

ATTACK PREPARATION

The purpose of this research is to assess what risks are inherent in the location calculation system designed by Texas Instruments (TI) which is implemented by the CC2400 series chipsets. In this section, the materials used are described, the TI nomenclature explained and the attack schemes explained.

Hardware and Software Used

The hardware used in the attacks was: a laptop (Windows 7); a SmartRF04EB Evaluation Board with a CC2430 chip flashed with the location dongle (see below) hex file; and a CC2430DB used as a capture device.

Software required was the TI SmartRF Protocol Packet Sniffer; TI SmartRF Studio 7 v1.2.3 and the SmartRF Flash Programmer, (used to flash the devices memory with the appropriate firmware).

Wireless Network Nodes

The two dimensional location system designed by TI utilises the Zigbee protocol over the 802.15.4 standard and is designed to transmit small amounts of data over a mesh-like network of nodes denoted as one of:

- A reference node: A node that is given an X, Y position and is utilised by a blind node when it (the blind node) wishes to perform a location calculation;
- A Blind Node: A node that moves around the system and performs location calculations; or
- A Location Dongle: Connected to the laptop and is used to receive location data to be displayed on-screen.

Specific Attacks

Initial tests will be performed to obtain packet data using various sniffing software and hardware configurations. The process will help define the later attack patterns and give initial insight into what can be obtained.

Next, the system will be attacked by supplying fraudulent packet data by introducing a false reference node and then a false blind node into the system. This will be designed as a replay attack and the success will be determined by monitoring the effect it has while the system is collecting location data.

RESULTS AND ANALYSIS

Test#1: Use TI provided hardware and software to trial basic packet-sniffing.

The objective of the test is to obtain packet data using the CC2430DB as a receiver, and the CC2431 board as a transmitter. The test used the TI packet sniffing software and SmartRF Studio 7's packet transmitter with a SmartRF04EB Evaluation Board with a CC2430 chip flashed with the location dongle hex file and a CC2430DB as a capture device.

As figure 3 shows, the TI-provided packet sniffer has no problem picking up datagrams sent across the wireless network.

File Help

ZigBee 2007/PR

P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source PAN	Source Address	Encrypted MAC payload	LQI	FCS
RX 1	+0 =0	22	Type Sec Pnd Ack.req PAN_compr DATA 1 0 0 0	0x41	0xFD1B	0x4FB8	0xF668	0x7B72	14 99 CD D3 0D F0 44 3A B4	220	OK
RX 2	+101987 =101987	22	Type Sec Pnd Ack.req PAN_compr DATA 1 0 0 0	0x41	0xFD1B	0x4FB8	0xF668	0x7B72	14 99 CD D3 0D F0 44 3A B4	220	OK
RX 3	+103147 =205134	22	Type Sec Pnd Ack.req PAN_compr DATA 1 0 0 0	0x41	0xFD1B	0x4FB8	0xF668	0x7B72	14 99 CD D3 0D F0 44 3A B4	220	OK
RX 4	+102869 =308003	22	Type Sec Pnd Ack.req PAN_compr DATA 1 0 0 0	0x41	0xFD1B	0x4FB8	0xF668	0x7B72	14 99 CD D3 0D F0 44 3A B4	220	OK
RX 5	+102009 =410012	22	Type Sec Pnd Ack.req PAN_compr DATA 1 0 0 0	0x41	0xFD1B	0x4FB8	0xF668	0x7B72	14 99 CD D3 0D F0 44 3A B4	220	OK
RX 6	+101947 =511959	22	Type Sec Pnd Ack.req PAN_compr DATA 1 0 0 0	0x41	0xFD1B	0x4FB8	0xF668	0x7B72	14 99 CD D3 0D F0 44 3A B4	228	OK
RX 7	+102030 =613989	22	Type Sec Pnd Ack.req PAN_compr DATA 1 0 0 0	0x41	0xFD1B	0x4FB8	0xF668	0x7B72	14 99 CD D3 0D F0 44 3A B4	220	OK

Figure 3: TI Packet Sniffer Data Capture Test.

Test#2: Assessing packet loss with the CC2430DB.

The objective of the test is to evaluate the effect of interference from other devices and node spacing on packet-sniffing.

The test used the TI packet sniffing software, the Chipcon Z-Location Engine and the Texas Instruments Flash Programmer (used for initial Flashing of devices). The hardware used was a SmartRF04EB evaluation board, several boards with CC2430 chips flashed with the reference node hex file, a board with the CC2431 chip flashed with the blind node hex file and a CC2430DB as a capture device.

The packet sniffer was used to gauge the effect of interference from other devices by conducting a series of tests inside a building (with the nodes at one, two and four metre spacing), then performing the same tests outdoors. The tests highlighted several unexpected problems, viz: the nodes would move around (on-screen) when they should have been stationary and there was some packet loss. These faults could be attributed to interference or the distances between the nodes being sub-optimal for effective transmission. As the tests progressed, a single variable (distance or site) was changed with the other held constant to attempt to identify the problem. These changes did not affect the output enough to be considered successful.

The tests did, however, provide a side-benefit in that the data captured outlined the process that the system uses to perform a location calculation. The pattern that developed was:

1. The sniffer receives a few packets, then a return message from the blind node when requested to send position data with a cluster ID of 0x0014. This packet has a larger network payload than the 13 bytes expected from the blind node as a location calculation so it not a transfer of data from a reference node. is a return message from the blind node when requested to send position data.
2. Next there are a few similar packets, then a blast message (cluster ID 0x0019) from the blind node (a packet that is transmitted to any listening devices). This is a small radius message and come before a location calculation request. It is a small range packet designed to limit the number of reference nodes that are recipients
3. Next a single packet (cluster ID 0x0011) which is a reference nodes position request. The packet is quite small and its destination address is 0xFFFF which suggests the blind node is selecting reference nodes automatically
4. There are two single packets (cluster ID 0x0011) sent identified as an XY-RSSI Request, used to request an XY calculation response from the reference node
5. Next, (cluster ID 0x0012) an XY-RSSI Response, which broadcasts the blind node's X position, its Y position and its RSSI average value
6. Finally there are some smaller packets sent, a small response message and a return message.

This information could be used by an attacker to spoof the system with packet data similar to what it is expecting. It might also be possible to wait and read a blast from the blind node and injects packets at that point.

Test#3: Adding False Nodes.

The objective of the test is to add a false node of each type and observe the system's behaviour towards the false node(s). This can be done for each node type by:

a) Reference Node: Setting the RSSI value to the ideal signal strength and setting different XY coordinates on the GUI;

b) Blind Node: Setting a second blind node and monitoring the effect; and

c) Dongle: Setting a new dongle node both before the system is started and while it's operating.

This test (and later tests) used the same hardware/software configuration as test#2.

The results of test 3a were that the fake node showed up on the system as an existing node. It was initialised well outside of the system boundary for ease of detection on-screen. When an authentic node was moved near the fake node it wasn't changing the location of the blind node. This was somewhat expected, the CC2430DB doesn't have the capability to perform a location calculation. When the RSSI value is changed to 110 (the ideal RSSI response from a node) the blind node location doesn't change.

The blind node signals the CC2430DB to perform a location calculation but when it doesn't receive a signal it simply uses the next node for its calculation. It is assumed the fake node, as it can't send a proper calculation, will send a response of "0xFFFF" for the calculation and the node will then use the other nodes for the calculation. The authentic node's position became less accurate once the fake node was in use which would be due to it having to use a node which it wasn't co-located.

The results of test 3b were that the fake node showed up in the system as a new blind node but it wouldn't pick up any data. As it shows in the system as a blind node, it could possibly be populated with false location data.

The results of test 3c were that as expected, without an application to process the nodes it picks up there isn't any point in obtaining the information.

It would be useful to expand on the code used in the reference node attack. If the code can be modified so that it sends a hard-coded response to the blind node the attack will be able to bypass the need to perform a location calculation. If this can be achieved the credibility of a location calculation would be compromised. The blind node replacement could also be expanded, if the short address of the new blind node matches one of an existing node, the system won't be able to determine from which node it should accept data.

Test#4: Improving Adding False Nodes.

The objective of this test is twofold, first to modify the reference node to provide a location response and second, to modify the blind node so that the response sent is hard-coded, thus whenever a calculation is necessary the node will send the predefined location.

Recall that in the previous test, the reference node was sending out data correctly and interacting with the blind node. It was displayed in the GUI as a new node but wasn't assigned a position because it can't perform a location calculation (therefore it would just give the blind node error packets).

Figure 4 shows that the system accepts the fake nodes. The tests proved that the fake reference node can be convincing if it doesn't have to interact with the blind node. This issue could be addressed by examining the packet data and crafting an appropriate header/payload.

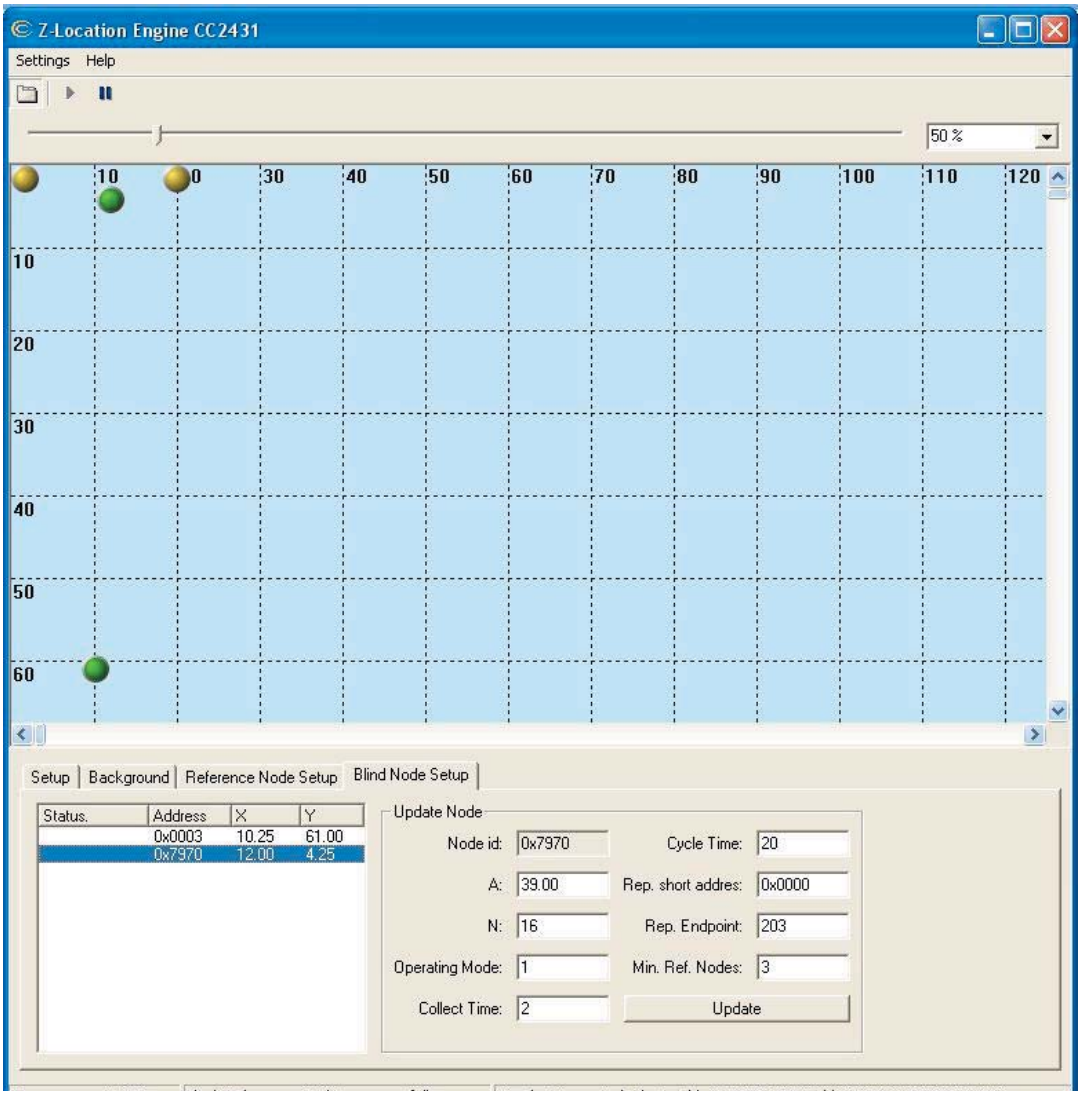

Figure 4: GUI Display of Two Authentic and Two Fake Nodes.

Analysis

Potok et al., (2003, p13) after Abadi (1999) state that it is practically impossible to construct a truly secure information system (or in this context, a wireless sensor network). Communications are secure if transmitted messages can be neither affected nor understood by an attacker, likewise, information operations are secure if information cannot be damaged, destroyed, or acquired by an attacker.

The experiments conducted show that the CC24XX wireless sensor devices are vulnerable. Interestingly, Texas Instruments recommend the following strategies:

- The use of encryption;
- More secure keys;
- A trust centre in which nodes have to be passed by the dongle before they can join the network; and
- Implementing key updates at predetermined intervals.

In fact, the original network keys utilise 128 bit AES encryption, which should have been robust, however, we detected some weaknesses with this implementation:

- The original design does not use the keys;
- The default key provided by the system is widely available. The key is: 0x01, 0x03, 0x05, 0x07, 0x09, 0x0B, 0x0D, 0x0F, 0x00, 0x02, 0x04, 0x06, 0x08, 0x0A, 0x0C, 0x0D;

- Keys are shared over the entire network and stored in flash memory, so if an attacker can obtain the flash data, s/he can obtain the key; and
- There is no documentation that outlines the need to change this key before the system is implemented.

It was remarkable that the security key for the devices was located in f8wconfig.cfg, written in plain text. The code is available online and if the code is not changed when the system is set up the security system is accessible to any attacker who looks for the network key.

A private or public key infrastructure is an obvious solution to the integrity problem, however issues of secure storage for the keys and over-the-air transmission of keys must still be addressed, otherwise the strength (size) of the key does not matter. The issue of key management is perhaps further complicated by the ever-decreasing cost of the hardware required to conduct a brute-force attack. For example, a multi-TeraFLOP GPGPU cluster can be purchased for as little as AUD$10,000.

CONCLUSIONS AND FURTHER WORK

This study sought to explore the vulnerabilities claimed to exist in wireless sensor networks. The basics of wireless sensor networks were explained and the potential vulnerabilities articulated. A series of attacks were planned and executed on a simple network. A solution was proposed for the wireless sensors (strong cryptography), but this has yet to be proven as effective as there are issues of key management still to be solved.

Due to their low cost and robustness, networks of wireless sensors have many potential uses ranging from the mundane such as tagging goods, weather reporting and home automation to more innovative uses of this technology such as home monitoring of individuals in aged care environments and automated meter reading for public/private utilities. Considering just the last application, wireless meter reading is obviously more efficient than traditional walk-about meter reading and thus would be an attractive solution because of its lower on-going costs. Whilst this research has shown that these devices are vulnerable, perhaps the security fears of being able to misrepresent a meter reading are not significant for domestic consumption as it would be expected that the utilities concerned would be immediately aware of any 'outlier' readings. The issue is a larger one of public confidence in such automated systems, especially given the pervasiveness of software systems.

The next step in this research programme is to implement 128 bit AES encryption and then use a GPGPU cluster to perform brute force attacks on the system. In tandem with this approach, later versions of the chipset, specifically the CC25XX series of sensors could also be evaluated.

REFERENCES

Aggélou, G. (2009). *Wireless Mesh Networks*. New York, NY: McGraw-Hill.

Akyildiz, I.F. and Vuran, M. C. (2010). *Wireless Sensor Networks*. Chichester: John Wiley.

Bishop, M. (2005). *Introduction to Computer Security.* Boston, MA: Addison Wesley.

Boudhir, A.A., Bouhorma, M., and ben Ahmed, M. (2010). Multi-Agents Platform for Security in Wireless Sensor Networks. *IJCSNS International Journal of Computer Science and Network Security*, 10(10), October, pp. 198-201.

IEEE (2007). IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. New York: IEEE Computer Society.

Potok, T., Phillips, L., Pollock, R., Loebl, A. and Sheldon, F. (2003). Suitability of Agent-Based Systems for Command and Control in Fault-tolerant, Safety-Critical Responsive Decision Networks. *Proc. 16th Int'l Conf. On Parallel and Distributed Computing Systems*, Aug. 13-15,2003 Reno NV.

RFC 2828 (2000). *Internet Security Glossary*. Internet Engineering Task Force. Retrieved September 22, 2009, from http://www.ietf.org/rfc/rfc2828.txt

Sastry, N. and Wagner, D. (2004). Security Considerations for IEEE 802.15.4 Networks. *ACM Workshop on Wireless Security (WISE 04)*, pp. 32–42.