Defensive Rekeying Strategies for Physical-Layer-Monitored Low-Rate Wireless Personal Area Networks

Benjamin Ramsey, Barry Mullins

Chapter 5

DEFENSIVE REKEYING STRATEGIES FOR PHYSICAL-LAYER-MONITORED LOW-RATE WIRELESS PERSONAL AREA NETWORKS

Benjamin Ramsey and Barry Mullins

Abstract ZigBee networks are integrating rapidly into critical infrastructures such as the smart grid and public health centers. Numerous ZigBee-based smart meters have been installed in metropolitan areas and hospitals commonly employ ZigBee technology for patient and equipment monitoring. The associated ZigBee networks transport sensitive information and must be secured against exfiltration and denial-of-service attacks. Indeed, novel tools that exploit and disrupt ZigBee networks are already under development. Security monitors that can uniquely identify nodes by their radio frequency characteristics can be a valuable countermeasure if implemented in a practical manner. This paper investigates rekeying in response to suspected malicious devices that may be internal or external to a ZigBee network. It extends prior discussions of practical physical layer monitor implementation, and introduces a novel backward-compatible ZigBee message obfuscation technique based on preamble modifications. Experimental results demonstrate that common wireless ZigBee sniffers can be thwarted with 100% effectiveness without reducing packet reception to specific transceiver models.

Keywords: ZigBee networks, security, RF fingerprinting, cyber-physical systems

1. Introduction

ZigBee networks provide low-rate, low-power and low-cost wireless connectivity through standards that build upon the IEEE 802.15.4 low-rate wireless personal area network (LR-WPAN) physical (PHY) and medium access control (MAC) specifications [13]. ZigBee Smart Energy, Building Automation and Health Care standards have enabled ZigBee networks to become significant components in critical infrastructures around the world, including tens

of millions of utility meters with bidirectional communications incorporated in advanced metering infrastructures [23]. Critical ZigBee networks form cyber-physical systems, where malicious activity on the networks affects the physical behavior of appliances and electrical systems. Public health networks employing ZigBee technology are also common in civilian and military hospitals. Disruptions of these networks can impact medical equipment and patient monitoring, possibly endangering lives.

The ZigBee security architecture relies on the safekeeping of symmetric keys to implement message confidentiality, message integrity, and device authentication. While the small size and low complexity of ZigBee devices make them effective to deploy in large numbers, these traits also result in tight constraints on device memory and computations. A single network key (NK) is shared by every device in a ZigBee network, although device-to-device confidentiality is also possible using link keys (LK) at the application layer. Small, inexpensive wireless sensors are unlikely to have robust defenses against theft and tampering, resulting in physical vulnerabilities to key confidentiality [20]. Key extraction from first- and second-generation ZigBee chips has been shown to be relatively straightforward [9], and inexpensive tools have been developed for locating ZigBee devices [15, 18]. Keys may also be compromised through social engineering or keys may intercepted (if transmitted to end devices without encryption) by open source tools such as KillerBee [24] and Api-do [21].

The ZigBee specification is dependent on symmetric cryptography, which precludes key distribution without a central authority called the trust center (TC). The computational complexity of symmetric cryptography is lower than that of public key cryptography. The tradeoff, however, is a significantly more challenging key management process.

Previous research has investigated the application of public key cryptography to ZigBee devices and networks [2, 10–12, 17]; this research has contributed to a provision in the ZigBee Smart Energy Profile for secure LK establishment through certificates signed by a certificate authority [25]. The alternative key establishment procedure is based on a shared master key (MK) used to derive the LK. If the MK is not preloaded on the end device, the TC broadcasts it without encryption, potentially compromising the subsequent LK establishment.

Methods for detecting rogue devices in ZigBee-type networks are an active area of research, including anomalous-behavior-based fingerprinting [14] and radio frequency (RF) device fingerprinting [6, 8, 19]. Given the threats to ZigBee symmetric keys, an efficient and secure redistribution of keys must occur to thwart a known eavesdropper or active rogue device on the network.

This paper examines rekeying strategies for a compromised ZigBee network. In particular, it investigates how PHY-based monitoring systems proposed in [6, 8, 19] can be integrated in ZigBee networks. Also, the paper describes a novel method for protecting sensitive ZigBee traffic from eavesdroppers using modified PHY preambles.

2. Key Distribution

Key distribution mechanisms in wireless sensor and control networks have four general requirements [3]:

- **Scalability:** The key distribution mechanism must remain practical for a large increase in the number of network devices.

- **Efficiency:** The key distribution mechanism must involve limited memory usage, computational complexity and communications complexity.

- **Probability of Key Sharing:** Key sharing among devices must be limited to what is necessary to implement the desired network functionality.

- **Resilience:** The key distribution mechanism must be resistant to node tampering and theft. In particular, security credentials that are extracted from a device or eavesdropped should not reveal security information of other devices in the network.

Note that these four requirements are generally mutually exclusive. Thus, every key distribution solution must make trade-offs as appropriate.

2.1 ZigBee Nodes and Topologies

The IEEE 802.15.4 standard for wireless MAC and PHY specifications defines two primary device types: (i) full function devices (FFDs); and (ii) reduced function devices (RFDs) [13]. Mains-powered FFDs are always actively listening on the network, whereas RFDs are battery powered and primarily operate in the sleep mode, waking up only to check for pending messages or periodic updates.

ZigBee specifies three node classes within the IEEE 802.15.4 construct: (i) ZigBee coordinator (ZC); (ii) ZigBee router (ZR); and (iii) ZigBee end device (ZED). The ZCs and ZRs must be FFDs, while ZEDs can be either FFDs or RFDs. There can only be one ZC per WPAN, and it is responsible for establishing the network, allocating network layer addresses and routing traffic. The network fails without the ZC. ZRs extend the wireless range by routing messages between their child RFD ZEDs using multi-hop configurations, such as the cluster tree and mesh topologies is shown in Figure 1. Note that the star topology is shown for completeness; however, it does not support multi-hop communications. In a cluster tree topology, ZEDs have no children and can only communicate with the ZC and other ZEDs through their parent ZR. The ZigBee stack profile `0x01` limits the number of children for each ZR to $N_c = 20$, six of which can be ZRs. The ZigBee PRO specification (stack profile `0x02`) increases this limit to $N_c = 254$ children per ZR. Mesh topologies are only allowed under ZigBee PRO, and permit FFD ZEDs to communicate directly with each another to form a self-healing network.

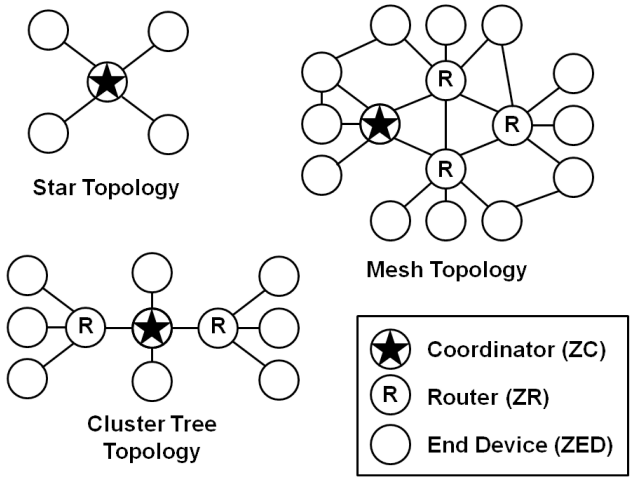

Figure 1. ZigBee LR-WPAN topologies.

2.2 Stationary Networks

The term "stationary network" refers to a network in which the logical and physical topologies are both fixed. It includes the star and cluster tree topologies. ZigBee networks for building and home automation, utility meter reading, industrial control and environmental sensing typically utilize stationary networks. Unlike mobile nodes, stationary nodes can be mains powered and always active.

The ZigBee Smart Energy Profile describes standard practices suitable for securing stationary networks [26]. In order to join a network, every new device must pass an administrator-directed commissioning process. The commissioning process is not typically automated because neighboring systems have no way to identify the devices that should be associated with one another without administrator guidance [4]. At no point should any cryptographic keys be transmitted in the clear, a recommendation that is mirrored in [16]. At a high level of abstraction, a typical device commissioning process involves the following steps [26]:

- The ZC is informed via out-of-band means (e.g., remote login, handheld controller or phone call to a service center) that a new device must be added to the network.

- The network enters into a permit joining ON state for a short period of time (e.g., 10 seconds).

- The installer, with physical access to the new device, presses a button or navigates a menu that instructs the new device to attempt to join the network through a join procedure.

- The new device attempts to join the network. After the new device is verified as being authentic by its MAC address and/or secret key, it receives new cryptographic keys and successfully joins the network.
- The installer receives visual feedback from the new device that the network join was successful.
- The new device may now operate in the network.

The short window of the commissioning process protects against rogue devices from joining the network. This process is also robust against tools like KillerBee's `zbassocflood` that transmit numerous association requests from spoofed MAC addresses to exhaust the network address pool in a denial-of-service attack.

Smart Energy Profile standards identify three types of keys: (i) NK; (ii) application layer LKs shared by pairs of devices; and (iii) trust center link keys (TCLKs), which are LKs that are shared by a device and the TC. The NK is common to all devices in a network and protects management and control communications [26]. LKs provide end-to-end confidentiality. The TC periodically refreshes the NK to protect the network from cryptographic attacks from outside the network (e.g., key cracking). The NK must also be refreshed in response to a suspected rogue device within the network. Rekeying is thus performed point-to-point from the TC to every trusted device on the network using TCLKs for confidentiality. The complexity of this action amounts to $O(n)$, where n is the number of devices in the network. After every trusted device receives the new NK, the broadcasted `Switch Key` command instructs all devices to simultaneously switch to the new NK. The TC also revokes any LKs that were previously established to a rogue device.

2.3 Mobile Networks

The term "mobile network" refers to a network in which the logical and physical topologies change unpredictably. Examples include medical patient monitoring and inventory tracking systems. Mobile devices may drift out of communication range of other network nodes long enough to require a network rejoin, which cannot be performed manually as with stationary networks. Physical security is also significantly more difficult to maintain for mobile devices.

The ZigBee Health Care Profile [27] provides key distribution recommendations for mobile networks. As with stationary networks, key delivery in the clear is prohibited. Instead, NK and LK distribution proceeds via mandatory TCLKs. Mobile nodes are also much more likely than stationary nodes to be battery powered; therefore, the nodes spend significant time in the sleep mode. If a network layer rekey occurs while a medical device is asleep, it will experience a delay in reporting its sensing data because it must first receive the new NK. To minimize such delays, devices should check periodically if they have the current NK [27]. Rekeying requires all end devices to expend more of their finite energy supply than is functionally necessary, so the rate of periodic

rekeying must be set based on battery longevity requirements. Nevertheless, rekeying in response to a suspected rogue device is essential.

2.4 Alternative Key Distribution Methods

Numerous key distribution mechanisms exist for ZigBee-like wireless sensor networks and *ad hoc* networks [3]. Each key distribution mechanism results in different computational, energy and memory burdens being placed on network nodes. The ZigBee protocol stack requires that the NK be shared by all the devices in a network. This fundamental requirement necessitates that NK rekeying be performed on every network device if the NK is suspected to be compromised.

The TC must revoke all point-to-point LKs established between a benign device and the suspected rogue device. In this case, techniques that reduce the number of required LKs also reduce the number of required LK revocations. For example, the upper limit for a ZigBee network is $2^{16} \approx 65{,}000$ nodes, requiring $n(n-1)/2 \approx 2$ billion LKs for full connectivity. In a hierarchical keying system [22], the number of required LKs reduces to $log_2(n) + 2 = 18$ LKs. However, full connectivity is rare in practice, and LKs are frequently limited to communications between end devices and their associated data aggregation nodes. Note that spatial clustering methods, including the Hubenko Architecture [1], are not relevant to ZigBee LK rekeying because LKs secure unicast traffic rather than multicast traffic. Indeed, a compromised node reveals nothing about the LKs in use by any two other nodes, so TC-directed LK revocation is sufficient for threat mitigation.

3. ZigBee Air Monitor Integration

Recent research [8, 19] has proposed a ZigBee air monitor (ZAM) system to secure networks by observing wireless transmission characteristics to augment bit-layer security mechanisms. Dubendorfer, *et al.* [8] have demonstrated the feasibility of unique device-level identification under realistic indoor office conditions. The next three sections explore how ZAMs can integrate into and defend critical ZigBee networks organized in star, cluster tree and mesh topologies.

3.1 Star Topology

The star topology is the least complex ZigBee network topology. In this topology, end devices communicate solely with the ZC. ZigBee PRO allows a ZR or ZC to have up to $N_c = 254$ child nodes, but such a dense network can experience significant transmission congestion. A best practice is to limit the number of devices in an area to utilize spatial reuse [5]. The utilization ratio U is given by:

$$U = \frac{4 < Density < 16}{Total\ Devices} \tag{1}$$

where the density (i.e., number of devices per unit area where the unit used is the square of their reliable range) is between four and sixteen. A star network should, therefore, be limited to approximately fourteen end devices (15 devices − 1 coordinator).

The ZAM is co-located with the ZC and observes every transmission, sending a packet reject notice to the ZC through a wired channel if the RF fingerprint of the packet does not closely match the fingerprint stored in the ZAM for the claimed origin device [19]. The ZigBee application support (APS) layer permits up to two retries for each packet. False rejections of packets from trusted devices follow a binomial distribution with the probability p of successful packet delivery given by:

$$p = 1 - (\rho + PER)^{1+r} \tag{2}$$

where ρ is the probability of false rejection, PER is the packet error rate for the network and r is the number of total retries by the APS layer. For $\rho = 0.2$, $r = 2$ and $PER = 0.1$, the probability of benign packet delivery exceeds 99%. Dubendorfer, *et al.* [8] report rejection rates of $\rho \leq 0.2$ for seven like-model devices at $SNR = 10$ dB. For star topology networks with mains-powered devices, the energy spent on retransmission due to false packet rejections is negligible, particularly in a neighborhood advanced metering infrastructure where utility usage and pricing updates occur a few times per hour. A low utilization ratio as mentioned above mitigates network congestion introduced by frequent packet retransmissions.

The ZAM must make verification decisions within a short time, constrained by the transmission timeout settings at the APS layer. A typical value for the unicast timeout is $t = 1.6$ seconds per try, for a total of $t_{total} = 4.8$ seconds [7]. Computational and memory usage requirements increase for the ZAM as the number of nodes increases, but the unicast timeout places a strict upper limit on the total packet accept/reject response time.

3.2 Cluster Tree Topology

The cluster tree topology extends the scale of a stationary ZigBee network beyond that of the star topology by leveraging spatial reuse and a data aggregation backbone of ZRs. The utilization ratio defined in Equation (1) still applies, limiting the practical number of child nodes per ZR, while the total number of nodes can increase substantially.

ZAMs co-located with every ZR provide oversight of all network traffic and provide the same per-packet rejection feedback for child nodes in their cluster as described in the star topology. The device fingerprint database in each ZAM remains at the approximate scale of the star topology scenario. No sharing of fingerprint information between ZAMs is necessary because the network nodes do not stray from their ZRs.

3.3 Mesh Topology

The mesh topology poses the greatest security challenge among the three topologies. ZAMs distributed throughout the mesh network physical topology observe all network traffic, including point-to-point traffic between ZEDs. ZAMs co-located with the core ZRs may be sufficient for full network observation, otherwise additional ZAMs must be co-located with sufficient FFD ZEDs to cover the outermost traffic. FFDs are mains powered, so power is also available for the co-located ZAMs.

FFD ZRs and ZEDs, with co-located ZAMs, receive wired per-packet feedback as in the star and cluster tree topologies. However, the logical network topology is variable and an FFD ZED unattached to a ZAM can receive a packet from a neighboring ZED. In this case, the receiving ZED requests a packet accept/reject feedback from the observing ZAM using the ZigBee network. An alternative solution is to require all packets to traverse FFDs with co-located ZAMs while still allowing the logical topology to change over time. The variable mesh topology increases the number of fingerprints that each ZAM must store to verify true packet origin, with an upper bound of n representing a fingerprint profile for every network node.

4. PHY-Based Sensitive Message Obfuscation

Although they are functionally consistent, transceiver implementations vary between manufacturers due to their design, components used and other unique attributes. These variations provide an opportunity to develop unique signatures for fingerprinting devices based on their implementation characteristics. This section describes a method for preventing common wireless sniffers from detecting ZigBee packets, while retaining their compatibility with available hardware. The method involves varying the length of the IEEE 802.15.4 preamble and measuring response differences that result from manufacturer implementations.

4.1 Methodology

The IEEE 802.15.4 beacon request is a standard command used by a device to locate all coordinators within transmission range. Replies by the coordinators are compulsory (i.e., unicast frames with the acknowledgment flag set garner replies from individually addressed devices).

A Tektronix TDS6124C digital storage oscilloscope was used to receive and store a single beacon request using a Ramsey LPY2 log periodic antenna with 6 dBi gain. The sampling rate R_s was 1.25 GS/s and the collection length l_c was 1 ms, which was long enough to capture the entire $l = 512$ μs beacon request. The collection yielded a vector of 1.25 million data points. All ZigBee transmissions at $f_c = 2.4$ GHz begin with a $l = 128$ μs preamble of $l_p = 32$ bits represented by eight O-QPSK symbols. As shown in Figure 2, when the

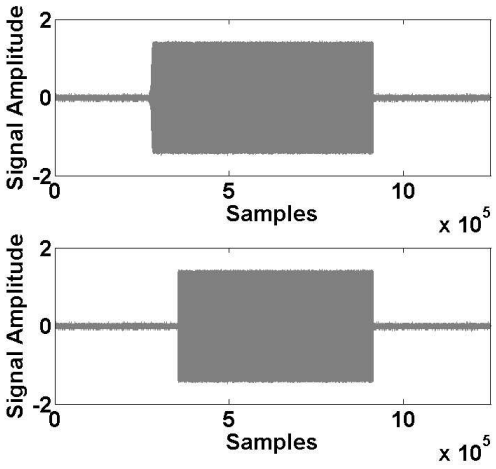

Figure 2. Beacon requests with standard (top) and shortened preambles (bottom).

first half of the preamble (four O-QPSK symbols) is absent, the entire beacon request transmission shortens by 12.5%.

A Tektronix AWG7102 arbitrary waveform generator was used to replay the original and modified beacon requests through the same Ramsey LPY2 log periodic antenna used for signal collection. The center frequency remained constant at $f_c = 2480$ MHz to mitigate interference from neighboring IEEE 802.11 networks during the experiments. The received signal strength was approximately -79 dBm at the target devices and the response behavior remained constant through differing distances and signal strengths. All the hardware devices correctly replied to the original beacon request, but not all the devices replied to modified beacon requests. This indicates that some transceiver types cannot synchronize to the shortened preambles and are unable to interpret the message contents.

Figure 3. Atmel RZUSBStick (left) and Microchip ZENA wireless adapter (right).

4.2 ZigBee Sniffer Hardware

Two widely available hardware platforms for wireless ZigBee sniffing are the Atmel RZUSBStick and the Microchip ZENA wireless adapter (Figure 3). Both platforms are inexpensive, contain a USB connector, include support for real-

time viewing of ZigBee packets and can save the captured traffic to a local file. A limiting factor common to both devices is the lack of support for an external antenna.

The RZUSBStick includes an Atmel AT86RF230 transceiver with -101 dBm receiver sensitivity and maximum transmit power of $P_t = 3$ dBm. Atmel Wireless Studio is the associated free application for wireless sniffing. Alternatively, open source KillerBee software and firmware fully support the RZUSBStick, enabling arbitrary ZigBee packet generation, key sniffing, denial-of-service attacks and transmitter positioning.

The ZENA wireless adapter includes a Microchip MRF24J40MA transceiver module with -94 dBm receiver sensitivity. The Microchip Wireless Development Studio, which controls wireless sniffing on the adapter, may be downloaded free of charge.

4.3 Results and Analysis

The six device types considered in this research included the two sniffers (Atmel AT86RF230, Microchip MRF24J40MA) and four other transceivers configured as coordinators (XBee XBP24CZ7PIS, Freescale MC13213, Texas Instruments (TI) CC2420 and Jennic JN5148). Beacon requests with shortened preambles emanated from the arbitrary waveform generator toward the six device types simultaneously. Beacon replies occured within milliseconds and were staggered to avoid collisions using the PHY Carrier Sensing Multiple Access Collision Avoidance Algorithm. In the case of the two sniffers, correctly interpreted packets appeared in the display windows of the sniffing control software. The four transceiver models configured as ZigBee coordinators replied to the correctly-interpreted beacon requests, and the two sniffers recorded all the replies for post-experiment analysis. Each preamble modification was transmitted a total of 100 times – five repetitions of 20 transmissions each.

The mean packet reception exceeded 98% for all device types when the beacon requests were transmitted with standard preambles. When the standard eight O-QPSK symbol preamble was shortened to five symbols, two of the six transceiver models began experiencing difficulty interpreting the packets. Figure 4 shows the mean ZigBee packet reception rates at a 95% confidence interval when only 5/8 of the preamble was present. The Atmel transceiver received a mean of 40% of the beacons, while the XBee transceiver received a mean of 87% of the beacons. No significant loss in packet reception occured for the Microchip, Freescale, TI and Jennic transceivers.

Figure 5 shows the mean ZigBee packet reception rates at a 95% confidence interval when only 4/8 of the preamble was present. In both cases, the sniffers were unable to interpret any of the packets. The XBee transceiver was thwarted, while the Freescale, TI and Jennic transceivers experienced no significant difficulty receiving packets. The Freescale MC13213 could not interpret packets with preambles shorter than four O-QPSK symbols, while the TI CC2420 and Jennic JN5148 still received all the packets.

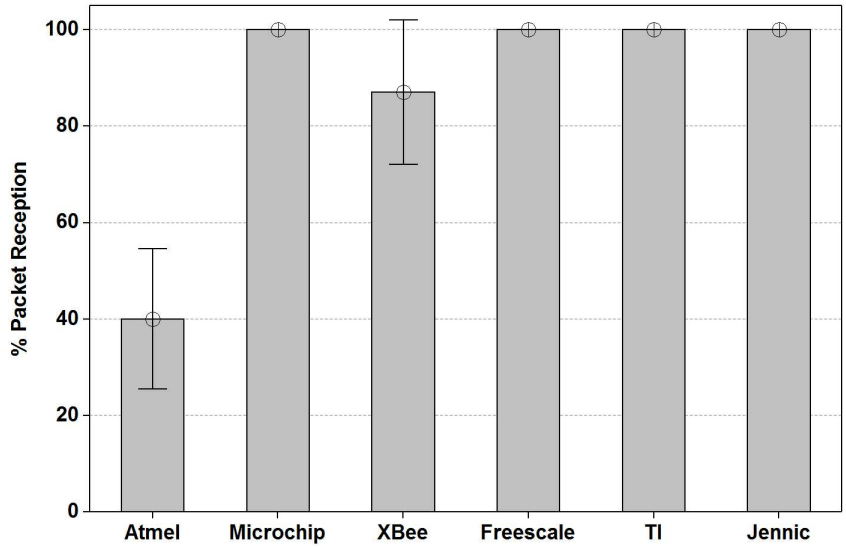

Figure 4. Packet reception rate per device type (5/8 preamble).

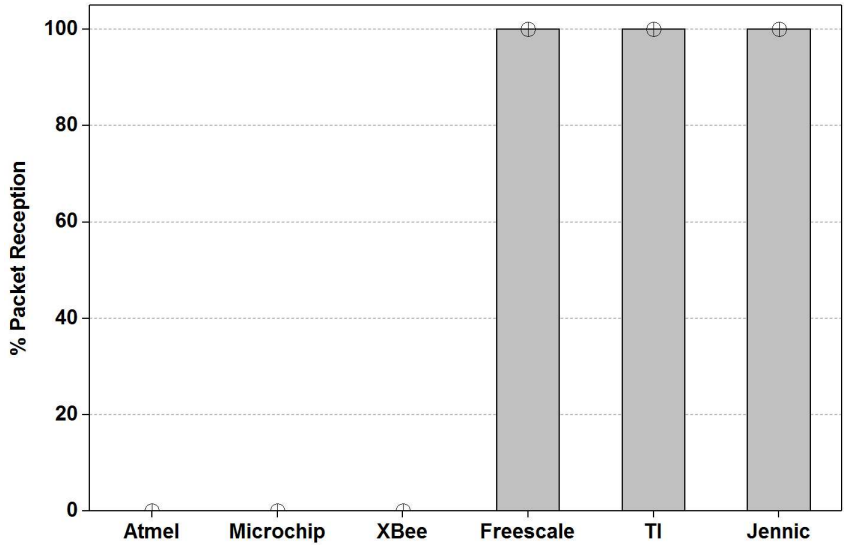

Figure 5. Packet reception rate per device type (4/8 or 12/8 preamble).

Figure 6 shows the mean ZigBee packet reception rates at a 95% confidence interval when only 2/8 of the preamble was present. Consistent with the experimental trend, only the TI and Jennic hardware correctly interpreted packets. The Jennic JN5148 continued to interpret packets without difficulty, while the TI CC2420 failed to interpret approximately half of the packets.

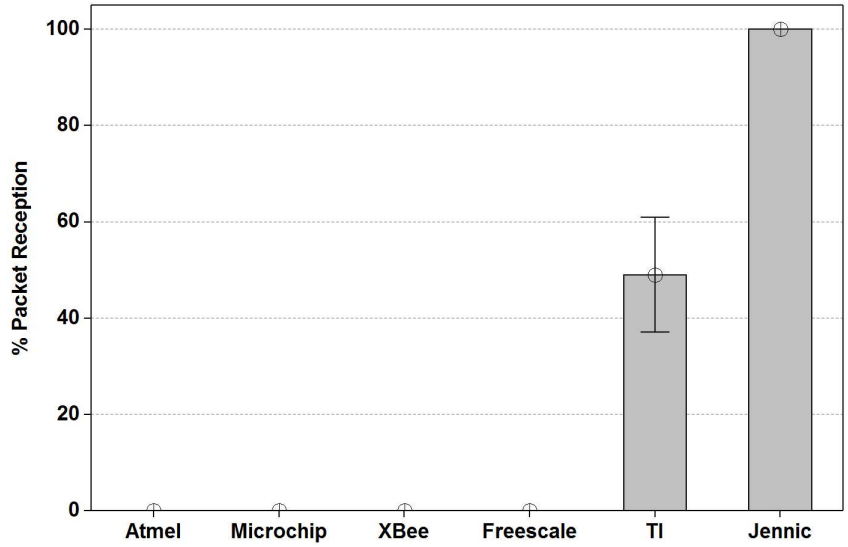

Figure 6. Packet reception rate per device type (2/8 preamble).

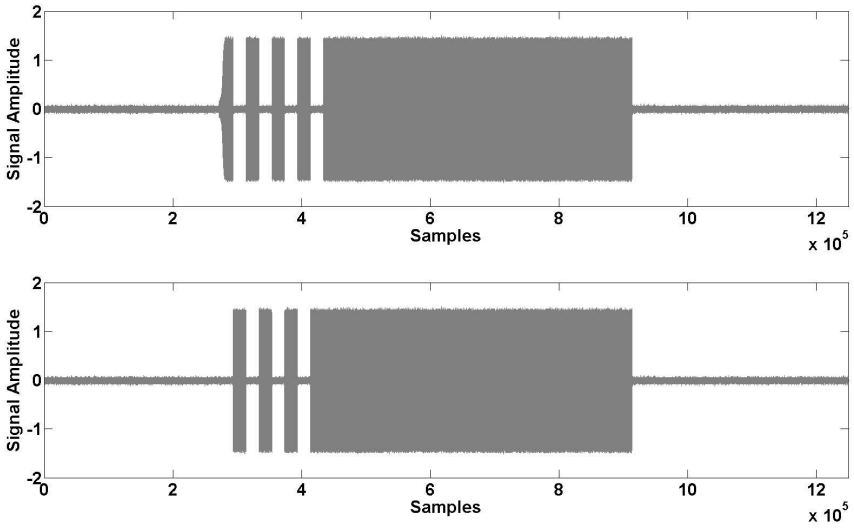

Figure 7. Beacon requests – odd (top) and even (bottom) preamble symbols.

Hundreds of O-QPSK symbol-wise modifications to the standard preamble are possible, providing a sizable search space for future research. The two modifications investigated in this paper are perforated preambles with only odd numbered symbols and only even numbered symbols present (Figure 7). Packet reception rates for the two perforated preamble modifications are shown in Figures 8 and 9. When only odd-numbered symbols were present (symbols

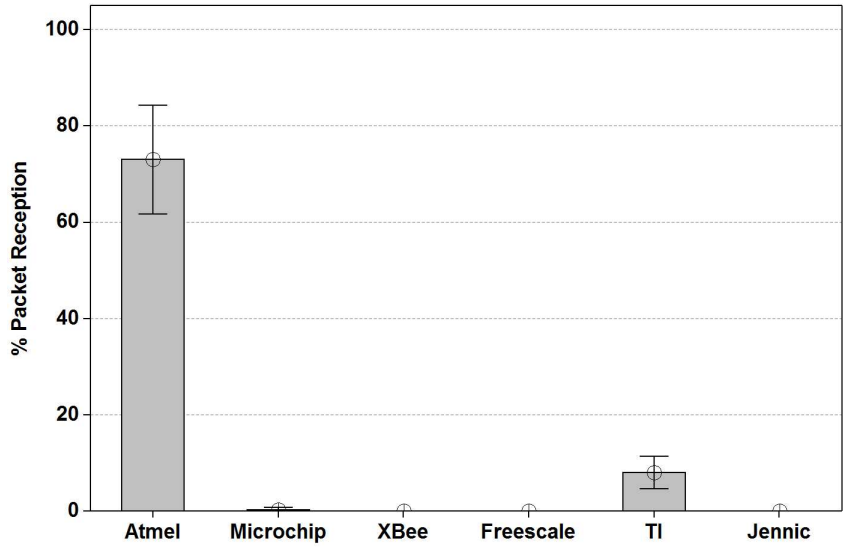

Figure 8. Packet reception rate per device type (odd preamble symbols).

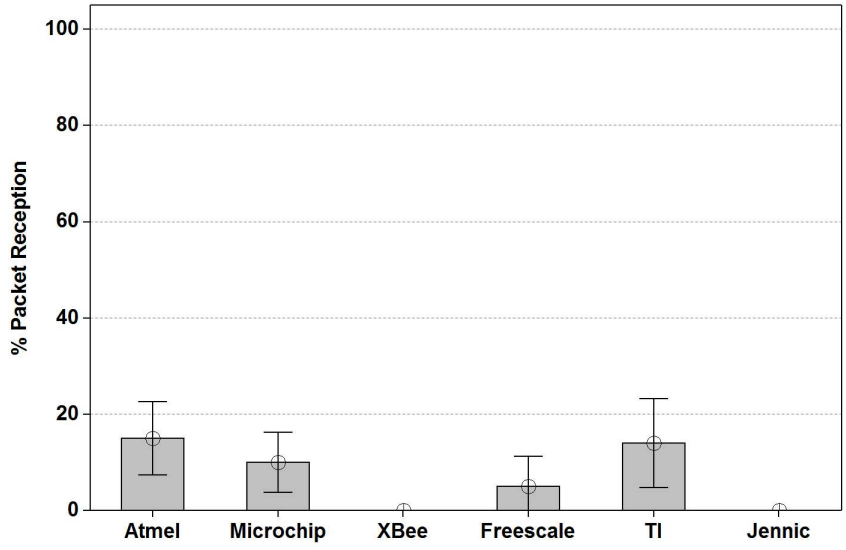

Figure 9. Packet reception rate per device type (even preamble symbols).

1, 3, 5 and 7), the Atmel AT86RF230 interpreted approximately 75% of the packets, the TI CC2420 interpreted less than 10% and all the other devices were effectively thwarted. In the case of the second perforated preamble where only even-numbered symbols were present, all six device types found it difficult to interpret packets – the mean packet reception rates fell below 20% for all

the transceivers. The two perforated preamble modifications were similar in composition, but they resulted in significantly different packet reception rates. The transceiver-specific responses to non-standard preambles reveal underlying differences in hardware implementations.

A promising research avenue stemming from these results is the potential for the remote identification of the make and model of ZigBee transceivers. Remote classification of transceiver models is essential for comprehensive cyber situational awareness, including identifying rogue devices and locating devices that are vulnerable to specific key extraction techniques.

5. Discussion

The packet reception rates suggest that interrogation packets with appropriately modified preambles can uniquely identify transceiver models based on the packet reply rates. This fingerprinting technique somewhat mirrors the network and transport layer based operating system fingerprinting techniques pioneered by Nmap. For example, based on the six device types investigated in this paper, an unknown ZigBee device that responds to packets with 4/8 preamble but not to packets with 2/8 preamble is most likely to be a Freescale MC13213 transceiver. Our future research will leverage these preliminary results to develop a remote ZigBee transceiver identification methodology to enhance cyber situational awareness.

An alternative strategy for identifying rogue devices and active eavesdroppers is to broadcast acknowledgement requests with preambles that are invisible to trusted hardware. An observed reply could only have originated from a device that is not a part of the trusted network. Such a device could then be tracked down and investigated.

6. Conclusions

ZigBee networks are widely used in the critical infrastructure, but an increasing number of tools are being developed to exploit and disrupt these networks. Under these circumstances, a rekeying technique that can effectively respond to suspected malicious ZigBee devices has considerable value. The rekeying technique proposed in this paper employs novel backward-compatible ZigBee message obfuscation based on preamble modifications.

Experiments involving the rekeying technique reveal that common wireless ZigBee sniffers can be thwarted with 100% effectiveness without reducing packet reception by transceiver models. In particular, specially-modified ZigBee messages sent to Freescale MC13213, TI CC2420 and Jennic JN5148 transceivers can be protected from interception by two popular sniffers and the XBee XBP24CZ7PIS, simply by shortening the IEEE 802.15.4 preamble by 50%. Packets with preambles reduced to 25% of the IEEE 802.15.4 specification completely thwart the Freescale MC13213. Alternatively, covert communications to Atmel AT86RF230 transceivers is possible using perforated preambles where only the odd-numbered O-QPSK symbols are present. When combined

with effective rekeying strategies, message obfuscation can effectively complement network defense. An important advantage is that the message obfuscation is backward compatible with tens of millions of existing ZigBee devices and can be leveraged to defend key exchanges or any other sensitive message traffic in wireless networks.

Note that the views expressed in this paper are those of the authors and do not reflect the official policy or position of the U.S. Air Force, U.S. Department of Defense or the U.S. Government.

Acknowledgements

The authors wish to thank John M. Greenwell for his invaluable support with the test equipment.

References

[1] C. Antosh, B. Mullins, R. Baldwin and R. Raines, A comparison of keying methods in the Hubenko Architecture as applied to wireless sensor networks, *International Journal of Autonomous and Adaptive Communication Systems*, vol. 3(3), pp. 350–368, 2010.

[2] M. Blaser, Industrial-strength security for ZigBee: The case for public-key cryptography, *Embedded Computing Design*, May 13, 2005.

[3] S. Camtepe and B. Yener, Key Distribution Mechanisms for Wireless Sensor Networks: A Survey, Technical Report TR-05-07, Department of Computer Science, Rensselaer Polytechnic Institute, Troy, New York, 2005.

[4] Daintree Networks, Understanding ZigBee Commissioning, Mountain View, California, 2007.

[5] Daintree Networks, Building and Operating Robust and Reliable ZigBee Networks, Mountain View, California, 2008.

[6] B. Danev and S. Capkun, Transient-based identification of wireless sensor nodes, *Proceedings of the International Conference on Information Processing in Sensor Networks*, pp. 25–36, 2009.

[7] Digi International, XBee/XBee-PRO ZB SMT Modules, Minnetonka, Minnesota (`ftp1.digi.com/support/documentation/90002002_C.pdf`), 2012.

[8] C. Dubendorfer, B. Ramsey and M. Temple, An RF-DNA verification process for ZigBee networks, *Proceedings of the Military Communications Conference*, 2012.

[9] T. Goodspeed, Extracting keys from second generation ZigBee chips, Presented at the *Black Hat USA Conference*, 2009.

[10] J. Heo and C. Hong, Efficient and authenticated key agreement mechanism in low-rate WPAN environment, *Proceedings of the First International Symposium on Wireless Pervasive Computing*, 2006.

[11] Q. Huang, J. Cukier, H. Kobayashi, B. Liu and J. Zhang, Fast authenticated key establishment protocols for self-organizing sensor networks, *Proceedings of the Second ACM International Conference on Wireless Sensor Networks and Applications*, pp. 141–150, 2003.

[12] Q. Huang, H. Kobayashi and B. Liu, Energy/security scalable mobile cryptosystem, *Proceedings of the Fourteenth IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, vol. 3, pp. 2755–2759, 2003.

[13] Institute of Electrical and Electronics Engineers, Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs), Part 15.4, IEEE Standard 802.15.4a-2007, New York, 2007.

[14] P. Jokar, H. Nicanfar and V. Leung, Specification-based intrusion detection for home area networks in smart grids, *Proceedings of the IEEE International Conference on Smart Grid Communications*, pp. 208–213, 2011.

[15] C. Kiraly and G. Picco, Where's the mote? Ask the MoteHunter! *Proceedings of the Thirty-Seventh IEEE Conference on Local Computer Networks (Workshop on Practical Issues in Building Sensor Network Applications)*, pp. 982–990, 2012.

[16] K. Masica, Recommended Practices Guide for Securing ZigBee Wireless Networks in Process Control System Environments, Draft, U.S. CERT, Department of Homeland Security, Washington, DC, 2007.

[17] S. Nguyen and C. Rong, ZigBee security using identity-based cryptography, *Proceedings of the Fourth International Conference on Autonomic and Trusted Computing*, pp. 3–12, 2007.

[18] B. Ramsey, B. Mullins and E. White, Improved tools for indoor ZigBee warwalking, *Proceedings of the Thirty-Seventh IEEE Conference on Local Computer Networks (Workshop on Practical Issues in Building Sensor Network Applications)*, pp. 925–928, 2012.

[19] B. Ramsey, M. Temple and B. Mullins, PHY foundation for multi-factor ZigBee node authentication, *Proceedings of the IEEE Global Telecommunications Conference*, pp. 813–818, 2012.

[20] E. Shi and A. Perrig, Designing secure sensor networks, *IEEE Wireless Communications*, vol. 11(6), pp. 38–43, 2004.

[21] R. Speers and R. Melgares, Api-do: Tools for ZigBee and 802.15.4 Security Auditing, Department of Computer Science, Dartmouth College, Hanover, New Hampshire (`code.google.com/p/zigbee-security`), 2012.

[22] Y. Sun and K. Liu, Hierarchical group access control for secure multicast communications, *IEEE/ACM Transactions on Networking*, vol. 15(6), pp. 1514–1526, 2007.

[23] T. Whittaker, Final word, *Control and Automation*, vol. 18(3), p. 48, 2007.

[24] J. Wright, KillerBee: Framework and Tools for Exploiting Zigbee and IEEE 802.15.4 Networks, Version 1.0, InGuardians, Washington, DC (`code.google.com/p/killerbee`), 2011.

[25] E. Yuksel, H. Nielson and F. Nielson, ZigBee-2007 security essentials, *Proceedings of the Thirteenth Nordic Workshop on Secure IT Systems*, pp. 65–82, 2008.

[26] ZigBee Alliance, ZigBee Smart Energy Profile Specification, ZigBee Document 075356r15, San Ramon, California, 2008.

[27] ZigBee Alliance, ZigBee Health Care Profile Specification, ZigBee Document 075360r15, San Ramon, California, 2010.