# Exploring Security in ZigBee Networks

Lindsey N. Whitehurst
University of South Alabama
School of Computing
Mobile, AL 36688
251-604-5419
lnw1222@jagmail.southalabama.edu

Todd R. Andel
University of South Alabama
School of Computing
Mobile, AL 36688
251-460-6701
tandel@southalabama.edu

J. Todd McDonald
University of South Alabama
School of Computing
Mobile, AL 36688
251-460-7555
jtmcdonald@southalabama.edu

## ABSTRACT
ZigBee networks have become popular for their low cost, low power, and ease of implementation. The ZigBee protocol has particularly become prevalent for home automation and controlling devices such as door locks and garage door openers. Preventing attacks and reducing vulnerabilities is imperative in cases where there can be high financial losses due to poor security implementations. For systems where low power and cost are desirable, but security is a priority, the application developer must be extremely cautious in the design of their network. This paper surveys security issues and vulnerabilities in the ZigBee specification and current key management schemes proposed for these networks.

## Categories and Subject Descriptors
C.2.0 [**Computer-Communication Networks**]: General – *Security and Protection.*

## General Terms
Security, Design

## Keywords
Wireless Networks, Security, 802.15.4, ZigBee, Smart Grid

## 1. INTRODUCTION
The ZigBee wireless specification is being used in numerous, new applications. It provides a simple, reliable solution for wireless sensor networks. Its low power consumption and low cost make it suitable for lighting control, smart utility meters, door locks, medical systems, manufacturing systems, and energy usage monitors. However, as these networks are being used for more critical systems, security becomes a major issue. For example, if ZigBee devices are used in a home area network for the smart grid, shutting off power and modulating appliances can become vulnerabilities [5]. Systems where ZigBee devices are controlling the nation's critical infrastructure, like the smart grid, require much higher standards of security than general wireless area networks. Other ZigBee networks control water spill gates, natural gas valves, door locks, and HVAC systems. As these devices start to control aspects of the physical world, more risks are introduced, and security becomes a priority [10].


*CISR '14, April 08-10 2014, Oak Ridge, TN, USA*
*ACM 978-1-4503-2812-8/14/4.*
*http://dx.doi.org/10.1145/2602087.2602090*

## 2. BACKGROUND
ZigBee is a wireless device-to-device communication standard for low power applications, such as lighting control, fire or smoke detectors, and energy usage monitors. ZigBee networks are characterized by low power usage and a low data rate. They operate in the 2.4 GHz range, and have a maximum data rate of 250 kbps [7]. Networks can be as small as two devices to as large as thousands of devices covering a hotel. They provide end-to-end communications, and use 802.15.4 for the medium access control (MAC) layer and the physical (PHY) layer. On top of these layers, ZigBee defines the network (NWK) layer and provides an application layer (APS) framework.

A ZigBee network is made up of three different types of nodes: coordinator, router, and end device. The coordinator is responsible for forming the network, selecting a personal area network ID (PAN ID), and acting as a router, network manager, and trust center. There can only be one coordinator per ZigBee network. End devices are leaf nodes within the network. They only communicate to their parent, and may be sleepy or non-sleepy devices. Sleepy end devices turn off their radio when they are idle, and wake up roughly every couple of seconds to poll their parent to see if they have any messages or acknowledgements. Non-sleepy end devices are always powered, but do not relay messages for other devices. Routers within the network may act as end devices, provide routing services, and cannot be sleepy. The network topologies that ZigBee supports include tree and mesh networks.

### 2.1 ZigBee Security Background
The ZigBee specification defines security features, such as key establishment, key transport, frame protection, and device authorization. It assumes trust between layers within the ZigBee stack, so cryptographically securing transmission is only done between devices. All services within the network use the same security level once it is decided upon. The specification does not define out-of-band methods for key setup, policy for accepting new devices to the network, policy for expiration and update of keys, or the handling of security errors, loss of counter synchronization, and loss of key synchronization [11]. The coordinator acts as the trust center for the network. As the trust center, it is responsible for managing the security settings and authorizations on the network. It stores the keys for the network, configures other devices with keys, and authorizes devices. All other nodes on the network have full trust with the coordinator to obtain keys and gain access.

There are three different types of keys that can be used: master, link, and network. The master key is used for symmetric-key key establishment (SKKE) to establish link keys. It is either preconfigured or sent in the clear from the trust center. Link keys are shared between two devices on the network, usually between a device and the trust center. It is established dynamically. The network key is shared by all the devices that are on the network. The trust center keeps a set of network keys and the current one is identified by a key sequence number. Network keys are updated by encrypting the new network key with the old network key and broadcasting it to all the network devices. These symmetric keys can be distributed through the network in three ways, pre-installation, transport, or key establishment. The types of keys used on the network depend on the network topology as well as the security level desired.

## 2.2 802.15.4 Security Background

The two lower layers, the MAC and PHY layers, are defined by the 802.15.4 protocol. The 802.15.4 protocol has three optional security features: frame encryption, integrity protection, and access control. These three features can be combined and implemented in three different security modes: unsecured mode, access control list mode, and secured mode. In unsecured mode, no security services are utilized. In access control list mode, each node on the network maintains a list of devices and how to communicate with them. In secured mode, different AES-based security suites are used. The different suites include no security, encryption only, authentication only, and authentication and encryption. AES-CTR mode is used for encryption, and AES-CBC-MAC is used for authentication. Although 802.15.4 supports 32 bit, 64 bit, and 128 bit key sizes, we only consider 128 bit key sizes that because that is the size required by the ZigBee specification and by AES.

AES-CTR mode provides confidentiality through encryption in counter mode. The nonce values that are input to the block cipher are composed of: flags (1 byte); the source address (8 bytes); a frame counter (4 bytes); a key counter (1 byte); and a block counter (2 bytes). The frame counter identifies the packet. It is usually maintained by the hardware and it is incremented for each packet sent by a node. When the frame counter reaches its maximum value an error is returned to the upper layers. The key counter is controlled by the application layer. It can be incremented if the frame counter ever reaches its maximum to increase the time before declaring an error. A requirement for secure communications using AES-CTR mode is that nonce values are never repeated within the lifetime of a single key. The frame and key counters are used in order to prevent nonce reutilization so that communication remains secure.

AES-CBC-MAC is used for integrity protection. The MAC function is applied over the entire payload and packet headers. Therefore, not only can the integrity of the payload be checked, but the integrity of the headers may also be verified. In AES-CCM, encryption and authentication is utilized. First the AES-CBC-MAC is applied, and then AES-CTR mode is used to encrypt the payload and the MAC value.

In access control list (ACL) mode, each node maintains a list of devices in an ACL table, which is used to establish node-by-node security. The 802.15.4 protocol supports up to 255 ACL entries. An ACL entry consists of: address of the destination node; security suite to use with that node; the key to use with that node; the last value of the initialization vector used with the destination node; and a replay counter. The MAC layer looks up the address of the destination node, finds the corresponding security suite, key, and nonce to use for communications with that node. This protocol also supports a default entry in the ACL table. The default entry is used when an address for the other node is not found within the table. If there is no default entry, and the address cannot be found within the table, an error is reported back to the application if the security flag for ACL mode is set.

In addition to the higher layer ZigBee keys, 802.15.4 supports the following keys: network shared key; pairwise keys; group keys; and hybrid approaches. ZigBee only supports network keys, master keys, and link keys; therefore, we only consider the network shared key and pairwise keys at the MAC layer because all keys must be shared throughout the protocol stack. The network key is shared among all nodes and is compromised if a single node is compromised. A pairwise key is shared between two specific nodes. Each pair of nodes shares a different key. If a node is compromised, only messages to and from that node are compromised. However, the use of pairwise keys requires much more overhead than simply using a network key; each node has (n-1) keys versus a single network key.

# 3. SECURITY CONSIDERATIONS

The strength of a ZigBee device's security is reliant on the security at the different layers in the ZigBee protocol stack. The two lowest layers, PHY and MAC layers, are defined using the 802.15.4 protocol. The next two layers, the network and application layers, also have additional, optional security capabilities. However, even if the security suite is resistant to successful attacks, its strength is limited by the confidentiality of the key used. Therefore, it is also necessary to have a secure key management scheme. First, we reflect on the security features in the 802.15.4 protocol, and then we analyze the security at the two upper layers. Then, we cover current and proposed key management schemes for a ZigBee network. Finally, current, open source ZigBee exploitation frameworks are discussed.

## 3.1 Vulnerabilities in 802.15.4 and ZigBee

There are a number of vulnerabilities in 802.15.4 security that must be considered when setting up a ZigBee network. First, there is an issue with reusing nonce values with the same key. This introduces vulnerabilities because there is a way for an attacker to recover two plaintexts using their ciphertexts in AES-CTR mode. Because of the properties of the XOR (exclusive OR) function, if the attacker takes the XOR of two ciphertexts that have been encrypted with the same key and nonce values, they can recover the two plaintexts. This is similar to the Borisov, Goldberg, Wagner attack on WEP [1]. Their attack shows an important vulnerability in WEP security, and is one of the reasons WEP in 802.11 is no longer used for secure communications.

There are a number of ways that nonce reuse may occur. First, if multiple ACL entries have the same key, the nonce may be reused. Because the separate ACL entries will have their own, independent values for their initialization vector (IV), they may use the same value for the IV with the same key, causing confidentiality to be broken. One way of preventing this from happening is to design the application layer to deal with shared keys in such a way that IV values are bound to key values [6]. Another way IVs may be reused is when ACL entries are lost because of power loss. If the application layer repopulates the

table with the appropriate keys, it may populate the IV values with a value that has already been used with the keys. Again, confidentiality will be broken. In order to prevent this flaw, new keys should be established after power disruptions or key counters should be stored in flash memory so that they are not lost [6]. Key establishment becomes an issue for the first solution, and memory speed and efficiency become issues for the second solution. One problem especially pertinent to ZigBee networks is how to maintain ACL entries during sleepy mode. If power is maintained to the memory containing the ACL entries during sleep mode, power consumption is raised, causing an issue for ZigBee devices because they are designed for low power applications. An alternative is to have the upper layer software store the ACL entries for all the sleepy devices during their low power mode. However, there is over two kilobytes of overhead associated with using this solution [6]. Because ZigBee does not specify what to do for loss of synchronization, this issue is left up to the application designer, and must be dealt with carefully.

Another issue with 802.15.4 MAC security is with unauthenticated encryption. If replay protection is used, an adversary can send a message with the key counter and frame counter set to the maximum. Even with a payload that is encrypted with the wrong key, the MAC layer will set their counters to the max and any further packets received will be discarded because it will appear to be a replay [6]. This is a simple implementation of a denial-of-service (DoS) attack on the 802.15.4 MAC layer. In order to prevent this DoS attack, there must be an integrity check before updating counters from received packets. Finally, there are no integrity checks on acknowledgement packets. Acknowledgement packets are sent with a sequence number. Sequence numbers are easy to intercept since they are sent in the clear in the original packet [6]. Because there are no integrity checks on these packets, it would be easy for an attacker to forge an acknowledgement packet back to the sender. Therefore, acknowledgements at the MAC layer may not be trustworthy.

Because of the issues of IV reuse, unauthenticated encryption, and untrustworthy acknowledgements, we now look to upper layer security features for additional safety measures. The NWK and APS layers on the ZigBee stack support four levels of security, similar to 802.15.4. These levels are: no security; confidentiality; authentication; and both confidentiality and authentication. The security suites possible are also the same as the MAC layer, AES-CTR, AES-CBC-MAC, and AES-CCM. The strength of the security at these layers depends on keeping the keys secret. Because ZigBee uses symmetric key cryptography through the use of AES, each communicating node must have the same key. Therefore, secure key exchange is important during key updates and device association.

## 3.2 ZigBee Network Key Management

As stated in Section 2, there are three types of keys on a ZigBee network: master; link; and network. Encryption and authentication can be enabled at the APS, NWK, and MAC layers, and each layer shares the same keys. The optional, master key is a shared secret between two devices, and it is used to establish link keys between them. It can be installed during manufacturing, out-of-band, or sent from the trust center. The link key is either transported from the trust center, installed out-of-band, or setup using (SKKE) or public-key key establishment (PKKE). Network keys are either installed out-of-band or sent from the trust center. Network keys must continually be updated in order to prevent IV reuse, and it is commonly updated by sending it in the clear or encrypting it with the old network key. Even with the encryption option, encrypting the new network key with the old one does not preserve forward secrecy. Because this is extremely insecure, it is necessary to establish a robust key management scheme to enhance it. Furthermore, because network keys are broadcast, any new devices on the network must necessarily be authenticated before joining the network.

There is a new optional cluster in the smart energy profile called the key establishment cluster. This cluster can be used for link key establishment on the network. However, the network key is still simply broadcast in the clear, even in this profile. This cluster has two options for establishing link keys, SKKE and PKKE. SKKE uses the master key to derive the pair of symmetric keys. Therefore, the master key must be installed on each device or shared over the air prior to running this protocol. If the master key is public or known, then SKKE is known as "Unprotected Key Establishment." PKKE is implemented using certificate-based key establishment (CBKE), using elliptic curve cryptography [12]. Public key cryptography has traditionally been overlooked as a solution for sensor networks because of the computational overhead. However, preliminary work has shown they may be a more viable option because the message complexity, memory usage, and security resilience are more beneficial than symmetric-key schemes for some environments [9]. This specification does not have proper key revocation, and compromised nodes may still access the network.

There has been research on how to implement more efficient systems on ZigBee networks, such as broadcast group key management scheme (BGKM) [4]. In this protocol, each node contains specific secrets that it uses along with broadcasted public information to derive a symmetric key. BGKM minimizes the amount of data requirements to a few secret information items. Revoking a node involves removing its corresponding secrets from the coordinator's table. When a node is added, its information is added to the coordinator's table. However, this method requires that each node be successfully authenticated. Their solution to this problem is to add a challenge-response at the time of installation so that each node can be properly authenticated. This solution can cause issues with scalability as well as how to deal with a compromised node after it is authenticated. This solution also adds additional difficulty to replace and maintain nodes.

Choosing the proper key management scheme for different ZigBee networks may become application specific. However, there are a few "best practices" that should be followed with whichever key management scheme is chosen. This includes, when possible, preloading network keys on nodes that are joining the network for authentication purposes, enabling security at all three layers of the protocol stack, using the ACL mode at the MAC layer, and loading keys out-of-band [3]. Keys that are loaded out-of-band are the most secure because they are not transmitted over the air, and a manufacturer does not have a copy of the keys. Although a secure key management scheme is necessary, it does not prevent attackers that have physical access to the device. KillerBee, described in the next section, leverages physical access and wireless data captures to gain access to keys. Even if the KillerBee attack is not possible, other successful active and passive attacks have broken AES implementations [2].

Therefore, key management extends to the physical location and access to the device as well.

## 3.3 Exploiting ZigBee Security

There are a few open source tools available that can help assess the security on a ZigBee network. KillerBee, a framework developed by Joshua Wright, is able to sniff and manipulate ZigBee traffic [10]. Some available KillerBee tools include:

- zbdump: saves any sniffed traffic to a libpcap format
- zbreplay: replays traffic from a saved libpcap file
- zbdsniff: searches for keys that are sent in the clear
- zbgoodfind: searches a firmware dump for keys
- zbassocflood: floods ZigBee router or coordinator with association requests

With these tools, an attacker can be anyone with the hardware to sniff and send wireless traffic within the ZigBee network. Furthermore, because ZigBee specification allows network keys to be updated by sending them in the clear over the air, any device that saves network traffic during this update will have access to the new key. Any new data sent after an attacker obtains the key is no longer confidential. Even if the key is not updated over the air, it is still stored physically on the device. If an attacker has access to the device, they can use a firmware dump to retrieve contents from memory, which is used along with captured data to determine the key. Keys on many vendor devices, including Ember and TI, can be recovered from memory even when they are locked [10]. With access to the key, an attacker can decrypt any traffic using Wireshark. Another tool available is Api-do [13]. It expands upon KillerBee to add support for additional radios and jamming. It also added support for the MAC layer to Scapy software.

Another extremely effective attack can be performed utilizing zbassocflood to flood the coordinator with association requests. Because these types of packets are sent in the clear, no authentication is required, and the attack can be easily carried out. Additionally, ZigBee provides minimal replay protection, a 4-octet frame counter, so using zbreplay can be extremely effective attacking strategy. In [8] the authors provide a proof of concept for using an anomaly-based detection system in order to prevent association and replay flooding on a network. They claim that an anomaly-based intrusion detection system is more feasible than a pattern-based system because it uses fewer resources, which are scarce on ZigBee devices. They propose that the system learns normal behavior for a general case, rather than a specific environmental setting. After learning normal behavior, it monitors at specific time intervals to minimize resource utilization. The simulations that they have run are on an extremely small scale, and issues of scalability may arise for larger systems. Although they claim that network learning should be general, their simulation used network learning during normal operation. This cannot be guaranteed for all systems, where attacks may occur as soon as the network is running, causing the 'normal' state for the network to be skewed, and any intrusion detection to be inaccurate. Although this is a step towards improving the security of a ZigBee network, KillerBee is still an effective tool for attacking the security of a ZigBee network, with and without enabled security features.

# 4. CONCLUSIONS

ZigBee security is a complex problem involving issues at the MAC, NWK, and APS layers. Known attacks and vulnerabilities on 802.15.4 networks, such as IV reuse, ACL loss, and DoS attacks, are also pertinent to ZigBee networks. The application developer must take these types of attacks into account when implementing their security features, because it is not as simple as just choosing an encryption algorithm. If no features are enabled the default security level on a ZigBee network is no security. A network with security implementations at all layers of the stack is only as secure as its key. Therefore, an appropriate key management scheme for ZigBee networks is extremely importance. Key management applicability involves scalability, efficiency, and its ability to evolve. Although the ZigBee key establishment cluster provides SKKE and PKKE, they are not infallible. The ultimate strength of ZigBee network security is based on the secrecy of its keys and on secure application development.

# 5. ACKNOWLEDGEMENTS

This material is based in part upon work supported by the National Science Foundation under grant DUE-1241675.

# 6. REFERENCES

[1] Borisov, N., Goldberg, I., and Wagner, D. Intercepting Mobile Communications: The Insecurity of 802.11. *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, ACM (2001), 180–189.

[2] Clavier, C. Passive and Active Combined Attacks on AES. *Fault Diagnosis and Tolerance in Cryptography*, (2007).

[3] Masica, K. *Recommended Practices Guide for Securing ZigBee Wireless Networks in Process Control System Environments*. 2007.

[4] Nabeel, M., Zage, J., Kerr, S., et al. *Cryptographic Key Management for Smart Power Grids - Approaches and Issues*. 2012.

[5] Saponara, S. and Bacchillone, T. Network Architecture, Security Issues, and Hardware Implementation of a Home Area Network for Smart Grid. *Journal of Computer Networks and Communications 2012*, (2012).

[6] Sastry, N. and Wagner, D. Security Considerations for IEEE 802.15.4 Networks. *Web Information Systems Engineering*, (2004), 1–11.

[7] Silicon Labs. Ember Application Development Fundamentals: Overview. (2013), 1–61.

[8] Stelte, B. and Rodosek, G.D. Thwarting Attacks on ZigBee - Removal of the KillerBee Stringer. (2013), 219–226.

[9] Wang, H., Sheng, B., Tan, C.C., and Li, Q. Comparing Symmetric-key and Public-key Based Security Schemes in Sensor Networks: A Case Study of User Access Control. *The 28th International Conference on Distributed Computing Systems, 2008. ICDCS '08*, (2008), 11–18.

[10] Wright, J. Killerbee - Framework and tools for exploiting ZigBee and IEEE 802.15.4 networks. 2011. http://www.willhackforsushi.com/presentations/toorcon11-wright.pdf.

[11] ZigBee Alliance. ZigBee Security Specification Overview. 2005.

[12] ZigBee Alliance. ZigBee Smart Energy Profile Specification. (2008), 1–217.

[13] *Api-do: Tools for ZigBee and 802.15.4 Security Auditing from the Dartmouth Trust Lab*. 2011.