# On Building Automation System Security

Christopher Morales-Gonzalez[a], Matthew Harper[a], Michael Cash[b], Lan Luo[c*], Zhen Ling[d], Qun Z. Sun[b], Xinwen Fu[ab]

[a]University of Massachusetts Lowell, 1 University Ave, Lowell, 01854, MA, USA
[b]University of Central Florida, 4000 Central Flordia Blvd, Orlando, 32816, FL, USA
[c]Anhui University of Technology, 59 Hudong N Rd, Huashan District, Maanshan, 243099, Anhui, China
[d]Southeast University, 2 Sipailou, Xuanwu, Nanjing, 210018, Jiangsu, China

**Abstract**

Building Automation Systems (BASs) are seeing increased usage in modern society due to the plethora of benefits they provide such as automation for climate control, HVAC systems, entry systems, and lighting controls. Many BASs in use are outdated and suffer from numerous vulnerabilities that stem from the design of the underlying BAS protocol. In this paper, we provide a comprehensive, up-to-date survey on BASs and attacks against seven BAS protocols including BACnet, EnOcean, KNX, LonWorks, Modbus, ZigBee, and Z-Wave. Holistic studies of secure BAS protocols are also presented, covering BACnet Secure Connect, KNX Data Secure, KNX/IP Secure, ModBus/TCP Security, EnOcean High Security, ZigBee Pro and Z-Wave Plus. We point out how these security protocols improve the security of the BAS and what issues remain. A case study is provided which describes a real-world BAS and showcases its vulnerabilities as well as recommendations for improving the security of it. We seek to raise awareness to those in academia and industry as well as highlight open problems within BAS security.

*Keywords:* Building automation system, BAS protocols, security, attack

## 1. Introduction

A Building Automation System (BAS) is a type of cyber physical system whose purpose is to automate numerous processes such as maintaining heating, ventilation, and air conditioning (HVAC) controls, granting physical access through electronic locks, and lighting control within a building. BASs can be found controlling nuclear power plants [1], maintaining the climate in medical facilities [2], ensuring the operation of an energy grid [3], enabling a city resource management system [4] and use in smart homes [5].

A BAS is often based on the Open Systems Interconnection (OSI) model given their history. Wired BAS protocols such as Building Automation and Control Networks (BACnet), KNX, LonWorks and Modbus were created in 1995, 1999, 1988 and 1979 respectively. Wireless BAS protocols such as EnOcean, ZigBee, and Z-Wave were created in 2012, 2003, 1999 respectively. Recall that TCP/IP became popular after the release of its source code into the public domain by UC Berkley in 1989. Many of the BAS protocols now provide TCP/IP support given the convenience of the Internet. BACnet is the predominant communication standard in smart building automation with an estimated market share of 60% [6].

The rising adoption of BASs in modern society is accompanied by an increase in functional complexity. This complexity leads to a deeper integration of BASs into everyday operations, rendering them attractive targets for potential attackers. There have been numerous attacks against BASs. For instance, in 2016, attackers targeted the central heating and hot water systems of a Finnish facilities services company [7]. In 2021, hundreds of building automation control devices of a German engineering company were fully locked, forcing manual operation of the BAS [8, 9]. Most recently in June 2022, a BAS was targeted by hackers using an advanced persistent threat against the BAS engineering computers which allowed access to the main network. [10]. This notoriety creates a pressing need for a comprehensive review of their security.

In this paper, we provide a comprehensive, up-to-date survey on the types of BASs and their corresponding security landscape. It encompasses both established technologies and emerging ones, including new and upcoming secure communication protocols that may not be widely adopted. We present the network architectures of seven popular BASs, covering four wired BASs (BACnet, KNX, LonWorks, and ModBus) and three wireless BASs (EnOcean, ZigBee, and Z-Wave) and highlight the similarities and differences in their network architectures. Additionally, since BASs often do not use cables and connectors

*Corresponding author.

like those for Ethernet, we list each BAS protocol's supported communication mediums.

We provide a thorough review of many types of attacks against BASs including: brute-force attacks, covert channel attacks, cryptographic attacks, device reprogramming attacks, denial-of-service (DoS) attacks, eavesdropping attacks, false data injection (FDI), fuzzing attacks, man-in-the-middle (MITM) attacks, physical attacks, reconnaissance attacks, replay attacks, spoofing attacks, and side channel attacks. Most of the attacks are against the insecure BAS protocols and devices, which are prevalently deployed in real-world buildings.

In light of various attacks against BAS protocols, security extensions have been created for many BAS protocols, including BACnet Secure Connect, KNX Data Secure, KNX/IP Secure, ModBus/TCP Security, EnOcean High Security, ZigBee Pro and Z-Wave Plus. We analyze these secure protocols and discuss if the protections provided in the standard mitigate vulnerabilities discussed in this work against their insecure variants. We find details are often missing in securing BAS devices. For example, there are no detailed guidelines for securing storage secure sensitive information such as encryption keys. No secure protocols discuss the use of secure boot to ensure programs on the BAS devices cannot be modified. There is also a significant gap in securing BAS networks that are not strictly IP-based. Only one protocol extension–KNX Data Secure–attempts to secure twisted-pair, radio frequency, power-line and IP communication media by securing application layer data.

We also provide a case study and conduct a vulnerability assessment of a real-world BAS, which contains two buildings. The generalized and simplified architecture of the BAS is presented. The backbone network used between and within the buildings is a BACnet/IP network connected via Ethernet cables. A BACnet sub-network utilizing Master-Slave-Token-Passing (MS/TP) communications can be attached to the backbone BACnet/IP network through a controller, in which various MS/TP devices are interconnected with RS-485 cables. A controller within the BACnet MS/TP sub-network provides access to an additional KNX sub-network whose devices are interconnected with twisted-pair (TP) wires plugged into the various devices' KNX Red-Black block connectors. Insecure BAS protocols and devices are used in the BAS and are subject to the attacks discussed in this paper while firewalls and VLANs are used to limit the access to the BAS. We provide recommendations to secure this BAS installation from cyber attacks as these attacks against BASs are becoming more frequent.

Although there are efforts to perform surveys on BASs and their security including attacks against them [11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25], we seek to fill crucial gaps when looking at all surveys together and make the following major contributions. (i) While existing surveys are disjoint in that each focuses on a specialized subset of BAS protocols (i.e., wireless-only, wired-only, one protocols, two protocols, etc), we provide a unified survey that provides an overarching view of seven popular BAS-centric protocols and abstain from the more nuanced protocols which are typically associated with IoT rather than BAS such as Thread, Bluetooth, and Wi-Fi. (ii) Our work addresses the issue of not having an overview of BAS network architectures and goes further as it generalizes them into simplified architectures to highlight similarities and differences between them. (iii) While existing surveys are extensive in the types of attacks, we carry out a careful study to determine the root causes of the attacks against the vast array of BAS protocols. Additionally, we seek to combine all the attacks into a single work to discover commonalities between the types of attacks against the different protocols. (iv) One prevalent gap that is apparent across all the surveys mentioned is the lack of a discussion regarding the secure protocols that have been developed and published for these BAS protocols. Our work fills in this gap by providing a detailed, holistic study of the various secure extensions and standards developed for these common BAS protocols while simultaneously evaluating how the defenses provided address the vulnerabilities affecting their insecure variants; highlighting the ones that still exist.

The rest of this paper is organized as follows. In Section 2, we introduce the network architecture of various BASs. Attacks against those BASs are surveyed in Section 3. Holistic studies on BAS secure extensions are provided in Section 4. Using our findings, we discuss open problems in Section 5 as well as provide a case study in Section 6 and finally conclude this paper in Section 7.

## 2. Building Automation Systems

In this section we introduce a total of seven popular wired and wireless BASs with a focus on their network architectures to highlight similarities and differences between them.

### *2.1. Overview*

Figure 1 gives a simplified three-level BAS architecture. The Management level encompasses operator stations, monitoring units, programming units, and other peripheral devices linked to a server to facilitate the monitoring and management of information exchange within the automation system. For example, Siemens Desigo CC is a building management system that visualizes and controls devices in buildings [26]. The Automation level typically constitutes a specialized communication network for networking and control (automation). For example, specific controllers can be used to run a control schedule such as

Table 1: BAS protocol's supported communication mediums

| Protocol | Supported Mediums | Connectors |
|---|---|---|
| BACnet | MS/TP, ARCnet, Ethernet, Point-to-Point, LonTalk | RS-485 (MS-TP), Coax/TP/Fiber/93 Ohm RG-62 (ARCnet), RJ-45 (Ethernet), RS-232 [DB-9](P2P), LonWorks Connectors |
| KNX | TP, Power Line (PL), Radio Frequency (RF), KNX/IP (Ethernet) | TP-1 Cables, PL Connectors, ISM Radio Transmitter, Ethernet |
| LonWorks | TP, PL, RF, Fiber, Coax, LonTalk IP | TP-1 Cables, PL Connectors, ISM Radio Transmitter, Ethernet, Coax Cable, Ethernet |
| Modbus | *Medium-Independent (This is an application layer protocol)* | |
| ZigBee | Wireless (IEEE 802.15.4 - LR-WPAN) ISM Band | ISM Radio Transmitter |
| Z-Wave | Wireless ISM Band | ISM Radio Transmitter |
| EnOcean | Wireless ISM Band | EnOcean Radio Transmitter |

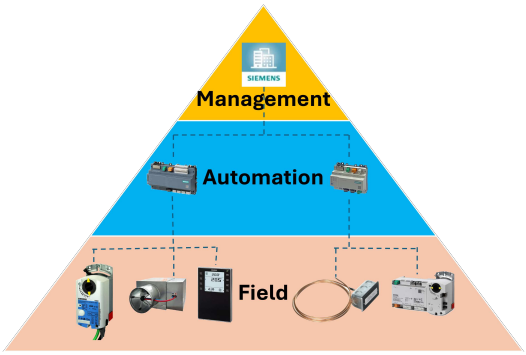


Figure 1: Three Layer BAS Architecture

switching off lights after work hours. The Field level consists of sensors, actuators, field controllers and other physical devices. For example, a field controller may collect sensor data, which can be fed into the controllers at the automaton level for optimization of HVAC schedules. The field controller may also perform control of actuators.

A BAS network is often based on the Open Systems Interconnection (OSI) model given their history. There are wired BAS protocols such as BACnet, KNX, LonWorks and Modbus and wireless BAS protocols such as EnOcean, ZigBee, and Z-Wave that were created. Many of the BAS protocols now provide TCP/IP support given the convenience of the Internet, e.g., BACnet/IP and KNX over IP. In Figure 1, entities on the management level and automation level are often interconnected through IP versions of BAS protocols. Entities on the field level can be interconnected or connected to controllers on the automation level through non-IP version of BAS protocols such as BACnet MS/TP and KNX TP. A non-networked field device may be connected to a controller through a RS232/485 serial interface.

In this section, we focus on how BAS devices are interconnected as a network and often ignore the management level components.

### 2.2. Wired BAS

Fig. 2 illustrates an example wired BAS network. There can be multiple types of physical mediums within a BAS and are not restricted to use only Ethernet. Table 1 presents the supported communication mediums for wired protocols. BACnet, KNX and LonWorks network architectures are based on the OSI model while Modbus is only defined for the application layer of the OSI model. BACnet, KNX and LonWorks have their own routing protocols and routers for their local networks while special BAS/IP routers can be used to interconnect multiple BASs together using the Internet. Modbus is different as it uses gateways for Internet access rather than routers.

#### 2.2.1. BACnet

A BACnet BAS is logically separated into three main portions: *internetworks*, *networks* and *segments* [27]. A BACnet *segment* consists of physical electrical media to which BACnet devices are connected to. A BACnet *network* has one or more BACnet *segments* that are connected via bridges. Multiple BACnet *networks* can be connected with BACnet *routers* to form a BACnet *internetwork*.

Fig. 2 can be used to illustrate a sample BACnet internetwork, which connects two BACnet networks - Network A and Network B - with BACnet/IP routers which communicate via the Internet. Network A can utilize the BACnet MS/TP protocol for its segment using RS-485 connectors. Network B can use the BACnet/IP protocol with Ethernet cables. The Programmable Logic Controllers (PLCs) gather the telegrams from the devices under their control and may send out these values to the other network.

A sample BACnet testbed is shown in Fig 3. This testbed represents a single BACnet network. There are two segments within this network; one utilizing BACnet/IP (which sends BACnet messages within IP packets) and the other utilizing BACnet MS/TP. The communication between the two segments is facilitated via a BACnet/IP-to-MS/TP router. In the BACnet MS/TP segment, there is a BACnet Controller and an Air Quality Sensor. There is a single BACnet/IP to MS/TP Router that makes up the BACnet/IP segment of the network. From a communications standpoint, the BACnet MS/TP Air Quality Sensor communicates over the twisted pair (TP) wire (that is attached to itself and the segment's router) using BACnet MS/TP messages. When the message reaches this router, it converts

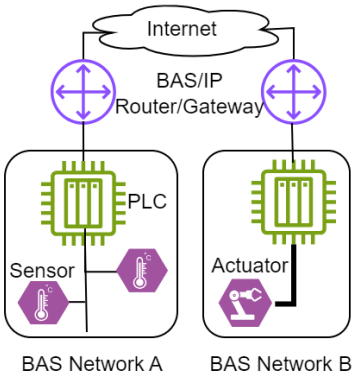


Figure 2: Example Wired BAS Network

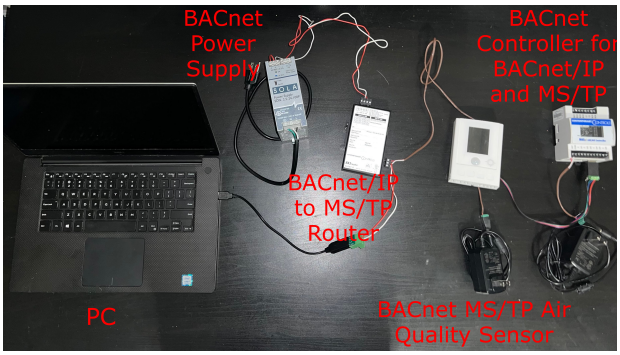


Figure 3: BACnet Testbed

this information into a BACnet/IP message and sends it out onto the BACnet/IP segment. The setup uses three power supplies that provides power through direct connections to the devices.

IP and the BACnet MS/TP segments. The BACnet MS/TP Air Quality Sensor communicates through the router's MS/TP terminal ports and is powered by a separate power supply. Also apart of the BACnet MS/TP communication segment is the BACnet Controller for BACnet/IP and MS/TP. This controller may communicate with other BACnet/IP networks, such as with the router, but it may also communicate with MS/TP segments using its USB port. The connection from the controller to the Air Quality Sensor is done using a USB-to-RS485 adapter. Similar to the Air Quality Sensor, power for the BACnet Controller is also provided by a separate power supply. These power supplies use terminal connection adapters to provide power to the sensor and the controller, respectively. TP cables connect the router, sensor, and controller together for MS/TP communication. Additional TP cabling is also used to power the BACnet/IP to MS/TP Router by its power terminals from the BACnet Power Supply. Together, this testbed forms a BACnet internetwork.

### 2.2.2. *KNX*

A KNX [28] BAS is logically separated into three portions: 1) *Domain*, 2) *Area*, 3) *Line*. A *Domain* is made up of connected *areas*. An *Area* is made up of a series of connected *lines*. A *Line* is a culmination of many KNX devices (up to 256). Individual lines can have the same or different communication mediums but they must be connected by a line coupler(s). The same premise can be applied for connecting multiple areas with area couplers to create a domain.

Fig. 2 can be used to illustrate a sample KNX BAS network. It is a single domain KNX network made up of two areas interconnected by two KNX/IP routers. In the two areas is a single line of devices interconnected with one another with a common communication media such as TP1 wires. The routers can be connected to each other through the Internet to facilitate communications between the two areas.

Figure 4 showcases a sample KNX BAS network which is a single domain made up of one area and one line. Within the BAS, it contains a temperature sensor, a presence detector and an actuator which is directly connected to a damper unit. Additionally, there is a KNX/IP interface which allows a PC or a similar device to connect to it through Ethernet and use it to send out KNX messages onto the main TP network it's connected to. There are also Raspberry Pis in which one (left) is connected to the KNX network through the KNX/IP interface and the other (right) is connected via a KNX Raspberry Pi HAT which attaches to the pi's general purpose input output (GPIO) pins. All devices are connected with each other through TP1 wires inserted into the KNX Red-Black Connectors present on each device. Finally, the devices are powered by an external KNX power supply; a transformer is required to power the actuator to open and close the damper.

### 2.2.3. *LonWorks*

A LonWorks [29] BAS consists of a peer-to-peer network that is logically separated into two major sections: 1) *Domains* and 2) *Subnets*. The *Domains* logically separate LonWorks networks and subnets. *Subnets* can be used to separate devices in a domain. The topologies available to the LonWorks network are dependent on the physical mediums that are used. LonWorks provides support for twisted pair, ethernet, power line, fiber optic, and radio frequency. The network stack and thus, the network topology of a LonWorks network changes slightly depending on the physical medium used.

Fig. 2 can be used to illustrate a sample LonWorks BAS network with two subnetworks (Networks A and B). Network A can use the twisted-pair communication medium and Network B can use the Ethernet communication medium. The sensors and other end devices in a LonWorks network can communicate with one another within

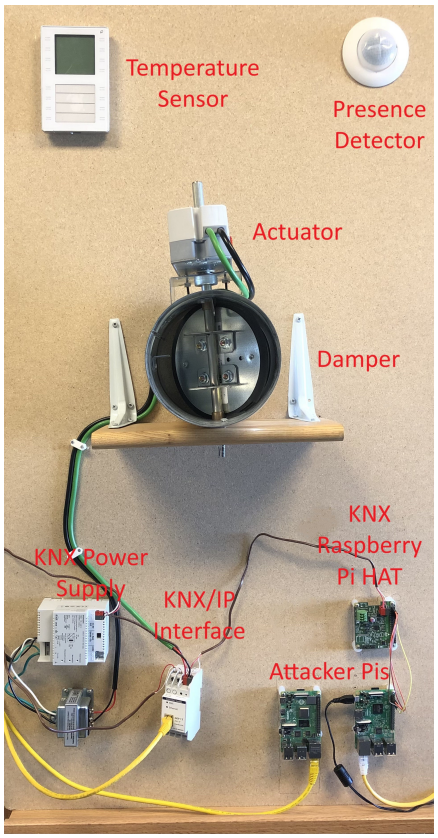


Figure 4: Sample KNX Testbed

the same domain. Inter-domain communications are facilitated through routers.

#### 2.2.4. *Modbus*

Modbus [30] is an application-layer-level protocol. The network model is defined based on the underlying data link communication protocol (i.e. RS-485) and the communication scheme is known as *Modbus on X* where *X* is that communication protocol, e.g., Modbus on RS-485 serial communication. Modbus TCP/IP is the Ethernet-oriented variant of the protocol that uses the Internet for communication between servers and clients such as sensors and actuators.

Fig. 2 can be used to illustrate a sample Modbus network. Network A can use the MS/TP RS-485 connectors, communicate using MS/TP and connect to a ModBus RS-485/IP Gateway to communicate with the other network. Network B can use serial port RS-232 cables to enable communication and connect to a ModBus RS-232/IP Gateway to speak with the other network.

### 2.3. *Wireless BAS*

ZigBee [31], Z-Wave [32], and EnOcean [33] are also all based on the OSI model and designed for low-power wireless communications. There has been limited effort to integrate ZigBee and EnOcean with the Internet while gateways can be used in Z-Wave networks for communications over IP networks such as the Internet.

#### 2.3.1. *ZigBee*

ZigBee is designed for wireless personal area networks (WPANs) and can only communicate in the Industrial, Scientific and Medical (ISM) and 2.4 GHz frequency ranges [31]. It provides support for three network topologies; namely, star, tree (shown in Fig. 5) and mesh. Regardless of the topology chosen, there are three entities that are present inside a ZigBee network at all times: *coordinator*, *router*, and *end device*. Fig. 5 shows an example ZigBee BAS network with a simple tree topology. The ZigBee coordinator is the root of the tree and the router can forward messages to/from devices that the coordinator is not directly connected to. The end devices cannot talk to another end device on the network except for its parent node which must be a router or coordinator.

#### 2.3.2. *Z-Wave*

Z-Wave is a wireless communication protocol that operates on a master-slave model in a mesh network topology. There are two classes of devices: the *controllers* issue commands to the *slave devices*. The *slave devices* perform operations or report information as requested by the controller. The logical separation of a Z-Wave network is provided by a 32-bit *HomeID* and nodes (controllers and slave-devices) are identified using the tuple of the *HomeID* and a unique 8-bit *NodeID*.

Fig. 6 shows a sample Z-Wave BAS network. The slave devices report back to their controllers, which process and handle the data. The Z-Wave Gateway is one of these controllers and it may report to a user's client (i.e. phone) or forward the information to another gateway. Slave devices may have the capability to act as routers in the mesh network where their routing behaviors are set by the controllers.

#### 2.3.3. *EnOcean*

EnOcean is a wireless protocol based on the Low-Rate Wireless Personal Area Network (LR-WPAN) that supports mesh, star or point-to-point topologies in which all communications can only utilize the ISM bands [33]. EnOcean devices have to register with each other to communicate. Due to the nature of LR-WPAN and its limited range, EnOcean does not contain mechanisms for network segmentation. The only requirement for a device to communicate on an EnOcean network is an EnOcean radio transmitter. Fig. 7 shows a sample EnOcean BAS network in which all the devices get the same messages. However, the devices only accept messages if the sender's address has been verified; this is depicted by the dotted lines in the figure.

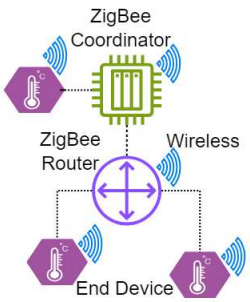


Figure 5: Example ZigBee Network

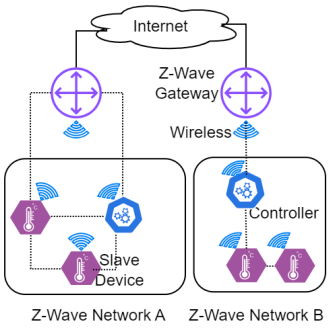


Figure 6: Example Z-Wave Network

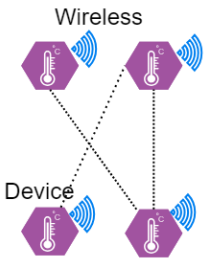


Figure 7: Example EnOcean Network

## 3. Attacks against BAS

In this section, we first discuss attack surfaces present in BASs and then review existing attacks against BASs.

### *3.1. Attack Surfaces*

A BAS contains various components such as *physical*–physical components that make up the device such as wires and devices, *firmware*–low level program that control's a device's hardware components, *software*–applications used to perform high-level operations to carry out the function of a BAS, *network*–protocols which allow devices to transfer information and commands within the network to facilitate the functions of the BAS, and the *data* generated and stored within a BAS such as temperature readings and device configuration information. Attacks may break confidentiality, integrity, availability, authentication, non-repudiation and other security requirements of these components.

### *3.2. Attacks*

Table 2 provides an overview of the attacks against BASs and provides guidance for future research on BAS security. For example, we find there is little work on the secure extensions and software security of a BAS.

#### *3.2.1. Physical Components*

The physical components of a BAS will be comprised of the devices themselves such as temperature sensors and actuators as well as physical wires that interconnect the devices and physical battery banks if applicable. During a physical attack, the physical components of the BAS are targeted. An attacker can have access to these for various reasons such as improper setup in which wires are exposed or panels hiding wires are easily removable. How they perform this attack can vary depending on the goal of the attacker. For example, an attacker could seek to manipulate, tamper, damage or destroy a physical component to carry out a DoS attack violating the availability requirement on the service that particular component provides [41, 34, 35, 36].

*Hardware Attack*: We are particularly interested in attacks that exploit the hardware such as circuits and interfaces of a device so as to hack into the device and BAS. In [67], an attacker split a TP wire and connected their Raspberry Pi to it; granting access to the KNX network. They performed a FDI attack that could reach the BACnet network that the KNX network was attached to. Other works have also accessed a KNX network through exposed TP wires in buildings [37, 14]. Authors have extracted a Z-Wave device's firmware and from it, as the firmware was not encrypted, extracted information such as various keys that were in use for normal communications as well as modifying the firmware [86, 74, 48, 47, 49]. This was also done to ZigBee devices in [43, 44, 45]. Some other methods against ZigBee include performing physical jamming attacks through a Software-Defined Radio (SDR) [40] and draining a device's battery [42]. Other physical attacks that have been done on protocols include standing next to a smart card reader and using a SDR to attack the card reader on a LonWorks network [38] as well as simply reading the sticker on an EnOcean device that contains the device's key used when adding it to an EnOcean network [46].

#### *3.2.2. Software/Firmware*

In this paper, firmware refers to the code that exists on the device while software refers to BAS management and control programs such as ETS [96] from the KNX organization and Desigo CC [26] from Siemens. Attackers can have various goals that could seek to modify software/firmware or cause unexpected behavior on the device.

*Device Reprogramming.* If an attacker gains access to a device and has control over the device's firmware or software due to poor software security practices such as not using technologies such as Secure Boot which results in an attacker having the ability to perform unauthorized modifications to the device, they can make the device perform actions that they were not originally programmed to do; violating the integrity requirement. Some researchers reprogrammed a device entirely in a KNX network [89], and Morgner et al in [52], developed a framework called *Z3sec*

Table 2: Attacks against BAS in the bibliography

| Attack Surface | Attack | BACnet | EnOcean | KNX | LonWorks | ModBus | ZigBee | Z-Wave |
|---|---|---|---|---|---|---|---|---|
| **Physical Components** | **Physical Attack** | [34, 35, 36] | [34] | [37, 14, 34] | [38] | | [39, 40, 41, 42, 43, 44, 45, 34, 46] | [47, 48, 49] |
| **Firmware/Software** | **Device Reprogramming** | [50] | | [37] | | [51] | [52, 46] | [49, 48] |
| | **Fuzzing Attacks** | | | [53] | | | [54] | [55, 56] |
| **Network** | **Covert Channel** | [57] | | | | [58] | | |
| | **Cryptographic Attacks** | | | | | | [59, 43] | |
| | **DoS** | [34, 22, 23, 36, 50] | [34] | [37, 14, 60, 53, 34] | [11, 61] | [51] | [62, 63, 13, 40, 43, 45, 34, 24, 64] | [65, 49, 66, 46] |
| | **False Data Injection** | [67] | [68] | [60, 69, 67] | [11] | [51, 70, 71] | [72] | [73, 74] |
| | **MITM** | [35] | | [14, 67, 75] | [76] | [51, 77] | [78, 72, 24] | [79] |
| | **Reconnaissance** | [80, 81, 75, 36, 50, 82] | | [75] | | [51, 83] | [84, 52, 41, 72] | [79] |
| | **Replay Attack** | | [68] | [67, 75] | [76] | [51, 71, 70] | [62, 85, 41, 43, 46] | [86, 87, 49] |
| | **Spoofing** | [36] | [24] | [60] | [11, 76] | [51, 83] | [62, 78] | [79, 88, 87, 49, 24] |
| **Data** | **Brute Force Attack** | | | [89] | [12, 11, 38] | | | |
| | **Eavesdropping** | [23] | [90, 91, 68] | [37, 14, 69, 67, 89] | [12, 11, 61] | [51] | [62, 59, 43, 63, 20, 13, 40, 41, 52, 44, 72, 45, 46] | [79, 74, 88, 49, 87, 92] |
| | **Side Channel** | | [93, 91] | | | [94] | | [95] |

that had the capability to factory reset any device or block a device permanently in a ZigBee network violating the integrity, availability, authentication and non-repudiation requirements.

*Fuzzing.* Fuzzing is a technique that is used to find security flaws in software and hardware in hopes of breaching the integrity of the device while potentially breaching confidentiality and availability. The basic principle of fuzzing is to generate inputs rapidly and automatically, send them to a target and observe the response. This can either be done randomly or by adhering to the standards used by the target. An example of the latter is a smart fuzzer that would input a valid message code and then try to input invalid data to see if the target device crashes or interprets it wrong. Existing fuzzing works target KNX/IP [53], ZigBee [54], and Z-Wave [55, 56] networks.

### *3.2.3. Network*

Network attacks will try to focus on using the implementation of the BAS network to achieve various goals. For example, an attacker may seek to overwhelm devices on the network through the use of a DoS attack or seek to tamper with information being sent throughout the network through a MITM attack. The appeal of network attacks to attackers is that everything can be done remote if they have access to the BAS network.

*Covert Channel Attack.* Covert channels misuse existing systems and procedures to establish unauthorized communication channels. For example, if a protocol reserves a field to be left as empty or has "future use" fields, then a covert channel can be established by placing messages into that field that another entity can parse out and interpret which breaks the confidentiality requirement if sensitive information is extracted. This method of using reserved bits in messages to share data was found to be present in both BACnet networks [57] and Modbus/TCP networks [58]

*Cryptographic Attack.* Some cryptographic functions used in BASs are inherently insecure. An example is the XOR operation for encryption as this was shown to be vulnerable in the encryption scheme of ZigBee in [59]. Attackers have also exploited the reutilization of nonce values in ZigBee's encryption algorithm to extract the original plaintext [43] violating the confidentiality requirement which may lead to later attacks which break the authentication and non-repudiation requirements.

*Denial-of-Service Attack.* Denial-of-service (DoS) and *distributed* DoS (DDoS) attacks are also utilized against BASs which affect their availability. (i) *Resource Consumption*: Resource consumption is one of the most popular methods used in a DoS attack. All physical devices have limited resources - BAS devices even more so as they are not typically expected to have the capabilities of a desktop for ex-

ample. When all of those resources are allocated, then the device won't be able to handle any more requests. Works such as [39, 42] have shown that the battery on devices is a vulnerable target as attackers are able to drain it and stop the device from functioning entirely. (ii) *Jamming*. A jamming attack is one where an attacker sends out frequencies that cause the legitimate frequencies to either drop or cause enough interference to not be processed correctly as was done in ZigBee networks [97, 40, 64]. This is particularly dangerous for wireless-only BAS protocols as they rely solely on radio signals to deliver their data. The plausibility of this attack is high as the attack can be carried out with a SDR that can be purchased easily.

*False Data Injection.* In a FDI attack, an attacker injects false data into the network violating the integrity and authentication requirements. Having a proper authentication method for devices to send messages on the network is often overlooked in these BAS protocols. The requirement for an attacker is that they gain access to the BAS network. This can be done through physical access methods such as connecting a device onto the bus via a TP-wire for a hybrid BACnet-KNX system [67]. The attackers then abuse the lack of authentication mechanisms present on both KNX and BACnet sent messages to inject false temperature readings onto the network incurring energy costs. Other methods we reviewed which carry out a FDI use the following methods: using an old key that is still accepted by a BACnet network [50], finding and using the correct function codes in a Modbus/IP network [51] or directly connecting into the switch [70], hacking the wireless access point for a KNX/IP network [60, 69], using an EnOcean developer kit [68], using specialized USB sticks for ZigBee [72] and Modbus [71], sniffing out the network key for LonWorks [11], using a specialized C1110 chip or a SDR [73, 74] for Z-Wave networks.

*Man-in-the-Middle.* In a MITM attack, the attacker is able to place themselves between two communicating parties and can perform a suite of actions including interception, interruption, modification, or fabrication all of which affect the aforementioned security requirements. The major underlying reason that these attacks are plausible is due to a lack of entity authentication. Some other methods used to gain this access include Address Resolution Protocol (ARP) poisoning or Content Addressable Memory (CAM) table attacks for protocols that use the IP-based communications such as KNX/IP, BACnet/IP and Modbus TCP [22]. Other methods include gaining physical access to the TP wires of a KNX-BACnet system and placing two Raspberry Pis into it [67]. (i) *Interception*. The attacker may choose to simply perform the equivalent of an Eavesdropping attack. In [49], Kim et al. used this capability to register a rogue device after they had sniffed out the network key from a Z-Wave network while Cash et al. in [67] simply forwards the messages in their attack on a KNX-BACnet system. (ii) *Interruption*. Because the messages have to go through the attacker's machine, the attacker can drop the packet to ensure that the message doesn't reach the intended destination. In a BAS, this could be dropping a message from a management app such as the IKEA Home Smart app used in a ZigBee network [78]. (iii) *Modification*. As the attacker has full control of the messages, the attacker can modify a portion of the message or create a new message entirely to cause some desired behavior. In a BAS, this could be modifying normal data between a router and controller on a Z-Wave network [49], modifying normal sensor traffic in a simulated LonWorks environment [76], modifying command/response messages in a Modbus network [51, 77] or a BACnet network [35]. (iv) *Fabrication*. Because the attacker has a communication channel with both parties, they can send a message to one entity as the other. In [49], Kim et al. are able to send messages as the controller in a Z-Wave network to start a pairing process for a remote device while others were able to impersonate any device on a LonWorks network [76] or a ZigBee network [24].

*Reconnaissance Attack.* In a reconnaissance attack, the attacker is trying to actively gather information about a network or device that they're targeting. During this process, they may discover sensitive information which would violate the confidentiality requirement. In the case of these BAS protocols, the notion of "implied trust" (i.e. no entity authentication mechanism) by being on the network allows an attacker to carry out these attacks. Some key information that an attacker may want includes: the number of other devices on a network, getting the other devices' serial numbers or figuring out which application version a device is running.

These attacks have been done before in [80, 75, 81] in which BACnet's *Who-Is* requests are sent to find BACnet devices remotely. BACnet device enumeration was done in [82] where Cash et al. made a tool to enumerate a device's object and property lists automatically. Attacks on KNX networks follow a similar idea in which they use the inherent *discovery* requests for KNX/IP servers [80, 37, 75]. Modbus researchers performed scans for the different controllers on the network [51, 83]. ZigBee researchers actively search for different networks and devices through the *beacon request* and *scan* services of ZigBee [84, 41, 52, 72]. Badenhop et al. in [79] describe using the *Get NL* primitive of Z-Wave to get the neighbor list of devices to gather a network topology [79].

*Replay Attack.* In a replay attack, a valid previous message can be resent onto the network and will be accepted by the network or the entity that the original message was intended for - violating the integrity requirement. A majority of BAS

protocols do not have any type of sequence number mechanism to prevent these types of attacks. As such, this type of attack was carried out by Fuller et al. in [87] in which they were able to attach a rogue controller to a Z-Wave network and replayed messages to reset a device into pairing mode. This was also shown in [86] in which Merdis was able to use an SDR, logic analyzer and multimeter to replay packets in a Z-Wave network. Other methods exist and have been acknowledged for EnOcean [68], KNX [67, 75], LonWorks [76], Modbus [51, 71, 70], and ZigBee [86, 87, 49].

*Spoofing.* In a spoofing attack (sometimes called an impersonation attack), an attacker impersonates a device on the network and sends out/responds to messages as the original device; violating the authentication requirement. Because of the lack of entity authentication in these BAS protocols, spoofing is a major issue that has been acknowledged and exploited in BACnet [36], EnOcean [24], KNX [60], LonWorks [11, 76], Modbus [51, 83], ZigBee [62, 78], and Z-Wave [79, 88, 87, 49, 24] networks.

#### *3.2.4. Data*

Data is an broad term that we use to denote the information that the BAS itself contains and produces while also accounting for leaks of information such as global encryption keys.

*Brute-Force Attack.* In brute-force attacks, an attacker tries to gain access to a system by guessing credentials or encryption keys. The implementation of the protocol or the protocol itself may not contain mechanisms to protect against these attacks such as a timeout or contain a key computationally-infeasible component. This attack can also be carried out by using default credentials that are used across components such as the leaked default master key for the ZigBee Light Link Profile [39] which violates the confidentiality requirement. This method was also showcased in [38, 12, 11].

*Eavesdropping.* All insecure BAS protocols surveyed suffer from this type of attack as shown in multiple works [23, 90, 91, 68, 37, 14, 69, 67, 89, 11, 61, 51, 62, 59, 43, 63, 20, 13, 40, 41, 52, 44, 72, 45, 98, 46, 79, 74, 88, 49, 87, 92] as BAS protocol communications are typically unencrypted violating the confidentiality requirement.

*Side Channel Attacks.* Side channel attacks are those that utilize the normal implementation of systems to gain access to restricted information which violates the confidentiality requirement. For example, Jonas et al. in [93] find there is a side channel attack in EnOcean networks because there is unintentional data leakage from the signals that are sent from the open and close signals. The different signals had different lengths. Therefore, an attacker doesn't need to know the specifics of the message being sent. They only need to look at the lengths and understand that one is shorter than the other to understand which message was sent. Tsalis et al. also described the reality of side channel attacks in the ModBus protocol with regards to the time between packet intervals [94]. Liou et al. in [95] had discussed the reality of side channels within the ZigBee protocol based on the packet interval, number of packets sent and total packet size.

## 4. Secure BAS Protocols

In light of the various attacks and security research done against BAS protocols, many BAS protocol developers have created security extensions. We provide the first holistic analysis a holistic analysis on these protocols to understand how they work. In this analysis, we discuss how the network model is affected, what key scheme is used, what is now implemented in new devices that allow the secure protocols to work and finally perform a security analysis in which we investigate if the protections provided in these secure standards mitigate vulnerabilities in their previous versions. We will use the following definitions: *Full Mitigation* - The original vulnerability that led to an original attack has been secured and is not present in the solution. *No Mitigation* - The original vulnerability has not been mitigated and the proposed solution may introduce additional vulnerabilities. *Unknown* - At the time of this holistic analysis, we were not able to determine if the attacks were fully mitigated. Further experimentation and research with testbeds utilizing the secure protocol will be required. We first discuss the secure wired BAS protocols and then the secure wireless BAS protocols.

### *4.1. BACnet Secure Connect*

BACnet Secure Connect (BACnet/SC) is the optional security extension developed by ASHRAE and was published in 2019 in Annex AB of the BACnet Standard [27]. It is an application layer protocol developed for use in BACnet/IP networks only and utilizes the WebSockets technology - more specifically, the Transport Layer Security (TLS) variant. BACnet/IP data will be encapsulated in a secured WebSockets packet.

#### *4.1.1. Network Model*

A BACnet/SC network follows the centralized hub-and-spoke network model in which a central hub server (i.e., hub function) routes and forwards messages to different devices (i.e., nodes) on the network; However, BACnet/SC does state nodes can provide support for direct unicast connections that don't utilize the primary hub function to send messages. To address the single-point-of-failure issue that is present with a centralized architecture, BACnet/SC provides support for a *failover hub function* to which the

Table 3: External attacks mitigated via secure BAS protocols for BAS protocols: ✓= full mitigation, ✗= not mitigated, U = Unknown

| **Attacks** | **BACnet/SC** | | **KNX Secure** | | | | | | | | **LonWorks** | **Modbus** | | **EnOcean** | **ZigBee Pro** | **Z-Wave** | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | **KNX DS** | | | | **KNX/IP Sec** | | | | | | | | | **S0** | **S2** |
| *Communication Medium* | *IP* | *Other* | *TP* | *RF* | *IP* | *PL* | *TP* | *RF* | *IP* | *PL* | *Other* | *IP* | *Other* | *RF* | *RF* | *RF* | *RF* |
| Brute Force | | | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ | ✗ | ✗ | | | | | | |
| Covert Channel | U | ✗ | | | | | | | | | | U | ✗ | | | | |
| Cryptographic Attacks | | | | | | | | | | | | | | | U | | |
| Device Reprogramming | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ | ✗ | | ✓ | ✗ | | ✓ | ✓ | ✓ |
| DoS | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ | ✗ | ✗ | ✓ | ✗ | ✗ | ✓ | ✗ | ✗ |
| Eavesdropping | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ | ✗ | ✗ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ |
| False Data Injection | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ | ✗ | ✗ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ |
| Fuzzing Attacks | | | U | U | U | U | U | U | U | U | | | | | U | U | U |
| MITM | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ | ✗ | ✗ | ✓ | ✗ | | ✓ | ✓ | ✓ |
| Physical Attack | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | | | ✗ | ✗ | ✗ | ✗ |
| Reconnaissance | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ | ✗ | | ✓ | ✗ | | ✓ | ✓ | ✓ |
| Replay Attack | | | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ | ✗ | ✗ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ |
| Spoofing | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ | ✗ | ✗ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ |
| Side Channel | | | | | | | | | | | | U | ✗ | U | | ✗ | ✗ |

nodes connect to if the primary hub function becomes unavailable.

To join a BACnet/SC network, a device must establish a connection with the hub function. This process requires digital certificates, public keys, and private keys on both the hub function device and nodes to authenticate each other and will be used to establish a secure communication channel.

#### *4.1.2. Key Scheme*

Using the TLS-secured version of WebSockets will require the utilization of public and private keys during key exchanges as well as a valid digital certificate. The secure management of sensitive information such as the private key is "a local matter" [27]. There are no shared keys amongst the BACnet/SC nodes.

#### *4.1.3. Device Implementation*

BACnet devices that choose to participate in a BACnet/SC network implement the BACnet/SC Virtual Link Layer (BVLL) for link control. The BACnet/SC Node also implements the *hub connector* for connecting to the hub function to be a part of the BACnet/SC network. Nodes in a BACnet/SC network are able to establish direct connections with each other through TLS-secured WebSockets. This requires the use of certificates signed by a common third-party certificate authority (CA) or an internal CA used for the BACnet/SC network.

#### *4.1.4. Security Analysis*

A brief analysis of the security provided by BACnet/SC is presented. As BACnet/SC is designed for BACnet/IP networks, we will be looking at vulnerabilities from external threat actors for the IP communication medium. Indicating all the previous vulnerabilities on insecure BACnet networks apply to the other supported communication mediums noted in Table 1.

*Vulnerabilities Addressed. Device Reprogramming*: Because any device on an insecure BACnet network could send out management commands after initiating a unicast connection to a device as there were no authentication mechanisms, an attacker could send out these commands to reprogram and reinitialize a device. However, to establish a unicast connection in a BACnet/SC network, a valid certificate is required to pass the required authentication scheme. Because the attacker does not have a valid digital certificate, a unicast session cannot be established, thus the attacker cannot reprogram the device.

*DoS*: As there was no authentication on an insecure BACnet network, an attacker could route all of the traffic to itself using *I-Am-Router* messages that will say that the attacker is a router. Once subsequent traffic is given to the attacker, they can drop the packets and cause a DoS. With the introduction of a central hub function and the utilization of TLS with mutual authentication, an attacker cannot identify itself as a router to the other nodes on the network without first authenticating itself to the network; mitigating the vulnerability.

*Eavesdropping*: Because an insecure BACnet network did not have any encryption on it, an attacker could gain access to the network by listening for radio signals or ethernet signals and passively sniff out traffic. However, with the utilization of TLS, encryption of network communications is present; thus an attacker cannot sniff out traffic.

*False Data Injection*: Because there is no authentication mechanism for sending out information on an insecure BACnet network, an attacker could send out false data

readings onto the network and cause unintended behavior on the BAS. Authentication is inherently added with the usage of TLS with mutual authentication as the devices will not receive any encryption keys used for communicating on the network without having a valid digital certificate. Thus, an attacker cannot send information onto the network without acquiring a valid certificate.

*MITM*: Because there is no encryption, integrity checking, or authentication mechanisms on a base BACnet network, an attacker could perform the four different functions discussed in Section 3. However, with the introduction of TLS, the attacker can no longer fabricate, modify, interrupt or intercept packets as it will not be allowed into a BACnet network without proper authentication.

*Reconnaissance*: Because there is no authentication mechanism to send out information-gathering messages on an insecure BACnet network, an attacker could send out any number of messages to gather extensive information about the network. With the introduction of TLS, an attacker cannot request information from a device without a valid digital certificate and corresponding private key as it would require a unicast connection which needs to go through an initial TLS handshake.

*Spoofing*: As there is no authentication mechanism in an insecure BACnet network, an attacker could simply forge packets that have the same source address as a legitimate device and could send out information on the network as that device. With the introduction of digital certificates within the TLS protocol, this vulnerability is addressed as the attacker would need the valid certificate and corresponding private key of the device to be able to spoof them.

*Vulnerabilities Remaining. Physical Attack*: A BAS will always have physical components and attackers can utilize physical penetration testing methods to gain access to these devices. The BACnet/SC protocol recommends that a factory reset of a device should be able to be done via a physical method [27]. Since BASs may have components in publicly-accessible locations, the attacker can research the model of the device and perform this factory reset; simultaneously carrying out a DoS attack as the data the device should be reporting will not be reported. Another attack that is possible stems from the software security of the device created by the manufacturers. More specifically, the offline storage of sensitive information of a device must be considered as attackers can steal the device and extract crucial information from it such as various session keys, certificates and most importantly, the private key used to gain access into the network.

### *4.2. KNX Data Secure*

KNX Data Secure [99] is the optional security extension of the KNX protocol developed by the KNX Association published in 2013 that affects the application layer of KNX messages. KNX Data Secure will allow options for no security, confidentiality-only, message authentication-only or both. It is designed to be used on every communication medium and keep the application data secure from unauthorized entities.

#### *4.2.1. Network Model*

Because KNX Data Secure is an application layer protocol, the network model is not influenced by this security extension.

#### *4.2.2. Key Scheme*

There are various keys that are used throughout the communications that occur during the BAS' lifecycle when using KNX Data Secure. Each of these keys are 128 bits long and are going to be used in either an AES-128 CTR mode encryption scheme or AES-128 CBC-MAC signature scheme. The KNX Association has introduced the *Factory Default Setup Key (FDSK)* that will aid in securely distributing keys to be used in management and runtime communications. The FDSK is unique and can typically be found printed on a sticker placed on the physical device. The FDSK will then be placed into a management software such as ETS to derive a unique *Tool Key (TK)*. This TK will then be sent to the device from ETS and will be secured by encrypting it with the FDSK. The FDSK will no longer be used unless a factory reset is done on the device. Now, the device will only accept management communications from the ETS client with the TK that initialized it and will receive separate *runtime keys* ($RK_i - RK_n$) for each of the its datapoints that it will publish to the network. These keys are programmed onto the device via ETS encrypting it with the TK.

#### *4.2.3. Device Implementation*

KNX devices that support KNX Data Secure now implement an extension to their network stack which is the *Secure Application Layer*. This layer will be responsible for encrypting and decrypting secure messages if confidentiality is desired and authenticating the message if message-authentication is desired.

#### *4.2.4. Security Analysis*

A security analysis is presented regarding the security provided by the KNX Data Secure standard on a pure KNX Data Secure setup. Meaning a network in which only KNX Data Secure devices are utilized and all messages are secured using confidentiality and message-authentication.

*Vulnerabilities Addressed. Brute Force*: The introduction of 128-bit keys used for encryption mitigate brute force attacks due to their lengths as it would be computationally infeasible to guess one of these keys.

*Device Reprogramming*: Because any device on an insecure KNX network could send out management commands after initiating a unicast connection, an attacker could send out subsequent management commands to reprogram and/or reinitialize a device. Because of the introduction of the Tool Key, the devices will only accept communications from ETS who also has that symmetric Tool Key. Assuming that the attacker does not have this key, the device reprogramming vulnerability is mitigated.

*DoS*: As there are no authentication mechanisms to allow a device to communicate within an insecure KNX network, attackers could perform a device reprogramming attack that would render the device and/or network useless [100]. However, requiring a tool key to perform management tasks on a device will mitigate this attack.

*Eavesdropping*: Because an insecure KNX network did not have any encryption on it, an attacker could gain access to the network and passively sniff out traffic. However, with the utilization of keys, encryption of network communications is present; thus an attacker cannot sniff out traffic if they do not have these keys.

*False Data Injection*: Because there was no device-authentication mechanism on an insecure KNX network, an attacker could send out information onto the bus to any group address and have unintended effects on the network. However, there is an inherent form of authentication on a KNX Data Secure network as each device needed to be commissioned via the same ETS client to gather all the runtime keys through the Tool Key derived from their original FDSK key. So, an external attacker cannot inject false data onto the network.

*MITM*: Because there was no encryption, integrity checking, or authentication mechanisms in an insecure KNX network, an attacker could perform the 4 different functions described in Section 3. However, with the requirement that every device can only get encryption keys from a single ETS client as well as having Message Authentication Codes (MACs) generated per message, an attacker cannot fabricate, modify, interrupt nor intercept packets if they are not added to the network via the original ETS client.

*Reconnaissance*: Because there is no enforced authorization or any authentication mechanism to gain access to a base KNX network, any device is able to read any information on a device such as the serial number and application version through management communications. However, with KNX Data Secure, these communications are now protected through the Tool Key for which only the original ETS client and the KNX device have.

*Replay Attack*: Because there is no unique portion of a KNX packet such as a timestamp or sequence number that will prevent replay attacks in the base KNX protocol, attackers were able to replay messages to cause unintended behavior. However, with KNX Data Secure, a sequence counter is introduced that will continuously be incremented throughout each communication. The requirement to accept a message is that the sequence counter of a message needs to be higher than the one that is stored within the receiver. And so, an exact old message cannot be sent as the sequence number will contain the old sequence number; thus the replay attack is mitigated.

*Spoofing*: Because there were no entity authentication mechanisms for messages sent on an insecure KNX network, an attacker could simply craft packets that would impersonate another legitimate device as if it were them and cause varying effects. Now, with the procedure in place to gather the tool and runtime keys, an attacker must be added into the network via the original ETS client that commissioned the network. An attacker will not be able to properly encrypt messages causing its messages to be dropped by the network; thus mitigating the spoofing vulnerability.

*Vulnerabilities Remaining. Physical Attack*: KNX Data Secure states the secure offline storage of keys and other sensitive information is a "local matter" [28]. The resulting implementation of secure offline storage can have security vulnerabilities inherent which could lead to a node being compromised. For example, symmetric keys for all types of communication are stored within the devices and ETS. An attacker can gather these and then bypass the protections provided by KNX Data Secure.

### *4.3. KNX/IP Secure*

KNX/IP Secure [101] is an optional security extension of the KNX Protocol developed by the KNX Association release in 2013 which seeks to secure KNX/IP messages by wrapping them in a KNX/IP Secure wrapper to be sent over an IP backbone communication media. The KNX/IP telegram is going to be encrypted and placed into the data segment of a normal IP telegram. This extension will provide confidentiality, integrity as well as user and message authentication.

#### *4.3.1. Network Model*

The network model is not affected by this security extension as it will place the KNX/IP telegram into a security wrapper which is located in the application layer of a typical IP packet.

#### *4.3.2. Key Scheme*

There are various keys that are going to be used throughout a KNX/IP Secure network for the different forms of communications that will be present.

For unicast communications (i.e. management communications), a fresh session key will be derived each time one of the connections needed to communicate is established. This session key is derived through an Elliptic Curve Diffie Hellman (ECDH) exchange. To add to the security of this ECDH session key generation scheme, passwords will be

introduced that will aid in protecting these management commands from being misused. For management communications, this is mandatory; for runtime communications, it is optional. These passwords will also be required to authenticate an entity looking to configure another device. The length of the passwords is not mandated and is left to those implementing the commissioning device such as ETS who will program these keys onto the device [102]. Another component that will aid in securing this ECDH exchange is the usage of an authentication code that will be used in the ECDH process to establish a session key. This authentication code is typically the FDSK mentioned before and needs to be programmed into the other device by the commissioning device.

For multicast communications, a single shared group key will be set when being added into the network through a management tool. This shared key will be secured through a management connection. It will be used in AES-128-CCM encryption schemes to provide confidentiality, integrity and message-authentication on KNX/IP messages. Key storage is not discussed in the KNX/IP Secure Standard aside from storing the lower 16 bytes of a SHA-256 hash digest in the password hashes table of a device.

#### *4.3.3. Device Implementation*

KNX devices that support KNX/IP Secure will have to implement the KNX/IP Security layer that will be responsible for encrypting and decrypting KNX/IP Secure messages as well as generating MACs for the encapsulated KNX/IP frame.

#### *4.3.4. Security Analysis*

A brief analysis of the security provided by KNX/IP Secure is provided. Because KNX/IP Secure only affects KNX/IP communications, all previous issues for the other communication mediums in Table 2 are still present and can be used as an extra attack vector into a BAS that has multiple communication mediums in use. The security analysis provided is going to discuss vulnerabilities that remain and are addressed for pure KNX/IP Secure networks (i.e, only KNX/IP Secure devices are in use).

*Vulnerabilities Addressed. Brute Force*: The introduction of 128-bit AES encryption keys as well as ECDH keys render brute force attacks computationally infeasible thus mitigating the attack.

*Device Reprogramming*: Because there was no authentication mechanism for management communication sessions in an insecure KNX network, an attacker could establish a session with a device and reprogram it. With the introduction of an authenticated ECDH key exchange, an outside attacker will not be able to establish a management session with devices; thus mitigating the attack.

*DoS*: As there are no authentication mechanisms to allow a device to communicate within an insecure KNX network, attackers could perform a device reprogramming attack that would render the device and/or network useless [100]; causing a DoS. However, with the utilization of the authenticated ECDH key exchange, an attacker will not be able to perform this attack or perform any other attack that would require managerial permissions (i.e. constant restart).

*Eavesdropping*: Because KNX/IP data is encapsulated within a KNX/IP Secure wrapper, the pertinent data such as the application data and KNX addresses are encrypted. And so, the vulnerability of passively listening on the KNX/IP bus for KNX data is mitigated.

*False Data Injection*: As there is no authentication mechanism for entities to send messages onto an insecure KNX network, an attacker could send out false data readings onto the network that would have adverse effects on the BAS [67]. With the introduction of an authenticated ECDH procedure to send out the network key used to send data onto a network, an attacker cannot inject false data without this key.

*MITM*: Because there was no entity authentication, encryption or integrity checks when sending messages onto the network in an insecure KNX/IP network, an attacker could place themselves in between two communicating entities and carry out all four attacks discussed prior in Section 3. With the introduction of entity authentication onto the network through the commissioning process, none of these attacks are feasible as connections with the attacker will not be established as the attacker cannot authenticate themselves to the devices.

*Reconnaissance*: Because there was no entity authentication in an insecure KNX/IP network and no protections on management communications which could allow complete readings of KNX devices, an attacker could gather a lot of information about the insecure KNX network with no repercussions. With the introduction of one-time session keys and a group key for runtime communications, an attacker will need to be authenticated onto the network to gather particular information. This will mitigate advanced reconnaissance attacks.

*Replay Attack*: As there was no mechanism to ensure data freshness in an insecure KNX/IP network, an attacker could transmit old telegrams onto the network and cause unintended effects. However, with KNX/IP Secure, KNX/IP messages are secured through the use of a sequence number; thus, the vulnerability is mitigated.

*Spoofing*: As there was no entity-authentication method in insecure KNX networks, an attacker could send out messages with the addresses of a legitimate devices. In KNX/IP Secure, the utilization of an authentication code will mitigate spoofing attacks.

*Vulnerabilities Remaining. Physical Attacks*: Proper storage of sensitive information such as keys is paramount to mitigating a physical attack that seeks to extract information from a device. Runtime communications are not secure as these use a shared group key that will be used to encrypt the messages. If an attacker gathers these, then they will be able to decrypt all past, present and future runtime communications. Because management session keys are used once per session, an attacker will not be able to decrypt previous communications as those keys no longer exist. However, if an attacker gathers the device's private key and the current session key, they will be able to bypass the protections provided by KNX/IP Secure as the private key is the single defense mechanism that derives the other keys in use; meaning they will be able to decrypt all present and future communications.

### *4.4. LonWorks*

At the time of this survey, there is no official security extension provided for LonWorks networks; thus all the vulnerabilities that were discussed prior in Table 2 are still present on BASs utilizing the LonWorks protocol.

### *4.5. ModBus/TCP Security*

The ModBus/TCP Security [103] protocol is an extension to the base ModBus protocol that will secure the application layer messages inside of a TLS wrapper.

#### *4.5.1. Network Model*

The network model of a ModBus/TCP Security network is not modified from the base ModBus/TCP network as ModBus/TCP is strictly an application layer protocol.

#### *4.5.2. Key Scheme*

In ModBus/TCP Security, the only information that needs to be stored on a Modbus/TCP Secure device is the private key, public key and the corresponding signed device certificate used for TLS communication. The guidelines for the secure storage of the private key, public key and certificate, according to the Modbus/TCP Security standard [103], is left up to the developers of the ModBus/TCP device.

#### *4.5.3. Device Implementation*

ModBus/TCP Security devices will need to be able to encapsulate Modbus/TCP packets inside of a TLS packet and perform the necessary cryptographic procedures to communicate. There are no new modifications to the network stack within the device.

#### *4.5.4. Security Analysis*

A brief analysis of the security provided by the Modbus/TCP Security protocol is provided. Because this protocol applies only for TCP communications, this security protocol will only analyze the defenses provided on pure ModBus/TCP networks. Other communication mediums will not benefit from the security provided by the TLS protocol. These communications may sabotage the security of the BAS as an attacker can gain access to the network through another communication medium and carry out attacks from that insecure network.

*Vulnerabilities Addressed. Device Reprogramming*: Modbus/TCP does not provide any authentication scheme which would allow for rogue devices to perform *data access* functions to overwrite data on the device and reprogram the device. Using mutual authentication with TLS in a Modbus/TCP Security network will prevent rogue devices from sending any management commands to the Modbus devices to reconfigure its application program.

*DoS*: Because there were not any authentication mechanisms for allowing devices onto an insecure Modbus/TCP network, an attacker could send out commands that could render the device unusable. With the introduction of TLS in a Modbus/TCP Security network, an attacker won't be able to send these commands to devices without being authenticated by the target device.

*Eavesdropping*: The lack of encryption on an insecure ModBus/TCP network allowed attackers to passively listen on the network. Using TLS will encrypt the network traffic of the KNX network, mitigating the potential for the attacker to eavesdrop on the network.

*False Data Injection*: The absence of authentication mechanisms to allow writing data into an insecure ModBus/TCP network would allow a rogue device to insert false data into the network. However, with ModBus/TCP Security, TLS and mutual authentication are required; meaning a rogue device cannot perform this.

*MITM*: As there are no encryption, integrity-checking or authentication mechanisms in place for a base ModBus network, if an attacker is able to place themselves in between two communications, the attacker can carry out any of the four attacks mentioned in Section 3. In a Modbus/TCP Security network however, the utilization of TLS which requires mutual authentication to establish connections on the network will prevent an attacker from being able to gain access on the network; thus preventing all goals of the MITM attack.

*Reconnaissance*: The lack of authentication to get into a base ModBus/TCP network allows attackers with rogue devices to scan the network for various pieces of information with no trouble. In a Modbus/TCP Security network, an attacker is required to have a valid certificate because of the use of TLS with mutual authentication; mitigating the

reconnaissance attack potential.

*Replay Attack*: Because there are no sequence numbers or sequence identifiers present in an insecure ModBus/TCP protocol, an attacker could replay the same message twice to cause the same effect the original message had done. Using TLS in ModBus/TCP Security solves this problem because it uses its own sequence numbers that are used between entities. This means that an old message cannot be sent onto the network and thus, the replay attack vulnerability is mitigated.

*Spoofing*: The lack of entity authentication within an insecure Modbus/TCP network allowed for attackers to send messages as another device on the network. By using TLS in ModBus/TCP Security with mutual authentication, a rogue device cannot send messages as another device because they will not have the appropriate means to authenticate themselves into the network.

### *4.6. EnOcean High Security*

EnOcean High Security [104] is a security extension to the base EnOcean protocol that will provide confidentiality, integrity, entity and message authentication mechanisms to secure EnOcean networks by employing its own encryption scheme known as Variable AES (VAES) and using Cipher-based Message Authentication Codes (CMACs).

#### *4.6.1. Network Model*

The network model of an EnOcean High Security Radio Network is not modified as extra devices are not added and the flow of traffic remains the same.

#### *4.6.2. Key Scheme*

In EnOcean High Security, security is provided by using VAES with 128-bit keys for encryption and CMACs for message authentication and integrity. Each communication direction with different partners will require a unique symmetric key and a rolling code (a.k.a. sequence number).

Each device will have its own, unique 128-bit AES key that is programmed onto it during manufacturing. The key will then be printed on a label and placed on the physical device. Once two devices want to communicate with each other, their keys will need to be programmed onto the other device through some manual interface on the device or a secure commissioning channel. Further communications will use these keys to encrypt and decrypt messages.

#### *4.6.3. Device Implementation*

Devices will now need to implement the EnOcean Security layer that will be present at the Presentation layer of the OSI model.

#### *4.6.4. Security Analysis*

A brief analysis of the security provided by the EnOcean High Security protocol is provided.

*Vulnerabilities Addressed. Eavesdropping*: As insecure EnOcean networks did not possess any encryption on the messages sent throughout, an attacker could simply listen and gather data. With the usage of VAES in EnOcean High Security, it will ensure that unauthorized parties will not be able to listen on communications that are sent throughout the network without the correct key.

*False Data Injection*: Because insecure EnOcean networks did not possess any entity authentication into the network aside from the EnOcean Unique Radio Identifier (EURID) contained on a device (which can be easily sniffed out as it's contained within packets), an attacker could send data onto the network and inject false data. With EnOcean High Security, it will require the correct rolling code, registered EURID, and proper encryption keys to send messages on the network; mitigating the vulnerability.

*Replay Attack*: In insecure EnOcean networks, there were no sequence numbers that were implemented leading to the replay attack vulnerability. In EnOcean High Security, a rolling code will be present on both devices and will be used to verify the freshness of the message sent, mitigating any replay attack attempts.

*Spoofing*: As an insecure EnOcean network does not provide any encryption on the network which would leak the EURID of a legitimate device that can be sniffed out, an attacker could impersonate another device using that information. With EnOcean High Security, an attacker would need the AES key, EURID and the Rolling Code to be able to successfully spoof a device. An attacker cannot do this by listening to the signals that are sent throughout the network if the BAS is already set up; thus, the spoofing vulnerability is mitigated.

*Vulnerabilities Remaining. DoS*: The DoS attacks that were discussed before (resource consumption and jamming) are still present. The cryptographic functions that are used in EnOcean High Security are expensive in terms of computation and will require additional power to carry out - more than can be given via energy harvesting. If an attacker can cause a device not on a power line to participate in these cryptographic sessions, they could drain the battery leading to a DoS.

*Physical Attack*: EnOcean High Security does not provide any security against physical attacks. They leave the issue of secure storage up to the manufacturers and acknowledge jamming attacks and battery draining attacks are feasible on these devices [105].

### *4.7. ZigBee Pro*

ZigBee Pro as stated in [106] is an extension to the base ZigBee protocol which will add new devices and constructs that will provide confidentiality, integrity and message-authentication to messages sent on the network.

#### 4.7.1. Network Model

ZigBee Pro provides the option to have either a centralized or distributed secure network architecture. In each of these architectures, the concept of a *Trust Center* is present. The Trust Center is a device that is trusted by the other devices within a network to distribute the various encryption keys used within a secure ZigBee network. It is also responsible for establishing, maintaining and updating the security policies for the network such as new devices need to submit a passcode to join the secure network. In a centralized architecture, there is only a single trust center within the network and inside a distributed network, the routers can handle a majority of functions that a single trust center can do including adding a new device to the network.

To join a secure ZigBee network, the device will need to follow the security policies that are set by the Trust Center which may require a passphrase or secret code set prior to joining the network; anonymous joining is a possibility though not recommended. If the device to join is a router, it will communicate directly with the trust center. If the device to join is an end device, it can interact with a router who will notify the trust center to start the joining procedure. During this process, the keys that are used within the network will be given to the device that is attempting to join the network.

#### 4.7.2. Key Scheme

There are numerous keys that are used throughout the ZigBee Pro network. These are all 128-bit symmetric keys that will be used in their encryption algorithm AES-CCM* for encryption, message authentication and integrity. ZigBee Pro introduces two main types of keys that will have distinct functions within the ZigBee Pro network; *link keys* and a *network key*. The link key is a key that is used for unicast communications between two devices that want to communicate and is not exclusive to the device and trust center. The network key is used to secure runtime multicast or broadcast communications.

Each *link key* is 128-bit AES key that is shared amongst the two communicating entities. The method in which the devices get this key is through an offline installation or through the network using a process known as *key-transport*. This process will protect the desired key when being sent throughout the ZigBee network. There are many types of link keys that are used for separate functions within the network. The first is a *centralized security global trust center key* which is used to join a centralized ZigBee Pro network. The ZigBee alliance denotes the default value 5*A* 69 67 42 65 65 41 6*C* 6*C* 69 61 6*E* 63 65 30 39. Next is a *distributed security global link key* which is the link key used to join a distribute ZigBee Pro network. Another *install code link key* is defined for creating a unique trust center key for joining. To secure the application layer of messages between devices communicating with each other, the *application link key* is created and used. Finally, the *device specific trust center link key* is used between the trust center and the device in the network for any trust center commands being sent. The network key, on the other hand, is of only one type that can be used in both centralized and distributed networks.

Key rotation is similar in the two types of networks with regards to the trust center. The trust center will define how the keys are distributed to devices that want to join the network. The trust center in both types of networks will define the policies that are needed for devices to join the network; this can include requiring devices to provide some type of passcode or the link key in use by the trust center at the time. The key difference that appears between the two network types concerns network key updates. In a centralized architecture, this functionality exists as there is a single trust center that can update the link key and then distribute it to all endpoint devices as they have registered with this single trust center. However, in distributed networks, the link key does not get updated as there is no single trust center that can perform this.

#### 4.7.3. Device Implementation

Devices wanting to communicate using the ZigBee Pro protocol will need to have the Application Support Sub-Layer (APS) that will be handling the security of the messages that are passing between the Application layer and the Network Layer. In this layer, it will utilize the link and network keys that are in the network to perform the AES-CCM* encryption algorithm to provide confidentiality, integrity and message-authentication onto messages on the network. Furthermore, the ZigBee Device Object (ZDO) contained within the device needs to be extended to handle security policies and security configurations of the device.

#### 4.7.4. Security Analysis

A brief analysis of the security provided by ZigBee Pro is provided.

*Vulnerabilities Addressed. Device Reprogramming:* Base ZigBee networks allowed any device to send out any type of unicast management command and could spoof the sender address, the attacker could reprogram any device. However, with the introduction of encryption keys that need to be used to communicate as well as the joining process with the trust center, a rogue attacker cannot send these management commands without being allowed into the network.

*DoS:* Insecure ZigBee networks gave the ability to attackers to send all types of messages including managerial commands such as constant restart or constant wake-up signals to drain the battery for devices that run on batteries. Since encryption keys and message-authentication codes are introduced, an outside attacker can no longer send these

messages without being added into the network through a trust center.

*Eavesdropping:* Because base ZigBee networks did not have any mechanisms for encrypting data, the data could be sniffed out without any issue. With the inclusion of the AES-CCM* encryption algorithm, the data will no longer be in plaintext when sent over the network. While there is a chance that an attacker could use the default centralized security global trust center key if the network utilizes this key, we consider eavesdropping fully mitigated as the mechanisms can prevent it if implemented correctly.

*False Data Injection:* As base ZigBee networks did not have any mechanism to send valid encrypted data onto the network, an attacker could freely send messages onto the network and carry out their goal. With the introduction of various keys and needing to be added into the network, an attacker cannot send messages without these pieces; thus mitigating the vulnerability from an outside attacker.

*MITM:* Base ZigBee networks did not have any protections for messages being sent across the network. This made it possible for attackers to listen in on, modify or fabricate messages sent on the network. With the introduction of encryption keys that will include message-authentication codes that include a hash as well as the trust center's joining process, an external attacker would not be able to do this.

*Reconnaissance:* As an attacker did not need to be authenticated into a base ZigBee network, they could perform reconnaissance attacks by querying the devices for information using *beacon requests* and *scan services*. With the introduction of a trust center that will handle the initial join procedure for new devices into the network and get the corresponding link and network keys, an outside attacker cannot send messages through this method; thus the reconnaissance attack is mitigated.

*Replay Attack:* As base ZigBee networks did not have any mechanism such as a sequence counter, an attacker could send old, captured messages and the other devices would accept it as is which could lead to unintended behavior. With the introduction of a sequence counter in ZigBee Pro, the sequence number is included in all messages and will be used to derive the message authentication codes resulting in a mitigation of the attack.

*Spoofing:* Base ZigBee networks did not have any entity-authentication in the network. Meaning, an attacker could send out any message and impersonate any other device on the network. However, with the introduction of the trust center that will handle the initial distribution of the network keys for devices that want to join the network, an attacker cannot send information without the correct keys and having gone through the trust center - resulting in the mitigation of external spoofing attacks.

*Vulnerabilities Remaining. Physical Attack:* Securing the various keys that are present within the network is important to mitigating physical firmware extractions or memory extractions on devices. ZigBee Pro acknowledges these attacks are possible, but caution the developer to minimize the risk of this if offline security isn't implemented in the device. However, a solution for this is not provided.

### *4.8. Z-Wave Plus*

The Z-Wave Plus [107] protocol is an extension to the base Z-Wave protocol which contains two optional security classes known as S0 and S2 that will provide security on a Z-Wave network. These classes aim to provide confidentiality, integrity, user and message authentication in a Z-Wave network while providing legacy support.

#### *4.8.1. Network Model*

As the security extension is applied at application layer, there is no modification to the traditional Z-Wave network architecture described earlier; it will remain a centralized architecture that is typically controlled by a single primary controller.

#### *4.8.2. Key Scheme*

There are two different *security classes* introduced in Z-Wave Plus which aim to provide security for both legacy and modern devices - S0 and S2.

*S0 Security Class.* The S0 security class aims to provide security for legacy devices that cannot support the security requirements of the *S2 Security Class* described shortly. In a pure S0 Security Class network, there will be a shared network key that is 16-bytes long. During the inclusion phase when network keys are exchanged, a *temporary key* will be used to encrypt the in-transit network key. Z-Wave Plus will utilize the AES-128 CCM algorithm and will use nonces that are exchanged with one another prior to encryption.

*S2 Security Class.* The S2 Security Class was designed for newer devices that have sufficient resources to carry out expensive cryptographic algorithms to provide greater security in a pure S2 Z-Wave Plus network. A pure S2 Z-Wave Plus network uses the ECDH key exchange protocol to establish the network keys that will be used to carry out encryption on future network communications using the AES-128-CMAC encryption algorithm.

#### *4.8.3. Device Implementation*

Devices that wish to participate in a secure Z-Wave Plus network utilizing S0 or S2 security classes must be able to support the cryptographic functions required to communicate with other S0 or S2 devices.

#### 4.8.4. Security Analysis

A brief analysis of the security provided by the Z-Wave Plus protocol and the security classes S0 and S2 is provided. Unless otherwise specified, the following analyses apply to both the S2 and S0 security classes as S2 is designed to provide backwards compatibility to S0 devices.

*Vulnerabilities Addressed. Device Reprogramming*: The lack of authentication or integrity checks in insecure Z-Wave networks led to attacks on the device's application program via Over-the-Air (OTA) updates and firmware modifications in [48] and [49]. However, the introduction of the authenticated inclusion process to authenticate communications and the use of CMACs to provide integrity checks in a Z-Wave Plus network with S0 or S2 devices will secure attacks from outside entities or unpaired entities from modifying the device.

*Eavesdropping*: The absence of encryption on the data that is sent throughout an insecure Z-Wave network led to outside devices being able to listen on the network and gather information from the packets. The introduction of encryption keys for the application data in a Z-Wave Plus network with S0 or S2 devices prevents potentially-sensitive application data from being listened in on.

*False Data Injection*: The lack of both device and message authentication in an insecure Z-Wave network allowed attackers to inject false data into the network by spoofing device addresses which are inherently trusted in the network. By requiring devices to be properly authenticated into the network during the inclusion process to gather network keys within a Z-Wave Plus network with S0 or S2 devices, a rogue device cannot inject data without getting onto the network.

*MITM*: Because an attacker could simply send out network management commands to devices on an insecure Z-Wave network without having to be included into the network, or encrypt messages, then it was possible to poison the routes that devices used when sending their messages with. However, with the the inclusion processes in a Z-Wave Plus Network as well as encryption with both S0 and S2 security classes, rogue devices cannot poison routes from outside the network.

*Reconnaissance*: Within an insecure Z-Wave network, an attacker can simply query each device using *Get-NL* messages, or other management frames to build a network map, and determine the services or functions of each device. Within a Z-Wave Plus network which contains S0 or S2 devices, this is not possible without the network key shared between devices included into the network.

*Replay Attack*: Replay attacks were present in base Z-Wave networks because there were no sequence numbers or nonces in use. However, in Z-Wave Plus S0 and S2 networks, nonces are required; thus mitigating replay attack vulnerabilities.

*Spoofing*: As there was no authentication in an insecure Z-Wave network, a rogue device was able to send messages with the source address modified to match a valid device included into the network. As devices implicitly trusted the source addresses of messages, rather then verifying it's authenticity through some explicit mechanism, the devices would accept these messages. Now, with the pairing processes and encryption keys in a secure Z-Wave network with S0 or S2 devices, a rogue device can't send messages without having knowledge of the key and other information used during the pairing process.

*Vulnerabilities Remaining. DoS*: Within a Z-Wave network it is still possible to preform a DoS attack on devices using the S0 and S2 security classes. This has been shown in [66, 49] as the authors were able to leverage the Nonce-Get command classes used by devices to get a Nonce for secure communications and the Transport command classes for message fragmentation to preform DoS attacks. These Nonce-Get messages are unencrypted, and thus an attacker external to the Z-Wave network may spoof these messages to perform a DoS attack by flooding the target device with these messages.

*Physical Attack*: Z-Wave Plus does not provide any guidelines regarding secure offline storage of any sensitive information or application program that is on the devices.

*Side Channel*: A side channel attack has already been carried out against S2 Z-Wave Plus networks by looking at the packet length of encrypted Z-Wave Plus S2 network messages [95]; implying that this can be done on the S0 network with some modifications.

### 4.9. Analysis of Common Security Schemes

With the various analyses done on these secure protocols, we provide a brief summary of the common security schemes used by the protocols and discuss which attacks are mitigated by each.

#### 4.9.1. End-to-End Encryption

End-to-end encryption states that for any communication from sender to recipient, the message being sent is encrypted at the sender's side and once it's encrypted, only the recipient is able to decrypt this message; not even the sender can decrypt it. BACnet/SC and ModBus/TCP Security are the only secure BAS protocols that utilize end-to-encryption at some stage during the BAS's lifecycle through the use of TLS. In TLS, the initial key distribution for the symmetric key will be end-to-end encrypted as this symmetric key will be encrypted with the recipient's public key; implying the recipient is the only entity that can decrypt it as it is the sole owner of the corresponding private key. This scheme will provide security against external attackers from listening in on the communications and

extracting the keys used for later communications. However, later communications using the shared symmetric key will then not be end-to-end encrypted as there is more than one entity which can decrypt the message.

#### *4.9.2. TLS*

Using TLS v1.3 secures BACnet/SC and ModBus/TCP Security BAS installations from all common networking, firmware/software and data attacks referenced in Table 2 as it provides encryption, integrity-checking, and authentication mechanisms.

Specifically, using an asymmetric key encryption scheme such as ECDH or RSA to asymmetrically encrypt a symmetric key such as an AES-128 or AES-256 key will solve the initial key distribution scheme problem that comes with symmetric key usage; ensuring the keys used during communications are secured. At minimum, the usage of longer encryption keys will prevent brute-force attacks against these keys. It can also protect the network against common network-based attacks as the attacker will not be able to correctly format packets in the network. Finally, the use of encryption would be able to secure against the eavesdropping attack against the data of a network.

Integrity checks will be present through the derivation of hash digests per message within each original communication to ensure it has not been tampered with (i.e. MITM modification).

Using mutual authentication through valid digital certificates and corresponding private keys will not allow an attacker to perform any active (i.e. sending out traffic in a reconnaissance attack) network-based or firmware/software attacks as they will not be allowed access to the network or device.

#### *4.9.3. Message Authentication Codes*

The usage of MACs will primarily provide defenses against the MITM attack as it will ensure that the message hasn't been tampered with through the use of a hash digest of the original message discussed previously. This is a crucial component in protecting against modifications done by a MITM attack. However, MACs also typically involve the use of a shared key so the communicating parties can decrypt the MAC and confirm the sender has the correct key. With this, it also provides protections against false data injection, MITM, reconnaissance, and certain device reprogramming attacks.

#### *4.9.4. Shared Network Key*

KNX Data Secure, KNX/IP Secure, ZigBee Pro and Z-Wave Plus all share a network key that is shared among the devices during normal non-managerial communications. Assuming the initial distribution of this key is secure, this will provide protections against external attackers that aim to perform false data injection, MITM, reconnaissance, and certain device reprogramming attacks that focus on using existing protocol capabilities to reprogram a device, and eavesdropping attacks as an attacker needs to have this key to perform any of the attacks on the network.

#### *4.9.5. Authenticated Inclusion Schemes*

Having an authentication scheme to join a network and be given corresponding encryption keys will provide protections against some device reprogramming attacks that focus on abusing protocol capabilities to reprogram the functionality of a device as well as stop a majority of attacks against the network of the BAS such as false data injection, MITM, reconnaissance and spoofing attacks. Most protocols will have some form of authentication for a device to access the network; i.e. ETS registration with KNX Data Secure, a trusted entity in ZigBee Pro or Device Specific Keys for EnOcean and Z-Wave S2.

#### *4.9.6. Lengthy Encryption Keys*

BACnet/SC, KNX Data Secure, KNX/IP Secure, Modbus/TCP Security, EnOcean High Security, ZigBee Pro and Z-Wave S0/S2 all use encryption keys that are of a secure length. For example, for an AES-based encryption scheme, an encryption key should be at least 128-bits. These lengthy keys mitigate brute-force attacks as it will be computationally infeasible for an attacker to guess the necessary keys.

#### *4.9.7. Perfect Forward Secrecy Schemes*

KNX/IP Secure management communications, BACnet/SC, ModBus/TCP Security, and Z-Wave S2 networks all achieve Perfect Forward Secrecy for their communications. TLS, KNX/IP Secure management communications and Z-Wave S2 network key distribution communications all will generate a new session key during a communication and not use it again. This prevents replay attacks and eavesdropping attacks.

## 5. Open problems

In light of this survey regarding the current state-of-the-art for BAS security including the secure extensions that have come out, we note the six factors of BAS security and key open issues that are aimed at those in academia and industry.

### *5.1. Six Factors of BAS Security*

Fig. 8 gives the six factors of BAS security. In addition to the five technical factors including physical components, operating system (OS), software, networking and data, we should also consider the human factor since humans are often the weakest link of a secure system. For example, we shall improve the security awareness of people involved in a BAS. We find the human factor of BAS security is rarely discussed in the literature.

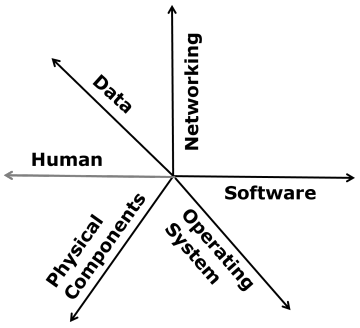


Figure 8: Six Factors for BAS Security

### *5.2. Network Security*

#### *5.2.1. Lack of Security for non-IP BAS Networks*

The adoption of TLS to secure IP-based BAS communications is justified as TLS is widely used within everyday communications on the internet. However, this method is only applied for BASs with IP communications. Existing BASs typically have TP, PL or RF components that will not be secured. This leads to major vulnerabilities in existing systems as these networks can be used as entrypoints for attackers.

#### *5.2.2. Limited Covert and Side Channel Attacks*

There is limited work done on covert and side channel attacks on the base protocols; even more so on the secure BAS protocols. Additional research will need to be carried out to understand how the different protocols withstand these attacks.

### *5.3. Software/OS Security: Unexplored Fuzzing Potential*

Software security is not a well-known topic by those developing the software programs that are present on these devices. This can lead to unintended behavior or even a complete Denial-of-Service depending on the extent of the attack done on the software. Fuzzing seeks to highlight these issues. There is little work done on fuzzing within insecure BAS networks and no work done on secure BAS networks.

### *5.4. Hardware/OS Security: Lack of Secure Boot Mechanisms*

Physical attacks and non-network-based device reprogramming attacks rely on the improper security of the firmware (which refers to the code inside of a device, including the OS application code and user written code) and software present on the devices. Secure boot will ensure that the firmware and software on a device is the original program and from a trusted source. Those who create these BAS devices shall adopt this practice to mitigate numerous physical attacks while also securing against non-network-based device reprogramming attacks.

### *5.5. Data Security: Lack of Flash Memory Encryption*

The secure offline storage of sensitive information such as encryption keys used throughout communications is nearly absent in both the insecure and secure versions of the BAS protocols. If it is referenced, it is discussed as a "local matter". In the case of [8], the security personnel responding to the complete DoS of a KNX network were able to search through the device's memory and locate the access level key the attacker used to lockdown all KNX devices as it was stored in plaintext. An advanced attacker could do the same thing and use that to carry out this attack again. However, if the memory of the device were to have been encrypted, no one would have been able to carry out this attack. This concept of memory encryption applies to all other keys and sensitive information used within devices in a BAS network.

## 6. Case Study

In this section, we first introduce a generalized and simplified architecture of a real-world BAS and then discuss the vulnerabilities of the BAS.

Fig. 9 provides a simplified overview of a real-world BAS used in a university which consists of two buildings; Building A (top) and Building B (bottom). The backbone network used between and within the buildings is a BACnet/IP network connected via Ethernet cables. Within each building, the BACnet/IP backbone network contains a local management device which could be a mobile computer with a serial port for local diagnostics and configurations of devices themselves. It also contains a BACnet Broadcast Management Device (BBMD) which is used to forward broadcast messages between the BACnet/IP networks of the two buildings. The BACnet router is present only in Building A's BACnet/IP network to facilitate communications between sub-networks local to the BACnet/IP network as shown by the white boxes in Fig. 9.

### *6.1. Example Real-world BAS*

In this BAS, the BACnet controllers are used to process data, and facilitate communications between the different physical mediums; they are present at both the BACnet/IP and BACnet MS/TP levels in both buildings. The Room Automation Controllers located in the BACnet MS/TP subnetwork of both buildings are used to facilitate communications between the BACnet MS/TP devices and also allow communication to the devices in the KNX sub-network via a KNX connector onboard the controller. Both the BACnet MS/TP and KNX sub-networks can contain sensors and other end devices.

This BACnet/IP network is segmented using VLANs that are implemented with managed switches and firewalls to provide additional security and isolation between the

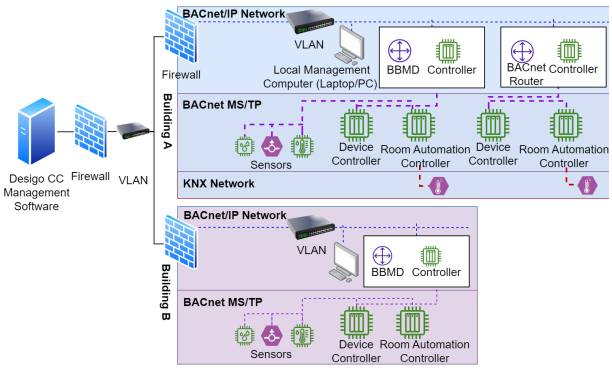


Figure 9: Simplified Real-world BAS Network

buildings and control systems. Desigo CC [26] is the Building Management System (BMS) used in this BAS.

Building A contains a backbone BACnet/IP network connected to a BACnet MS/TP sub-network through a controller in which the various MS/TP devices are interconnected with RS-485 cables. This MS/TP sub-network includes another controller which provides access to an additional KNX sub-network whose devices are interconnected with TP wires plugged into the various devices' KNX Red-Black block connectors. Building B only contains the backbone BACnet/IP network and a BACnet MS/TP sub-network using the same appropriate wires.

Desigo CC [26] Siemens is the Building Management System (BMS) software used in this architecture. The BMS allows network administrators to configure and monitor the BAS. The BMS is external to the buildings it manages and is protected through the use of VLANs and firewalls as shown in Fig. 9.

### 6.2. *Vulnerabilities*

We briefly discuss three major vulnerability types in the real-world BAS in Fig. 9 which uses insecure BAS protocols and may be subject to other attacks reviewed in Section 3.

#### 6.2.1. *Physical Vulnerabilities*

A BAS consists of numerous physical devices spread throughout one or more buildings such as temperature sensors and presence detectors in public areas which may be tampered with. The simplest attack is a DoS against that device through physical destruction. However, an alternative approach is an attacker can enter one of these buildings and use these physically accessible devices to gain access to the internal BAS network. For example, in any of the networks, an attacker can disconnect the existing device and connect a malicious device onto the network using the appropriate hardware for the network. Similarly, in the case of KNX, the attacker can simply plug in their device into one of the open ports in the KNX red-black connector if there is a slot available. The cables and connectors for the different communication mediums are typically inexpensive and readily available. Once they connect onto the network, they are able to control the network and perform any attacks they desire. We demonstrated this in [67], where we plug Raspberry Pis between a temperature sensor and a BAS controller, and deploy MITM attacks to inject false sensor data, affect readings within DesigoCC, and cause energy loss.

Another potential attack vector is through the use of programming mode for KNX devices. These devices have a specific button to place them into programming mode which allows new application programs and addresses to be written to them. An attacker can press this button and reprogram the device - potentially causing a DoS or causing unintended effects.

#### 6.2.2. *Software Vulnerabilities*

There are a number of common software vulnerabilities in BAS devices. Some KNX devices utilize HTTP servers to allow for configuration which has well-documented vulnerabilities. In general, a lack of security awareness and skill may have caused the issues. For example, we have reported software vulnerabilities to the KNX Association for their ETS software [96] and the developer of the Calimero suite [108]. Although the bugs are fixed, no CVEs were generated to notify the public of the danger.

#### 6.2.3. *Protocol Vulnerabilities*

In this simplified BAS network, the protocols BACnet/IP, BACnet MS/TP and KNX are all used. These protocols are the insecure versions of the protocols; meaning the attacks and vulnerabilities discussed in 3 are all feasible within this BAS.

For example, in KNX, there is an optional security mechanism known as *access level*s which are used to authorize access to services such as datapoints and memory locations within a device and are protected via a key of 8 hex characters. These access levels range from 0 - 3 or 0 - 15 dependent on the device capabilities. Access level 0 is given full access to the device. Intuitively, each level will inherit the access given to the lower levels (i.e. access level 10 will inherit the access given to levels 11 - 15); making access level 0 the most desirable.

Because the KNX traffic is not encrypted in this BAS, these keys are subject to eavesdropping. Once this happens, then an attacker can perform a few attacks using this. Assuming that the attacker got the key for access level 0, an attacker can fully reprogram the devices with a different application program or addresses used for communication and cause a DoS. After this, the attacker can change these keys used for the access levels and lock the devices, rendering them unusable.

### 6.3. Defense Recommendations

There are a few recommendations that we can provide to significantly enhance the security of this BAS installation.

*Secure Protocol Adoption*: The most intuitive and effective recommendation is to use the secure protocols BACnet/SC and KNX Data Secure for the BACnet/IP and KNX network. In this way, nearly all the protocol vulnerabilities discussed will be mitigated. However, it is important to reiterate BACnet/SC only applies to the BACnet/IP networks within the BAS; meaning the BACnet MS/TP network continues to remain a major vulnerability in the whole system as there is no secure protocol for BACnet MS/TP networks.

*Physically Secure Devices*: As BACnet/SC and KNX Data Secure do not provide protections against physical attacks, a critical recommendation that is proposed is to ensure the BAS devices are placed into tamper-proof lockboxes, cabling is not exposed, and critical infrastructure such as controllers and routers are placed in secured utility closets or rooms via physical access control mechanisms such as door locks and RFID readers. This is particularly true for BACnet MS/TP as that subnetwork is entirely vulnerable and an attacker can gain access through a single wire that is exposed.

*IT-Based Protections*: IT security procedures should be followed in terms of network segmentation and isolation for BAS networks. Firewalls, routers and all the software on every device including BAS devices should be regularly patched for any security vulnerabilities that may arise. Firewalls and routers should have appropriate rules that limit access to critical systems.

If these recommendations are to be enforced, the BAS will have an improved security stance against attackers. However, it must be realized that an effort to completely reconfigure a network to utilize these new protocols will require devices to be upgraded and a lot of manual effort will be required.

## 7. Conclusion

In this paper, we perform a comprehensive survey of BAS protocol network architectures and attacks against BASs and confirm they are vulnerable. These BAS protocols are widely utilized in modern buildings such as those in businesses, university campuses, apartments and houses, therefore they are susceptible to wide range of attacks. If a BAS is connected to the Internet or components of the BAS are physically accessible, severe consequences may follow. We also give a holistic study of secure extensions of various BAS protocols and analyze their capabilities against the reviewed attacks. A case study of a real-world BAS shows insecure BAS protocols and devices are often used in buildings. We then discuss open problems to promote future research as we seek to raise the awareness of the security issues of BASs to both academia and industry.


## Acknowledgment

This research was supported in part by US National Science Foundation (NSF) awards 2325451, 1931871 and 1915780, and US Department of Energy (DOE) Award DE-EE0009152. Any opinions, findings, conclusions, and recommendations in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.


## References


[1] I. B. de Brito and R. T. de Sousa Jr, "Development of an open-source testbed based on the modbus protocol for cybersecurity analysis of nuclear power plants," *Applied Sciences*, vol. 12, no. 15, p. 7942, 2022.

[2] N. Kraus, M. Viertel, and O. Burgert, "Control of knx devices over ieee 11073 service-oriented device connectivity," in *2020 IEEE Conference on Industrial Cyberphysical Systems (ICPS)*, vol. 1. IEEE, 2020.

[3] P. Amaro, R. Cortesão, J. Landeck, and P. Santos, "Implementing an advanced meter reading infrastructure using a z-wave compliant wireless sensor network," in *Proceedings of the 2011 3rd international youth conference on energetics (IYCE)*. IEEE, 2011.

[4] I.-V. Sita and P. Dobra, "Knx building automations interaction with city resources management system," *Procedia Technology*, vol. 12, 2014.

[5] D.-F. Pang, S.-L. Lu, and Q.-Y. Zhu, "Design of intelligent home control system based on knx/eib bus network," in *2014 International Conference on Wireless Communication and Sensor Network*. IEEE, 2014.

[6] B. International, "Research study indicates bacnet global market share over 60%," https://bacnet.org/press-releases/research-study-indicates-bacnet-global-market-share-over-60/, 2023.

[7] L. Mathews, "Hackers use ddos attack to cut heat to apartments," "https://www.forbes.com/sites/leemathews/2016/11/07/ddos-attack-leaves-finnish-apartments-without-heat/?sh=7cfc8dfa1a09", 2021.

[8] L. Security, "Knxlock - an attack campaign against knx-based building automation systems," "https://limessecurity.com/en/knxlock/", 2021.

[9] K. J. Higgins, "Lights out: Cyberattacks shut down building automation systems," "https://www.darkreading.com/attacks-breaches/lights-out-cyberattacks-shut-down-building-automation-systems", 2021.

[10] T. Seals, "China-backed apt pwns building automation systems with proxylogon," "https://www.darkreading.com/attacks-breaches/china-backed-apt-pwns-building-automation-proxylogon", 2021.

[11] W. Granzer, W. Kastner, G. Neugschwandtner, and F. Praus, "Security in networked building automation systems," in *2006 IEEE International Workshop on Factory Communication Systems*. IEEE, 2006.

[12] A. Antonini, A. Barenghi, G. Pelosi, and S. Zonouz, "Security challenges in building automation and scada," in *2014 International Carnahan Conference on Security Technology (ICCST)*. IEEE, 2014.

[13] M. A. B. Karnain and Z. B. Zakaria, "A review on zigbee security enhancement in smart home environment," in *2015 2nd International Conference on Information Science and Security (ICISS)*. IEEE, 2015, pp. 1–4.

[14] T. Mundt and P. Wickboldt, "Security in building automation systems-a first analysis," in *2016 International Conference On Cyber Security And Protection Of Digital Services (Cyber Security)*. IEEE, 2016.

[15] A. Humayed, J. Lin, F. Li, and B. Luo, "Cyber-physical systems securityâĂŤa survey," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1802–1831, 2017.

[16] R. Krejčí, O. Hujňák, and M. Švepeš, "Security survey of the iot wireless protocols," in *2017 25th Telecommunication Forum (TELFOR)*. IEEE, 2017.

[17] D. Celebucki, M. A. Lin, and S. Graham, "A security evaluation of popular internet of things protocols for manufacturers," in *2018 IEEE International Conference on Consumer Electronics (ICCE)*. IEEE, 2018.

[18] D. Mocrii, Y. Chen, and P. Musilek, "Iot-based smart homes: A review of system architecture, software, communications, privacy and security," *Internet of Things*, vol. 1, 2018.

[19] P. Ciholas, A. Lennie, P. Sadigova, and J. M. Such, "The security of smart buildings: a systematic literature review," *arXiv preprint arXiv:1901.05837*, 2019.

[20] L. Li, P. Podder, and E. Hoque, "A formal security analysis of zigbee (1.0 and 3.0)," in *Proceedings of the 7th Symposium on Hot Topics in the Science of Security*, 2020, pp. 1–11.

[21] E. Lee, Y.-D. Seo, S.-R. Oh, and Y.-G. Kim, "A survey on standards for interoperability and security in the internet of things," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 1020–1047, 2021.

[22] V. Graveto, T. Cruz, and P. Simões, "Security of building automation and control systems: Survey and future research directions," *Computers & Security*, vol. 112, p. 102527, 2022.

[23] L. P. Rondon, L. Babun, A. Aris, K. Akkaya, and A. S. Uluagac, "Survey on enterprise internet-of-things systems (e-iot): A security perspective," *Ad Hoc Networks*, vol. 125, p. 102728, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1570870521002171

[24] H. M. Rouzbahani, H. Karimipour, E. Fraser, A. Dehghantanha, E. Duncan, A. Green, and C. Russell, "Communication layer security in smart farming: A survey on wireless technologies," *arXiv preprint arXiv:2203.06013*, 2022.

[25] G. Li, L. Ren, Y. Fu, Z. Yang, V. Adetola, J. Wen, Q. Zhu, T. Wu, K. S. Candan, and Z. O'Neill, "A critical review of cyber-physical security for building automation systems," *Annual Reviews in Control*, 2023.

[26] Siemens, "Desigo cc - better. of course," https://www.siemens.com/global/en/products/buildings/automation/desigo/building-management/desigo-cc.html, 2024.

[27] ASHRAE Special Publications, "Ansi/ashrae standard 135-2020," ASHRAE, Tech. Rep., 2020.

[28] *KNX Standard Specifications*, 2nd ed., KNX, 2013.

[29] *Introduction to the LonWorks System*, 1st ed., Echelon Corporation, 1999.

[30] *MODBUS APPLICATION PROTOCOL SPECIFICATION*, v1.1b3 ed., Modbus Organization, 2012.

[31] Z. Alliance, *ZigBee Specification*, 05th ed., ZigBee, 2015. [Online]. Available: https://zigbeealliance.org/wp-content/uploads/2019/11/docs-05-3474-21-0csg-zigbee-specification.pdf

[32] *Z-Wave Device Class Specification*, Z-Wave Alliance, Beaverton, United States, 3 2021.

[33] E. S. IoT, "Radio technology," https://www.enocean.com/en/technology/radio-technology/, 2021.

[34] M. Zeng, "A review of smart buildings protocol and systems with a consideration of security and energy awareness," in *13th International Green and Sustainable Computing Conference (IGSC)*. IEEE, 2022.

[35] T. Yimer, E. Smith, P. Harvey, M. Tienteu, and K. Kornegay, "Error correction attacks on bacnet ms/tp," in *2022 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, 2022, pp. 77–80.

[36] D. G. Holmberg, Nov 2003. [Online]. Available: https://search.proquest.com/openview/11de4f48550d34fec12de2d8aab2a273/1?pq-origsite=gscholar&cbl=41118

[37] D. Schneider and W. Przybilla, "Ernw newsletter 49/august 2015," https://ernw.de/download/ERNW_Newsletter_49_SecurityOfHomeAutomationSystems_signed.pdf, 2015.

[38] C. Schwaiger and A. Treytl, "Smart card based security for fieldbus systems," in *EFTA 2003. 2003 IEEE Conference on Emerging Technologies and Factory Automation. Proceedings (Cat. No.03TH8696)*, vol. 1, 2003, pp. 398–406 vol.1.

[39] D.-G. Akestoridis and P. Tague, "Hiveguard: A network security monitoring architecture for zigbee networks," in *2021 IEEE Conference on Communications and Network Security (CNS)*. IEEE, 2021.

[40] S. Khanji, F. Iqbal, and P. Hung, "Zigbee security vulnerabilities: Exploration and evaluating," in *2019 10th international conference on information and communication systems (ICICS)*. IEEE, 2019.

[41] O. Olawumi, K. Haataja, M. Asikainen, N. Vidgren, and P. Toivanen, "Three practical attacks against zigbee security: Attack scenario definitions, practical experiments, countermeasures, and lessons learned," in *2014 14th International Conference on Hybrid Intelligent Systems*. IEEE, 2014, pp. 199–206.

[42] X. Cao, D. M. Shila, Y. Cheng, Z. Yang, Y. Zhou, and J. Chen, "Ghost-in-zigbee: Energy depletion attack on zigbee-based wireless networks," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 816–829, 2016.

[43] J. Ďurech and M. Franeková, "Security attacks to zigbee technology and their practical realization," in *2014 IEEE 12th international symposium on applied machine intelligence and informatics (SAMI)*. IEEE, 2014, pp. 345–349.

[44] P. Radmand, M. Domingo, J. Singh, J. Arnedo, A. Talevski, S. Petersen, and S. Carlsen, "Zigbee/zigbee pro security assessment based on compromised cryptographic keys," in *2010 International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*. IEEE, 2010, pp. 465–470.

[45] W. Razouk, G. V. Crosby, and A. Sekkaki, "New security approach for zigbee weaknesses," *Procedia Computer Science*, vol. 37, pp. 376–381, 2014.

[46] G. Kambourakis, C. Kolias, D. Geneiatakis, G. Karopoulos, G. M. Makrakis, and I. Kounelis, "A state-of-the-art review on the security of mainstream iot wireless pan protocol stacks," *Symmetry*, vol. 12, no. 4, p. 579, 2020.

[47] C. W. Badenhop, S. R. Graham, B. E. Mullins, and L. O. Mailloux, "Looking under the hood of z-wave: Volatile memory introspection for the zw0301 transceiver," *ACM Transactions on Cyber-Physical Systems*, vol. 3, no. 2, 2018.

[48] C. W. Badenhop, B. W. Ramsey, B. E. Mullins, and L. O. Mailloux, "Extraction and analysis of non-volatile memory of the zw0301 module, a z-wave transceiver," *Digital Investigation*, vol. 17, 2016.

[49] K. Kim, K. Cho, J. Lim, Y. H. Jung, M. S. Sung, S. B. Kim, and H. K. Kim, "WhatâĂŹs your protocol: Vulnerabilities and security threats related to z-wave protocol," *Pervasive and Mobile Computing*, vol. 66, p. 101211, 2020.

[50] D. G. Holmberg, Jul 2003. [Online]. Available: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=916765

[51] M. Bashendy, S. Eltanbouly, A. Tantawy, and A. Erradi, "Design and implementation of cyber-physical attacks on modbus/tcp protocol," in *World Congress on Industrial Control Systems Security (WCICSS)*, 2020.

[52] P. Morgner, S. Mattejat, Z. Benenson, C. Müller, and F. Armknecht, "Insecure to the touch: attacking zigbee 3.0 via touchlink commissioning," in *Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2017, pp. 230–240.

[53] C. Vacherot, "Sneak into buildings with knxnet/ip," in *Sneak into buildings with KNXnet/IP*, 2020.

[54] X. Wang and S. Hao, "Don't kick over the beehive: Attacks and security analysis on zigbee," in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, 2022, pp. 2857–2870.

[55] C. K. Nkuba, S. Kim, S. Dietrich, and H. Lee, "Riding the iot wave with vfuzz: discovering security flaws in smart homes," *IEEE Access*, vol. 10, 2021.

[56] J. L. Hall, "A practical wireless exploitation framework for z wave networks," AIR FORCE INSTITUTE OF TECHNOLOGY WRIGHT-PATTERSON AFB OH WRIGHT-PATTERSON âĂę, Tech. Rep., 2016.

[57] J. Kaur, J. Tonejc, S. Wendzel, and M. Meier, "Securing bacnetâĂŹs pitfalls," in *ICT Systems Security and Privacy Protection: 30th IFIP TC 11 International Conference, SEC 2015, Hamburg, Germany, May 26-28, 2015, Proceedings 30*. Springer, 2015, pp. 616–629.

[58] K. Lamshöft and J. Dittmann, "Assessment of hidden channel attacks: Targetting modbus/tcp," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 11 100–11 107, 2020.

[59] M. Qianqian and B. Kejin, "Security analysis for wireless networks based on zigbee," in *2009 International Forum on Information Technology and Applications*, vol. 1. IEEE, 2009, pp. 158–160.

[60] M. Ibrahim and I. Nabulsi, "Security analysis of smart home systems applying attack graph," in *2021 Fifth World Conference on Smart Trends in Systems Security and Sustainability (WorldS4)*. IEEE, 2021.

[61] W. Granzer, F. Praus, and W. Kastner, "Security in building automation systems," *Industrial Electronics, IEEE Transactions on*, vol. 57, pp. 3622 – 3630, 12 2010.

[62] X. Fan, F. Susan, W. Long, and S. Li, "Security analysis of zigbee," *MWR InfoSecurity*, vol. 2017, pp. 1–18, 2017.

[63] B. Yang, "Study on security of wireless sensor network based on zigbee standard," in *2009 international conference on computational intelligence and security*, vol. 2. IEEE, 2009, pp. 426–430.

[64] D.-G. Akestoridis, M. Harishankar, M. Weber, and P. Tague, "Zigator: Analyzing the security of zigbee-enabled smart homes," in *Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2020, pp. 77–88.

[65] N. Boucif, F. Golchert, A. Siemer, P. Felke, and F. Gosewehr, "Crushing the wave–new z-wave vulnerabilities exposed," *arXiv preprint arXiv:2001.08497*, 2020.

[66] D. Cheng, P. Felke, F. Gosewehr, and Y. Peng, "S0-no-more: A z-wave nonceget denial of service attack utilizing included but offline nodeids," *arXiv preprint arXiv:2205.00781*, 2022.

[67] M. Cash, C. Morales-Gonzalez, S. Wang, X. Jin, A. Parlato, J. Zhu, Q. Z. Sun, and X. Fu, "On false data injection attack against building automation systems," in *2023 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2023, pp. 35–41.

[68] A. Camek, F. Hölzl, and D. Bytschkow, "Providing security to a smart grid prosumer system based on a service oriented architecture in an office environment," in *2013 IEEE PES Innovative Smart Grid Technologies Conference (ISGT)*. IEEE, 2013, pp. 1–7.

[69] J. Molina, "Learn how to control every room at a luxury hotel remotely: The dangers of insecure home automation deployment," *Black Hat USA*, 2014.

[70] L. Rajesh and P. Satyanarayana, "Detection and blocking of replay, false command, and false access injection commands in scada systems with modbus protocol," *Security and Communication Networks*, vol. 2021, 2021.

[71] W. Alsabbagh, S. Amogbonjaye, D. Urrego, and P. Langendörfer, "A stealthy false command injection attack on modbus based scada systems," in *2023 IEEE 20th Consumer Communications & Networking Conference (CCNC)*. IEEE, 2023, pp. 1–9.

[72] J. Wright, "Killerbee: practical zigbee exploitation framework," in *11th ToorCon conference, San Diego*, vol. 67, 2009.

[73] C. W. Badenhop, "A multifaceted security evaluation of z wave, a proprietary implementation of the internet of things," AIR FORCE INSTITUTE OF TECHNOLOGY WRIGHT-PATTERSON AFB OH WRIGHT-PATTERSON âĂę, Tech. Rep., 2017.

[74] B. Fouladi and S. Ghanoun, "Security evaluation of the z-wave wireless protocol," *Black hat USA*, vol. 24, pp. 1–2, 2013.

[75] V. Graveto, T. Cruz, and P. SimÃűes, "Security of building automation and control systems: Survey and future research directions," *Computers & Security*, vol. 112, p. 102527, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404821003515

[76] T. Feng and Y. Wu, "Formal security analysis and improvement based on lontalk authentication protocol," *Security and Communication Networks*, vol. 2022, p. 8104884, Jul 2022. [Online]. Available: https://doi.org/10.1155/2022/8104884

[77] C. Parian, T. Guldimann, and S. Bhatia, "Fooling the master: exploiting weaknesses in the modbus protocol," *Procedia Computer Science*, vol. 171, pp. 2453–2458, 2020.

[78] N. Hussein and A. Nhlabatsi, "Living in the dark: Mqtt-based exploitation of iot security vulnerabilities in zigbee networks for smart lighting control," *IoT*, vol. 3, no. 4, pp. 450–472, 2022.

[79] C. W. Badenhop, S. R. Graham, B. W. Ramsey, B. E. Mullins, and L. O. Mailloux, "The z-wave routing protocol and its security implications," *Computers & Security*, vol. 68, pp. 112–129, 2017.

[80] F. Praus and W. Kastner, "Identifying unsecured building automation installations," in *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*. IEEE, 2014.

[81] O. Gasser, Q. Scheitle, B. Rudolph, C. Denis, N. Schricker, and G. Carle, "The amplification threat posed by publicly reachable bacnet devices," *Journal of Cyber Security and Mobility*, vol. 6, no. 1, p. 77âĂŞ104, Nov. 2017. [Online]. Available: https://journals.riverpublishers.com/index.php/JCSANDM/article/view/5227

[82] M. Cash, S. Wang, B. Pearson, Q. Zhou, and X. Fu, "On automating bacnet device discovery and property identification," in *ICC 2021-IEEE International Conference on Communications*. IEEE, 2021, pp. 1–6.

[83] H. Ochiai, M. D. Hossain, P. Chirupphapa, Y. Kadobayashi, and H. Esaki, "Modbus/rs-485 attack detection on communication signals with machine learning," *IEEE Communications Magazine*, 2023.

[84] T. Zillner and S. Strobl, "Zigbee exploited: The good, the bad and the ugly," *Black Hat–2015. Available online: https://www. blackhat. com/docs/us-15/materials/us-15-Zillner-ZigBee-Exploited-The-Good-The-Bad-And-The-Ugly. pdf (accessed on 21 March 2018)*, 2015.

[85] S. S. Rana, M. A. Halim, and M. H. Kabir, "Design and implementation of a security improvement framework of zigbee network for intelligent monitoring in iot platform," *Applied Sciences*, vol. 8, no. 11, p. 2305, 2018.

[86] V. Merdis, "Wireless communication protocols for home automation exploring the security and privacy aspects of smart home iot devices communicating over the z-wave protocol," Master's thesis, University of Twente, 2019.

[87] J. D. Fuller and B. W. Ramsey, "Rogue z-wave controllers: A persistent attack channel," in *2015 IEEE 40th Local Computer Networks Conference Workshops (LCN Workshops)*. IEEE, 2015, pp. 734–741.

[88] L. Rouch, J. François, F. Beck, and A. Lahmadi, "A universal controller to take over a z-wave network," in *Black Hat Europe*, 2017.

[89] A. Antonini, F. Maggi, and S. Zanero, "A practical attack against a knx-based building automation system," in *2nd International Symposium for ICS & SCADA Cyber Security Research 2014 (ICS-CSR 2014) 2*, 2014.

[90] K. Hofer-Schmitz, "A formal analysis of enoceanâĂŹs teach-in and authentication," in *Proceedings of the 16th International Conference on Availability, Reliability and Security*, 2021, pp. 1–8.

[91] Y. Wu and T. Feng, "An anonymous authentication and key update mechanism for iot devices based on enocean protocol," *Sensors*, vol. 22, no. 17, p. 6713, 2022.

[92] T. Oluwafemi, T. Kohno, S. Gupta, and S. Patel, "Experimental security analyses of {*Non−Networked*} compact fluorescent lamps: A case study of home automation security," in *LASER*, 2013.

[93] K. Jonas, B. Vogl, and M. Rademacher, *Security mechanisms of wireless building automation systems*. Dean Prof. Dr. Wolfgang Heiden, 2017.

[94] N. Tsalis, G. Stergiopoulos, E. Bitsikas, D. Gritzalis, and T. K. Apostolopoulos, "Side channel attacks over encrypted tcp/ip modbus reveal functionality leaks." in *ICETE (2)*, 2018, pp. 219–229.

[95] J.-C. Liou, S. Jain, S. R. Singh, D. Taksinwarajan, and S. Seneviratne, "Side-channel information leaks of z-wave smart home iot devices: Demo abstract," in *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, 2020.

[96] KNX, "Knx ets," https://www.knx.org/knx-en/for-professionals/software/ets-professional/, 2022.

[97] Y. Liu, Z. Pang, G. Dán, D. Lan, and S. Gong, "A taxonomy for the security assessment of ip-based building automation systems: The case of thread," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, 2018.

[98] L. N. Whitehurst, T. R. Andel, and J. T. McDonald, "Exploring security in zigbee networks," in *Proceedings of the 9th Annual Cyber and Information Security Research Conference*, 2014, pp. 25–28.

[99] *Application Note 158/13 v02 KNX Data Security*, 2nd ed., KNX, 2013.

[100] CISA, "Knx protocol," https://www.cisa.gov/news-events/ics-advisories/icsa-23-236-01, August 2023.

[101] *Application Note 159/13 v04 KNXnet/IP Secure*, 2nd ed., KNX, 2013.

[102] KNX, "Knx ip secure becomes world's first vendor-independent security standard for building automation as en iso 22510," https://www.knxtoday.com/2020/02/15234/knx%2Dip%2Dsecure%2Dbecomes%2Dworlds%2Dfirst%2Dvendor%2Dindependent%2Dsecurity%2Dstandard%2Dfor%2Dbuilding%2Dautomation%2Das%2Den%2Diso%2D22510.html, 2020.

[103] *MODBUS/TCP Security*, v21 ed., MODBUS, 2018.

[104] *Security of EnOcean Radio Networks*, v2.5 ed., EnOcean Alliance, 2018.

[105] E. Alliance, "Security of enocean radio networks," https://www.enocean-alliance.org/wp-content/uploads/2023/04/Security_of_EnOcean_Radio_Networks_v3.01.pdf, February 2023.

[106] C. S. Alliance, *ZigBee Pro Specification*, 05th ed., Connectivity Standard Alliance, 2023.

[107] *Z-Wave Plus v2 Device Type Specification*, Z-Wave Alliance, Beaverton, United States, 10 2021.

[108] Calimero, "Calimero project," https://github.com/calimero-project, 2022.