Survey of IoT Security Implementations

Ryan Morganti

**Abstract**

This project is a survey over the various Internet of Things (IoT) protocols that are used within the larger web environment. Four different protocols will be analyzed and compared for their security implementations and privacy practices. These protocols are: Bluetooth Low Energy (BLE), ZigBee, Z-Wave and LoRaWAN. The project will utilize Confidentiality, Integrity, and Authentication as a basis to analyze and compare the protocols. The project will address the similarities and differences for each of these categories and try to determine which, if any, of the protocols provides the most security in each of the said categories. Implementations of each protocol may differ in practice, so only the official specification and ideal use will be considered.

**Introduction**

Although IoT has had more interest and research conducted within it over the last few years, it is still an emerging field with many unknown variables (Mardiana). As the number of IoT devices connected to the internet is projected to increase to around 20 billion by the year 2025, the security and protection of these devices is essential (IoT-Analytics, Statistica). Things that historically have been controlled locally by human interaction can now be monitored and altered from a user miles away. Commonly, home items like coffee makers, ovens, washers, televisions, etc. are mentioned in the making of a smart home, but the application of IoT extends even further. Critical infrastructure systems like nuclear rod thermometers, power grids, temperature regulation, drones, and more are also being connected as IoT devices to the internet. Although IoT devices are making life easier by putting control on the go, the security implications are a necessity. Exposing anything to the internet allows malicious actors a chance to attack and potentially cause devastating impacts on critical infrastructure. This is why the study into security practices is essential for IoT protocols, as securing critical systems is of utmost importance. Unfortunately, many IoT devices still lack any built-in security besides what the protocol offers (Hampson). Although the protocols have added more security features, they only have so much that they can offer to insecure IoT devices.

**Bluetooth Low Energy v5 (BLE)**

BLE was first introduced as a branch of Bluetooth Classic v4 in 2010. Like Bluetooth, BLE focuses on short-range communication and connecting IoT devices together. However, BLE is incompatible with Bluetooth Classic, as it uses an entirely different chip set and method to communicate. Where Bluetooth Classic uses high speed connections and has high power consumption, BLE utilizes short send and receive windows to save energy and allow IoT devices with small batteries to communicate. BLE is a peer-to-peer connection protocol, meaning it directly connects two devices together and does not need a centralized server to manage a network.

In BLE, connecting devices is a three step procedure. To start, BLE devices send out unsecured advertisements demonstrating the capabilities of the device. The second step is done in either Legacy or LE Secure connection mode. The legacy protocol is used for all BLE 4.0 devices and earlier, while LE Secure can be used in BLE 4.2 or above. In legacy, devices generate a temporary key through either just works, out of band, or passkey entry methods. Just works generates a key of just 0s, out of band shares a key over a different band such as NFC, and passkey entry relies on the user to enter a pin on both devices. Once this temporary key is exchanged, a Short Term Key (STK) is calculated and used to encrypt further traffic. In LE Secure, instead of a temporary key, a Long Term Key (LTK) is generated using Elliptic Curve Diffie Helman (ECDH) which is used to encrypt all further traffic. Step three is used to transport the final derived keys between the devices. This message is encrypted with either the LTK or STK with the data values of the Long Term Key, Connection Signature Resolving Key, and the Identity Resolving Key. These keys are used to generate private Bluetooth addresses between the devices and to encrypt traffic.

Within these connections, BLE supports different security levels and security modes. The levels are: 1 – no security, 2 – AES-CMAC encryption when devices are unpaired, 3 – AES-CMAC encryption that requires pairing, and 4 – ECDH encryption when devices are paired. The different security modes are: 1 – without signing data, 2 – signing of data, 3 – a mixed mode supporting both signed and unsigned data. What follows is a look into how BLE implements the three aspects of CIA:

Confidentiality – Traffic is encrypted with the LTK using AES-CMAC or ECDH, preventing outsiders from reading the transmitted data.

Integrity – Packets are signed with a MIC which is used to check any changes during transmission.

Authorization – Devices are verified during the pairing procedure either by user data entry or by verifying using shared keys between devices.

**ZigBee**

ZigBee is a low-energy IoT protocol developed by the Zigbee Alliance designed for use in smart home networks. To use ZigBee networks, connecting devices must be compliant with the ZigBee protocol, and a central ZigBee hub must exist in range of the network. The ZigBee hub initializes the network and acts as the intermediary between the ZigBee network and the IP internet. ZigBee can function in three different topologies, either a star, tree, or mesh network. With mesh networks ZigBee allows devices to communicate in a peer-to-peer fashion which helps eliminate network faults as other routers can pick up the traffic and continue the communication. As long as there is a path between devices, a device may communicate with the central hub even if it is not directly in range of the hub. With this, a theoretical maximum of 65,000 devices can be connected to each ZigBee network.

ZigBee can act in two separate physical modes, either centralized or distributed. In a centralized structure, a device called a trust center, or network coordinator, is used to distribute the network key and provide each end device with a unique link key to communicate with the coordinator. In the distributed set up, the ZigBee routers enroll other routers and end devices by sharing the network key with them using a preconfigured link key generated before the device joins the network. Within these two modes, ZigBee can also operate at different security levels. In a standard smart home environment ZigBee will usually operate in standard mode, meaning the network key is distributed in the open to new devices. While in high-security mode, the network key will be distributed using Symmetric-Key Key Establishment to new devices.

To achieve this security, ZigBee utilizes two separate types of 128-bit AES keys. The first is the Network Key which is used to secure broadcasts from devices and allow them to securely communicate with each other in the network layer. It is either installed via key-transport or pre-installation to each device. The second is the Link Key which encrypts unicast traffic between two devices. ZigBee defines various values for this including default global trust center, distributed security global, install code, application link, and device specific trust link keys. All of these produce the same effect of encrypting messages on a peer-to-peer basis; they just differ one which devices are in communication with each other. What follows is a look into how ZigBee implements the three aspects of CIA:

Confidentiality – All traffic is encrypted using the network key for broadcast messages, and link key for unicast messages. The keys can be transmitted securely if the ZigBee network is operating in High-Security, otherwise it may be possible to discover the network key in plain text.

Integrity – All traffic is verified with a MIC appended to the message. This is calculated with the AES operating in CCM mode.

Authentication – If ZigBee is operating as a centralized structure, it requires a manager to officiate the devices that can join the network and all of their keys. In the distributed mode, the routers act as an authenticator for new devices.

**Z-Wave**

Z-Wave is a proprietary protocol owned by the Z-Wave Alliance. Like Zigbee it is designed to provide a mesh network topology to IoT devices. This means that end devices can act as intermediate hosts and forward the communication further in the network. Z-Wave allows for a maximum of 4 hops between devices before a message is dropped, as well as a maximum of 232 unique devices per network. Each Z-Wave network contains a central hub or controller that connects every end device and manages the network information. Each Z-Wave network has a 32-bit HomeID field which is used to identify the network, and each node is assigned a NodeID value to uniquely identify it. The Z-Wave physical protocol operates in the sub 1Ghz network band, meaning it does not interfere with other wireless communications that exist on the 2.4 or 5Ghz bands. This low frequency also enables Z-Wave devices to have extended battery life as the transmissions require less energy to produce.

Z-Wave has two different specifications for security levels, these being S0 and S2. S0 is the outdated, insecure authentication protocol, while S2 is the new mandatory security protocol introduced in 2017 and required by all 700 series devices and later. In S2 devices first generate a temporary key to communicate with the central hub by using ECDH protocol. This creates a shared secret between the end device and controller which is then input to an algorithm which generates a temporary 128-bit AES key. This temporary key is used to encrypt further key exchange packets using AES-CCM. From these further exchanges a permanent CCM key is generated which will be used for all further communication encryption.

The S2 security specification also allows for Authenticated and Non-Authenticated devices to join the network. Authenticated devices, such as smart door locks, are mutually authenticated between the device and controller. This is an additional add-on to protect critical devices. The controller will generate different keys between the authenticated and non-authenticated networks to separate them. In either group, all communication with the internet is done through a TLS-1.1 encrypted tunnel, protecting all IP traffic past the Z-Wave network. What follows is a look into how Z-Wave implements the three aspects of CIA:

Confidentiality – All traffic on the Z-Wave network is encrypted with AES-CCM, which protects data from being read. Further, all traffic on the IP network is run through a TLS-1.1 encryption tunnel so that no outsider can read the data.

Integrity – Traffic is protected with AES-CMAC to calculate a MIC and verify the message integrity.

Authentication – All packets are sent with AES-CCM encryption on them, which provides authentication as part of the algorithm.

**LoRaWAN1.1**

The LoRaWAN architecture is a general-purpose protocol for connecting sensors, equipment, and other IoT devices to the internet. It has been especially used in agricultural work as the underlying LoRa physical protocol allows for long range communication and low power consumption. The LoRa physical protocol allows devices to communicate in chirps and connect using the ALOHA approach, meaning the device can uplink at any time and maintain downlink windows for a specified time after said uplink. With this physical protocol, LoRaWAN devices can last years powered by a single button battery. LoRaWAN has been seeing an expanded reach into other daily appliances as well, going from use in just sensors into other everyday appliances as well.

The basic topology of LoRaWAN devices is a star-to-star network. This means that devices using the LoRa protocol send packets to all gateways in their area as a broadcast message. All gateways that receive the message will then forward the packet over UDP to a central Network Server (NS). This server takes incoming messages and proceeds to run a deduplication process on them as multiple gateways could have forwarded the same packet to the server. Depending on the implementation, the NS may then forward the uplink packet to a corresponding Application Server (AS) which processes the data from the device.

There are two methods of joining the network, Over the Air Activation (OTAA) or Authentication by Personalization (ABP). In ABP mode all session keys are pre-generated and entered by the administrator on the NS. In OTAA, keys are generated at activation time. The process starts when the device sends a Join-Request packet to the server. This packet consists of the JoinEUI, DevEUI, DevNonce and MIC. The DevNonce is a random generated value that is used to prevent replay attacks as the server tracks previously used values and drops packets that reuse the value. The MIC is calculated by using AES in CMAC mode with the NwkKey. The server verifies this information and will respond with a Join-Accept message containing elements of the JoinNonce, NetID, DevAddr, DLSettings, RxDelay, CFList and MIC encrypted with the same NwkKey using AES Decrypt in ECB mode. The device can then encrypt the packet with the NwkKey and recover the information. After this process is completed, the device calculates the various session keys that are used in further communication with the network. What follows is a look into how LoRaWAN implements the three aspects of CIA:

Confidentiality – LoRaWAN keeps all sensor data transmission with the network encrypted. If the data is sent to the NS, the NwkSEncKey is used, and if the data is sent to the AS, then the AppSKey is used. Depending on the implementation of LoRaWAN, the NS may not have access to the AppSKey, keeping all data going through it confidential.

Integrity – LoRaWAN calculates a MIC using AES CMAC on every message and uses the stored keys to verify that the message has not been changed.

Authentication – LoRaWAN1.1 allows for an additional Join Server which is used to authenticate a device to the NS if requested. Otherwise, all generated keys and the DevEUI are used to verify a joining device in the NS.

**Conclusion**

IoT security is a growing concern as more potentially insecure devices are being connected to the internet and exposing networks to risk. However, the protocols aforementioned in this paper are taking steps to further secure the network and device traffic in IoT devices. All of these protocols are using AES 128-bit keys or ECDH 128-bit keys, which are considered secure in modern infrastructure. All provide for Confidentiality, Integrity, and Authentication. If used properly, these protocols are secure; however, there are some potential flaws as security modes are not always enforced, key transport protocols may be intercepted, and default values may be used. Also, there may be potential issues in the specific software or hardware implementations which could be exploited. In the most theoretical sense, these protocols provide a good step in securing IoT devices.

Works Cited

Introduction

D. Dragomir, L. Gheorghe, S. Costea and A. Radovici, "A Survey on Secure Communication Protocols for IoT Systems," 2016 International Workshop on Secure Internet of Things (SIoT), Heraklion, Greece, 2016, pp. 47-62, doi: 10.1109/SIoT.2016.012.

Mardiana binti Mohamad Noor, The results of IoT failures can be severe, Rathore, M. M., … Challa, S. (2018, December 1). *Current research on internet of things (IOT) security: A survey*. Computer Networks. Retrieved April 25, 2023, from <https://www.sciencedirect.com/science/article/abs/pii/S1389128618307035>

Hampson, M. (2022, August 18). *IOT security risks: Drones, vibrators, and kids' toys are still vulnerable to hacking*. IEEE Spectrum. Retrieved April 25, 2023, from <https://spectrum.ieee.org/iot-security-risks-drones-vibrators-iot-devices-kids-toys-vulnerable-to-hacking>

*State of IOT 2022: Number of connected IOT devices growing 18% to 14.4 billion globally*. IoT Analytics. (2022, June 14). Retrieved April 25, 2023, from <https://iot-analytics.com/number-connected-iot-devices/>

Vailshery, L. S. (2022, November 22). *IOT connected devices worldwide 2019-2030*. Statista. Retrieved April 25, 2023, from <https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/>

Bluetooth Low Energy

Afaneh, M. (2023, March 30). *Bluetooth Low Energy (BLE): A complete guide*. Novel Bits. Retrieved April 25, 2023, from <https://novelbits.io/bluetooth-low-energy-ble-complete-guide/>

Antoniou, P. (2021, April 28). *Bluetooth Low Energy (BLE) security in a nutshell!* Medium. Retrieved April 25, 2023, from <https://antonioupan6.medium.com/bluetooth-low-energy-ble-security-in-a-nutshell-5d20164cceb2>

*The Bluetooth LE security study guide*. Bluetooth® Technology Website. (2023, March 17). Retrieved April 25, 2023, from <https://www.bluetooth.com/bluetooth-resources/le-security-study-guide/>

Kynetics. (n.d.). *Ble pairing and bonding¶*. Kynetics. Retrieved April 25, 2023, from <https://technotes.kynetics.com/2018/BLE_Pairing_and_bonding/>

Loveless, M. (2018, January 9). *Understanding bluetooth security*. Decipher. Retrieved April 25, 2023, from <https://duo.com/decipher/understanding-bluetooth-security>

Matthias Cäsar, & 2009, A. S. its introduction in. (2022, January 6). *A survey on Bluetooth Low Energy Security and Privacy*. Computer Networks. Retrieved April 25, 2023, from <https://www.sciencedirect.com/science/article/pii/S1389128621005697>

Ren, K. (2022, March 29). *Bluetooth pairing part 5:legacy pairing - out of Band*. Bluetooth® Technology Website. Retrieved April 25, 2023, from <https://www.bluetooth.com/blog/bluetooth-pairing-part-5-legacy-pairing-out-of-band/>

ZigBee

Bhis. (2022, November 23). *Understanding zigbee and wireless mesh networking*. Black Hills Information Security. Retrieved April 25, 2023, from <https://www.blackhillsinfosec.com/understanding-zigbee-and-wireless-mesh-networking/>

Dattatray. (2023, April 22). *Zigbee security 101 (architecture and security issues)*. Payatu. Retrieved April 25, 2023, from <https://payatu.com/blog/zigbee-security-101/>

*Discover zigbee protocol 3.0*. Discover Zigbee Protocol 3.0 | Digi International. (n.d.). Retrieved April 25, 2023, from <https://www.digi.com/solutions/by-technology/zigbee-wireless-standard>

GeeksforGeeks. (2023, February 22). *Introduction of zigbee*. GeeksforGeeks. Retrieved April 25, 2023, from <https://www.geeksforgeeks.org/introduction-of-zigbee/>

Rudresh, V. (2018, January 23). *Zigbee security: Basics (part 1)*. Kudelski Security Research. Retrieved April 25, 2023, from <https://research.kudelskisecurity.com/2017/11/01/zigbee-security-basics-part-1/>

Rudresh, V. (2018, January 23). *Zigbee security: Basics (part 2)*. Kudelski Security Research. Retrieved April 25, 2023, from <https://research.kudelskisecurity.com/2017/11/08/zigbee-security-basics-part-2/>

Z-Wave

Admin. (2023, March 20). *Z-wave explained: What it is and how it works*. DIY Smart Home Solutions. Retrieved April 25, 2023, from <https://www.diysmarthomesolutions.com/z-wave-wireless-networking/>

Brown, M., & Editor, E. (2018, January 8). *Z-wave 700 promises to give new smart home devices longer range and better battery life*. TechHive. Retrieved April 25, 2023, from <https://www.techhive.com/article/583133/z-wave-700-launch.html>

*Community - Silicon Labs*. Silicon Labs Community. (n.d.). Retrieved April 25, 2023, from <https://community.silabs.com/s/article/Introduction-of-Z-Wave?language=en_US>

*Examining the S2 security protocol for Z-wave*. Alarm Grid. (n.d.). Retrieved April 25, 2023, from <https://www.alarmgrid.com/blog/examining-the-s2-security-protocol-for-z-wave>

*Q&A: The lowdown on Z-wave’s S2 security support - electronic design*. (n.d.). Retrieved April 25, 2023, from <https://www.electronicdesign.com/technologies/embedded/article/21805285/electronic-design-qa-the-lowdown-on-zwaves-s2-security-support>

Timothy, M. (2022, November 12). *What is Z-wave and how does it work?* MUO. Retrieved April 25, 2023, from <https://www.makeuseof.com/what-is-z-wave/>

*Z-wave security - silicon labs*. Z-Wave Security - Silicon Labs. (2022, August 17). Retrieved April 25, 2023, from <https://www.silabs.com/wireless/z-wave/specification/security>

LoRaWAN

*Denial-of-service-attacks-on-lorawan/thesis document v1\_0.pdf at master ...* (n.d.). Retrieved April 25, 2023, from <https://github.com/E-v-Es/Denial-of-Service-Attacks-on-LoRaWAN/blob/master/Thesis%20Document%20v1_0.pdf>

*End device activation*. The Things Network. (2021, November 26). Retrieved April 25, 2023, from <https://www.thethingsnetwork.org/docs/lorawan/end-device-activation/>

LoRaWAN Backend Interfaces. (n.d.). Retrieved April 25, 2023, from <https://hz137b.p3cdn1.secureserver.net/wp-content/uploads/2020/11/TS002-1.1.0_LoRaWAN_Backend_Interfaces.pdf?time=1681265365>

*Lorawan OSI reference. | download scientific diagram - researchgate*. (n.d.). Retrieved April 25, 2023, from <https://www.researchgate.net/figure/LoRaWAN-OSI-reference_fig2_336619369>

LoRaWAN Specification v1.1. (n.d.). Retrieved April 25, 2023, from <https://hz137b.p3cdn1.secureserver.net/wp-content/uploads/2020/11/lorawantm_specification_-v1.1.pdf?time=1680130768>

*Message types*. The Things Network. (2021, December 13). Retrieved April 25, 2023, from <https://www.thethingsnetwork.org/docs/lorawan/message-types/>

*Security*. LoRa Alliance®. (2022, November 19). Retrieved April 25, 2023, from <https://lora-alliance.org/security/>