IET Information Security

Research Article

IET Journals
The Institution of Engineering and Technology

ISSN 1751-8644
doi: 0000000000
www.ietdl.org

IoT Security Vulnerabilities and Predictive Signal Jamming Attack Analysis in LoRaWAN

Max Ingham[1], *Jims Marchang*[1], *Deepayan Bhowmik*[2]*

[1] *Department of Computing, Sheffield Hallam University, Sheffield, United Kingdom*
[2] *Division of Computing Science and Mathematics, University of Stirling, Stirling, United Kingdom*
* *E-mail: deepayan.bhowmik@stir.ac.uk*

Abstract: Internet of Things (IoT) gains popularity in recent times due to its flexibility, usability, diverse applicability and ease of deployment. However, the issues related to security is less explored. The IoT devices are light weight in nature and have low computation power, low battery life and low memory. As incorporating security features are resource expensive, IoT devices are often found to be less protected and in recent times, more IoT devices have been routinely attacked due to high profile security flaws. This paper aims to explore the security vulnerabilities of IoT devices particularly that use Low Power Wide Area Networks (LPWANs). In this work, LoRaWAN based IoT security vulnerabilities are scrutinised and loopholes are identified. An attack was designed and simulated with the use of a predictive model of the device data generation. The paper demonstrated that by predicting the data generation model, jamming attack can be carried out to block devices from sending data successfully. This research will aid in the continual development of any necessary countermeasures and mitigations for LoRaWAN and LPWAN functionality of IoT networks in general.

1 Introduction

The term “Internet of Things" (IoT) is an oft-bandied term that refers to a series of many interconnected smart devices with myriad different functions. Such devices have seen an explosion in popularity and public awareness, due in no small part to the various technology centric headlines that have appeared over the years concerning the security of these devices. Various Low Power Wide Area Networks (LPWANs) have emerged in an attempt to tackle the ongoing IoT security problem, but major security issues still remain, with many attacks being fairly trivial to execute against these devices. Since IoT device adoption shows no rate of slowing down, these security issues and any related attacks must be properly analysed and assessed, so that progress can be made to improve the protocols and standards that form the basis of these device ecosystems. Many IoT device standards are far from mature and thoroughly tested, so extreme care must be taken when choosing to implement an IoT network solution, as security must be at the forefront of any IoT deployment.

LPWANs are an attempt to assuage the many operational concerns present with current IoT infrastructure, *e.g.*, enabling devices to send small amount of data over longer distance at low power. According to Cisco, LPWANs accounted for 7% of global Machine-to-Machine communications worldwide in 2016, with that predicted to rise to around 31% by 2021 [1]. Long Range Wide Area Network (LoRaWAN) is a popular open standard supported by LoRa Alliance* and is essentially an effective LPWAN protocol. Therefore it is important, if not obvious, to understand the potential vulnerabilities of LoRaWAN.

The aim of this work is to carefully evaluate the security mechanisms in place in a LoRaWAN network, and whether this security may be compromised effectively in any way. We propose an experimental infrastructure using physical IoT boards and established network, *i.e.*, by sending data from two SODAQ ExpLoRer IoT boards† to the ThingPark network, with an indoor LoRaWAN gateway serving as the attacker attempting to gain information and identify potential attack vectors to disrupt network operations. Motivation of the the work is based on the existence of vulnerability of predictability of the slots of data sending by a device to the gateway. Since the sending slots of a device can be analysed and be predicted in a single channel communication, jamming attacks are realised to be possible.

While this paper outlined and discussed the general attacks that have plagued IoT devices and networks in recent times, our focus is on attacks that can be specifically applied to LoRaWAN. Such attacks and security operations of LoRaWAN are outlined and evaluated in this paper. Fundamental operations of device interconnectivity are analysed, along with various attacks that can be applied due to any failings inherent to the design of LoRaWAN.From this analysis, a justification of approach is defined, which outlines and gives evidence for taking any particular directions, as well as providing predictions and set deliverable. Main contributions of this work are:

- Proposing a framework to determine packet sniffing and eavesdropping in building attack profiles for a target LoRaWAN device, which is achieved by analysing the traffic and identifying exploitable trends.
- Compare the effectiveness of continuous and targeted DoS attacks on IoT devices in a LoRaWAN environment.
- Proposition for remedies to the attack scenarios carried out, and discussion on the implications of such mitigations.

The rest of the paper is organized as follows: Section 2 discuss about the Internet of Things and relevant security concerns in IoT, Section 3 describes about the low powered LoRa and LoRaWAN along with its operational steps. It is followed by Section 4 to analyse the LoRaWAN security and known attacks in details. Experimental Framework, Data Collection and methods of Analysis are discussed

* `https://lora-alliance.org/`

† `https://support.sodaq.com/Boards/ExpLoRer/`

in Section 5 and Section 6 presents the proposed prediction based jamming model along with the list of attack models. It is followed by discussion on the success and failure of the jamming attacks in Section 7 and followed by conclusion in Section 8.

2 Internet of Things and Security Concerns

2.1 Internet of Things and Applications

The Internet of Things is the rapidly growing trend of interconnecting many various types of devices to the internet. Devices in the Internet of Things range from sensing and actuation devices for use in industries including manufacturing and medical, to everyday home devices like televisions, thermostats, and security systems. As we move into the future with all our devices interconnected in this way, the advent of smart homes and cities, which can communicate together to save energy by way of advanced monitoring, as well as facilitating increased control, and greater user accessibility, becomes a reality. The act of devices communicating in this way is known as Machine-to-Machine (M2M) communications [2] define M2M as "communication (that) occurs among machines with computing/communication capabilities without human intervention".

This future of fully automated smart ecosystems could become reality a lot sooner than we may think. According to the research and advisory firm Gartner, 8.38 billion IoT devices were connected in 2017, with that set to rise to over 20 billion devices by 2020, this represents a greater than twofold increase in IoT device adoption in the space of just 3 years [3].

Aside from the home automation and monitoring purposes that have become increasingly more prevalent in general consumer use, one of the fastest rising and main proposed applications of IoT technology is in industrial environments. IoT technology has been developed and deployed for various industries, including; environmental monitoring, healthcare, inventory and production management, food supply, and transportation [4]. The UK government has also committed to driving IoT development and adoption through their IoTUK programme, which aims to achieve goals including smart bus stops with sensors/beacons, and mobile applications that allows people to "check-in" to their bus stop to let bus drivers know they are waiting for service, as well as medical applications like diabetes management, and dementia care [5].

2.2 Security Concerns in IoT Based Network

As with any interconnected network of devices, security is a paramount concern, but it is more challenging with the IoT mainly because the end devices are low powered and have limited computation power. As these networks grow, with more sensing devices being added at accelerated rates, not only does the amount of data being generated increase, but also the security risk associated with such growth increases proportionally. Talwana & Hua [6] pointed that a breach in an IoT network is unlike other high profile hacking episodes, which usually aim to compromise online data and privacy. Instead, IoT device insecurity can open up a gateway for the entire network to be compromised. Compromising a private key or a session key could be harder compared to a simple but effective attacks like replay attack and ack spoofing, which could happen in a low powered WAN like LoRa [7]. The same group of authors also explored another set of vulnerability in which a malicious gateway can be created and integrate into a network using a UDP spoofing attacks [7], which is highly possible in LoRa due to dependant on a gateway to connect to internet. It is interesting to note that an ack frame can be spoofed and use for acknowledging older or other data frames through the gateway, the only challenge could be using the right frame number if used. The traditional issue of a Denial of Service or Distributed Denial of Service attacks remains an open challenge in such low powered LoRa networks too.

With this in mind, the explosive growth of connected IoT devices should be a cause of real concern and trepidation, especially due to the potential usage of less mature standards that facilitate these communications. This is in contrast to time tested protocols, like the TCP/IP suite, that more traditional internetworking devices are built upon. Moganedi & Mtsweni [8] outlined that while IoT devices can introduce great convenience into human life, if the mass amount of personal data that can be collected are not secured or improperly protected against being leaked, any desire to adopt or use such technology will wane and the potential and the ability of what IoT devices can do will be less explored.

In a similar vein to Mirai, a new malware known as "*Brickerbot*" began targeting devices in a similar fashion. The difference this time, was that Brickerbot actually rendered devices unusable so that they couldn't be leveraged for use in DDoS attacks, hence the name. According to Mansfield Devine [9], the following weeks after the initial Brickerbot assault, updated versions started to appear which targeted different protocols and interfaces, rendering even more devices irrecoverably unusable. With so many devices now connected, as well as even the most conservative estimates predicting adoption rates of IoT devices to increase exponentially in the near future [1], one would be forgiven for thinking that the future of IoT security may seem bleak with the far too frequent reports of security issues arising around these technologies.

On the subject of the wariness of IoT device adoption, any simple online search yields numerous headlines that detail massive security flaws with IoT devices. Hackers at the *DEF CON*[*] security conference in 2016 found 47 new vulnerabilities in 23 IoT devices, including vulnerabilities with smart door locks, padlocks, thermostats, refrigerators, and even wheelchairs [10]. While many of these devices are consumer based ones not meant for industrial applications, such regular reports on the security concerns of these devices can sour public opinion of their use, which could ultimately slow down adoption rates. It seems until a common standardisation of IoT device communication protocols and security emerges, whether for commercial or industrial use devices, there will continue to be headlines of security breaches and hacking attempts made against the IoT.

The most recent and high profile attack that leveraged IoT devices was the Mirai botnet attack, which was used to cripple the services of large online companies, including web host and cloud provider OVH, as well as DNS service provider Dyn, which in turn affected the operation of popular sites like Twitter, Netflix, Reddit, and Github [11]. The main concerning factor of Mirai is in the simplicity of its operation. Kolias *et al.* [11] outlined Mirai operation thusly, first, Mirai scans random public IP addresses through TCP ports 23 or 2323. Next, the malware engages in a brute-force attack in an effort to discover which of the 62 possible hard-coded username and password pairs match for the device. When shell access has been gained, the Mirai loader software downloads the malware to the infected device and attempts to protect itself from other malware by shutting down ports such as Telnet and SSH. From this point onward, the infected device may attempt to infect other devices on the network, as well as compete in Distributed Denial of Service (DDoS) attacks against IP addresses specified by the Botmaster. The fact that Mirai could infect so many devices shows the sorry state of consumer IoT device security so far. It becomes quickly apparent that many IoT device manufacturers release their products to the market with woefully insecure default administrator credentials across all their devices, as well as having unnecessary network ports open that help facilitate these attacks.

Other form of attacks are Malware attacks and in recent times a high profile of such attacks on IoT devices has occurred *e.g.*, Mirai, and Brickerbot. It's not an unreasonable suggestion that security world could be heading towards a potentially devastating Stuxnet-like scenario in the near future. Such issues are exacerbated by the fact that deployed IoT devices, whether in domestic locations or

[*] https://www.defcon.org/

industrial plants, rarely see firmware updates of any kind often, or even at all. As noted by Teng *et al.* [12], if automatic updates are unavailable, that only leaves the option of a manual firmware update, which can be extensively time consuming (especially if the user is dealing with a large network of devices), as well as being error prone and potentially difficult. In an industrial setting, devices are generally deployed for long periods of time, and so, must be resilient to drastic changes and able to adapt to the ever changing security landscape. If software patches/updates aren't pushed out frequently enough, or indeed ever, the security threat posed to the network could outweigh the benefits of IoT device integration.

Security challenges threats and solutions are highlighted by [13] and [14], regarding replay attack, jamming attack, buffer reservation attack, spoofing attacks etc in IoT networks. Khan, M.A. *et al.* [13], also suggest Blockchain as a solution with a hope to secure data, prevent data hijacking attack, unauthorized access, man-in-middle attack and for maintaining data integrity. The authors [15], suggest to use a random number in a join procedure of a LoRaWAN protocol to avoid replay attack and tested the performance by using jammer and avoiding jammer. Knowing the packet generation pattern will have no impact if a jammer is used to jammed the entire channel continuously. In specific context to LoRa, Eldefrawy, M. *et al.*[16] conduct an extensive formal analysis to understand the vulnerabilities of LoRa network of LoRaWAN v1.0 and v1.1. It is found that LoRaWAN attacks on Niagree (non-injective agreement claim) and Nisynch (non- injective synchronization) attacks occurs in LoRAWAN v1.0 end devices, but in LoRaWAN v1.1 those vulnerabilities are patched. However, the vulnerability test conducted. Every wireless communication technology is susceptible to interference and jamming attacks. Eldefrawy, M. *et al.* [16] think that threat of signal jamming is not a serious issue because LoRa spread the use of wireless communication channels to a wider band unlike Bluetooth, Near Field Communication etc. However, even in presence of multi channel, selective jamming attacks cannot be prevented if the attacher knows the channel in which the device is sending and if the device communicates using a same fixed channel. In presence of a wide spread multi channels, LoRaWAN can use different channels for a same end device, so selective channel jamming may not harm all the transmitted data because some data taking different channel will eventually be delivered. Can an attacker learn the pattern at which the data is send by a device and know the pattern at which the device switches its channel? If so, can the attacker jam the channel only when the the data is on transit knowing the pattern of sending and knowing the channels used, which is the motivation of this work.

2.3 Other Current Security Challenges

One of the main reasons for why there's been no widely applied/adopted solution to IoT security, is due to the majority of these devices requiring low power computation to achieve optimum operating efficiency and lack of security. The trade-off with this, is that the devices are mostly incapable of utilising intensive and secure encryption methods, among many other factors.

LoRaWAN adopts Advanced Encryption Standard (AES) method for ensuring data confidentiality, which is a symmetric block cipher algorithm known for its efficiency and its fast and strong algorithm [17], AES can be implemented via hardware and software. Bui, Puschini, Bacles-Min, Beigné, & Tran [18] outlined that while AES is designed to benefit from software optimisation in modern computing systems, software implementation of AES introduces data processing and transmission delay, as well as an increase in power and energy consumption. Hardware implementations can provide high performance and throughput, but suffers from the same issue of high power consumption, which can be detrimental to IoT device operation.

In order to ensure data confidentiality and privacy, LoRa adopts encryption, however, many are attempting to designing methods that will consume minimal power with least computation. The traditional AES encryption consumes too much computation and power for low powered end devices. Thus, ([19]), proposes an AES model in which the number of encryption cycles are reduced in order to compensate the power loss at the end devices. The new method also incorporated techniques to avoid replay attack, known-key and eavesdropping attacks. However, reducing the number of encryption cycles means that its less secure, but providing security is always better than open end connection. The same authors *i.e.*, Tsai, K.L. *et al.* [20] also designed another variant of AES encryption method for LoRaWAN in 2019.

Bui *et al.* [18] stated that lightweight block cipher algorithms have begun to emerge recently. These algorithms are lightweight in their software and hardware implementations and result in lower memory footprints, but come with the trade-off of reduced security levels. Examples of such algorithms include "PRESENT" and "CLEFIA". Both of these implementations use more encryption rounds and smaller block sizes than AES, which leads to lower throughput. However, these lightweight algorithms are not yet adopted in new IoT proposals due to insufficient studies in terms of security and protocols [18]. Ensuring privacy and data confidentiality from end-devices to gateway is good, however, ensuring privacy between end-devices to application server directly is a better option, because higher the number of hops higher is the risk (more attacking points). So, providing data and communication security should be provided from end-to-end *i.e.*, from end-devices to the LoRaWAN gateway to the Network Server to the Application server, however, computation power and energy requirement for the computation will be a daunting tasks because end-devices are generally lightweight in nature. One such system is designed by [21]. In a normal LoRaWAN model, AppSKey key is used separately for securing between a Network server and an Application server.

3 Long Range Wide Area Network (LoRaWAN)

With the focus on Long Range Wide Area Network(LoRaWAN) security, in this section we briefly discussed LPWAN the background of LoRaWAN that leads to further security analysis in the following sections.

3.1 Low Power Wide Area Networks (LPWAN)

The main characteristics of a Low Power Wide Area Networks (LPWAN) network include ultra low-power operation of nodes to reduce costs and environmental impact of frequently changing batteries, the network should not require nodes to *wake up* unless there's a need to send or receive data (the ALOHA system), and data transfer should be fully secured [22]. Many well established and utilised LPWAN technologies already exist in the wireless IoT communications space, including LoRaWAN, Sigfox, Weightless-W, N and P, and DASH7. SigFox is perhaps one of the most commonly adopted LPWAN solutions next to LoRaWAN [23].

According to Silva *et al.* [24] the most glaring issues with SigFox in comparison to LoRaWAN, is the restrictive data rate uplink limit of 100b/s, a maximum packet payload of 12 bytes, and the number of packets per end device cannot exceed 14 packets a day. Due to these drawbacks and limitations imposed by SigFox, LoRaWAN seems the more robust option to explore in this paper. The LoRa Alliance outline the data rate of LoRaWAN as 250b/s - 50kb/s, with an unlimited amount of messages per day allowed to be sent by devices [25].

Despite the many LPWAN solutions available from different organisations and vendors, the focus of this work is to be turned to the particular IoT network that is the open source LoRa network due its popularity. To facilitate maximum low power operation of devices, LoRaWAN aims to optimise every aspect of device communication.

3.2 LoRa and LoRaWAN

At this juncture it is important to understand the distinction between LoRa (Long Range) and LoRaWAN (Long Range Wide Area Network). LoRa defines the physical layer wireless radio modulation

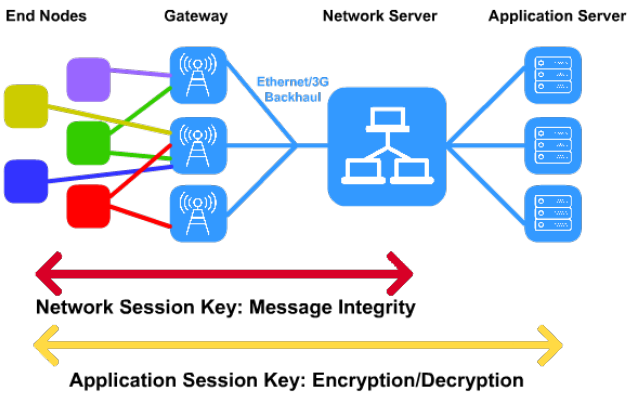

Fig. 1: Example LoRaWAN session.

used to create the long range communication link. Unlike certain legacy devices that use Frequency Shifting Key (FSK) physical layer modulation, LoRa uses Chirp Spread Spectrum (CSS) modulation to maintain the same low power characteristics of FSK, while significantly increasing the range of communications [25]. Depending on the region, LoRa can operate on the 868 (EU), 433 (EU), 915 (US), or 430 (AS) MHz ISM bands.

LoRaWAN in contrast, defines the communication protocol as well as the underlying system architecture for the network. LoRaWAN is especially important due to it being the deciding factor in determining the power consumption, and in turn, the battery life of a node, the capacity of the LPWAN network, the quality of service (QoS), and perhaps most crucially, the security and variety of the network applications [25]. While LoRa modulation technology is proprietary, LoRaWAN is an open standard in active development by the LoRa Alliance.

3.3 LoRaWAN Overview and Operation

As shown in Figure 1, the nodes in a LoRaWAN network are not primarily associated with a specific gateway, data transmitted by a node is typically received by multiple gateways. Upon receiving the data from the node, the gateway forwards the packets to the cloud-based network server via traditional network standards like Ethernet, Cellular, and Wifi. This network server will handle any intelligence and complexity, and will perform any needed filtering, security, and scheduling tasks [25]. The network server also performs the Adaptive Data Rate (ADR) needed for optimising the data rate and energy consumption of nodes in the network.

LoRaWAN divides devices into classes depending on what requirements the end device needs to serve. The classes trade off network downlink communication latency versus battery lifetime. Class A devices are grouped as *battery powered sensors*, which are the most energy efficient and must be supported by all devices on the network. Class B devices are known as *battery powered actuators*, which are energy efficient with latency controlled downlink, and finally, Class C devices are known as *main powered actuators*, since these devices don't run off of a battery, they can afford to listen constantly with no latency for downlink communications [25].

4 LoRaWAN Security and Known Attacks

LoRaWAN utilises AES-128 encryption for secure end-to-end encryption of exchanged application payloads between end-devices and application servers. However, Na *et al.* [26] revealed that during the Over The Air Authentication (OTAA) join method, in which an end-device and network server exchange messages to initiate the joining procedure, the join request message sent from the end-device, through the gateway to the network server, is un-encrypted. The authors theorised if an attacker were able to collect enough join request messages of certain end-devices and determined the optimal time to attack the target, they may initiate a replay attack in which the attacker sends a constant stream of the collected join request messages, leading the network server to attempt to connect with the attacker device, and discarding the request messages of each targeted end-device.

In addition to replay attacks, LoRaWAN is susceptible to many other security flaws and attacks, including weaknesses in key management, counter management, bit flipping attacks, eavesdropping, and encryption flaws [27]. Due to the coexistence issues faced in LoRaWAN, Denial of Service (DoS) can also occur if messages collide, although little research has been carried out with regards to DoS attacks in LoRaWAN, outside of replay attacks. This section aims to provide a much greater, and more in depth analysis of these security flaws and attacks which are also experimentally validated.

4.1 LoRaWAN Keys Evaluation

Devices on a LoRaWAN network have a session with the network server. Typically, this session contains the device address, and two session keys (NwkSKey and AppSKey). Frame counters count the number of uplink and downlink messages in the session. An Uplink message is a message from a device to an application, whereas a downlink is the opposite [28]. Since the network server can also incorporate the application server, two scenarios are depicted (as shown in Figure 1) to capture the real world application scenarios according to the LoRaWAN v1.1*.

Since LoRaWAN is a radio protocol, capturing transmitted messages is a trivial matter. However, it is impossible to read these messages without the AppSKey, as the message is encrypted. Tampering is also not possible without the NwkSKey, as the MIC (Message Integrity Check) will fail.

Keys differ depending on which activation method was utilised, with the available options being OTAA (Over The Air Activation) and ABP (Activation By Personalisation). The session keys and app key all have a length of 128 bits [28].

ABP keys:

- **NwkSKey:** Network Session Key. Used for identification and message integrity.
- **AppSKey:** Application Session Key. Used for payload encryption/decryption. Unless Fport is set to 0, then NwkSKey is used instead [27].
- **DevAddr:** Device Address. Used for identifying the device within the network

These keys are manually assigned to the devices in ABP. The advantage of this is the devices can begin sending data immediately upon power up, as well as if keys are stolen, the affected device can be easily deprovisioned without having to change all the other devices. However, having to keep track of frame counters for the messages between the device and network server between power cycles is a downside of this. The server will usually ignore messages with frame counters that differ from expectations, but this is usually disabled in testing environments.

OTAA keys:

- **AppEUI:** Application Identifier - Uniquely identifies the application

* *https://lora-alliance.org/resource-hub/lorawanr-specification-v11*

- **DevEUI:** End Device Identifier - Uniquely identifies the device
- **AppKey:** Used to derive the session keys (NwkSKey and AppSKey)

OTAA uses an application ID (AppEUI) and device ID (DevEUI) along with an application key (AppKey) to derive the network session key (NwkSKey), application session key (AppSKey), and the device address. The device address is dynamically assigned by the network.

When the end device is powered up, a join procedure is initiated where the negotiation of a new set of keys based on the application key is performed. After the negotiation, which usually takes around 5 seconds, the device behaves just like an ABP device. If the device is powered off, the join procedure must occur again. The biggest weakness with this method is the application key (AppKey), if this is stolen, an attacker can impersonate any device in the network. Key management using OTAA is much easier however, and has the benefit of key regeneration upon each new join procedure, something ABP lacks.

4.2 *Encryption and Protocol Vulnerability*

Message encryption in LoRaWAN is performed using AES128 in CTR mode. As previously outlined, the AppSKey is used for the encryption of the payload, unless the FPort is set to 0, then the NwkSKey is used instead. AES is a symmetric encryption algorithm, this means the sender (the sensor node) and receiver (the network server) use the same key to encrypt and decrypt the messages. The glaring downside of this, is if an attacker can obtain the symmetric key, all messages encrypted with that key can be read.

In LoRaWAN, a block cipher mode is used. The encryption is performed as follows, a keystream is produced using the FCntUp or FCntDown values, then, the plaintext frame payload is XOR'd with the keystream to produce the ciphertext, *i.e.*, Ciphertext = Payload $\oplus$ Keystream. Information such as the FPort and counters are sent unencrypted.

The issue with this method, is the use of the counters in the keystream instead of a cryptographic nonce. A nonce in cryptography is any random or pseudo-random arbitrary number that is used once, the counters used may increment and change on every message sent, but forcing the node to restart and rejoin the network would reset the counters to 0. In this case, an attacker would have knowledge of the current values of the counters, due to the linear way in which they increment. This could potentially lead to some identifiable, regular patterns in the resulting ciphertext.

If a situation arose where the counters didn't increment, or if the node was forced to rejoin the network and reset its counter values, a potential attacker who already had knowledge of the plaintext for one message could XOR the known plaintext with the ciphertext to reproduce the keystream [27]. Figure 2 displays how this could operate.

4.3 *Replay Attack Analysis*

Despite the need for certain session keys to tamper with packets, it's still possible for an attacker to retransmit these captured messages, known as a *replay attack*. These kinds of attacks may be detected and blocked by using frame counters. Upon initial device activation, both frame counters, FCntUp and FCntDown, are set to 0, whenever an end device transmits an uplink message, the FCntUp counter is incremented, and every time the network sends a downlink message, the FCntDown is incremented [28]. If the device or network server receives a message with a frame counter lower than the last one, the message is dropped.

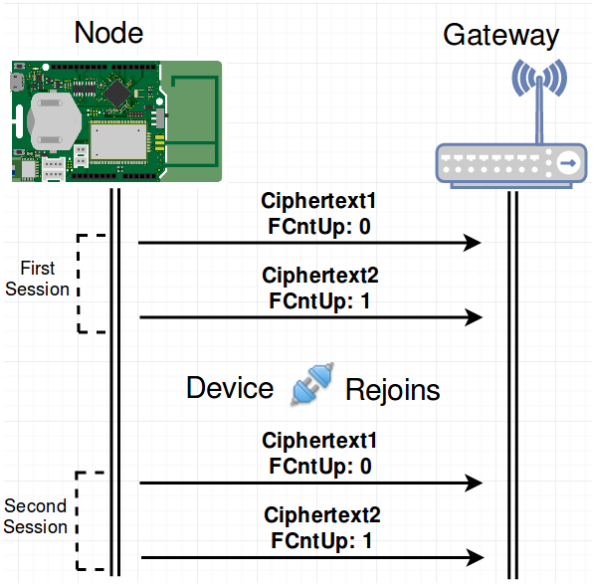

Fig. 2: Counter reset on forced rejoin

Frame counters reset to 0 upon every power cycle of a device. For ABP activated devices, this can be an issue when in the development stage, as some IoT platforms, like The Things Network (TTN), will discard messages and require the device to be re-registered. Based on initial testing, the ThingPark development mode seems to have this particular feature disabled, as devices don't need to be re-registered upon power cycle to continue sending messages. Due to this, it would be possible for an attacker to target some development environments, although such targets are less critical.

4.4 *Bit Flipping Attack Analysis*

Bit flipping is another attack that can be utilised in a LoRaWAN network. A bit flipping attack consists of changing specific portions of ciphertext to alter data without the need to decrypt it first. Since devices in a LoRaWAN environment utilise AES in CTR (Counter) mode, this makes it possible to perform a bit flipping attack as CTR mode simply performs an XOR logical operation for encryption of the plaintext [29]. The XOR operation keeps the order of the plaintext bits, the unshuffled nature of this enables the use of bit flipping attacks.

As outlined in Section 4.1, an attacker would need the network session key to also tamper with the MIC, as failure to change the MIC would result in a MIC mismatch and the packet being dropped. Figure 3 displays the process of a bit flipping attack failing due to a difference in MIC.

Since the only way to change the MIC would be to physically extract the network session key from the nodes, this method of attack is unlikely to occur in reality. Message integrity isn't checked between the network server and the application server, however; these two roles are often incorporated as a single device.

4.5 *Eavesdropping/Reconnaissance Attack Analysis*

Due to the fact that LoRaWAN is a radio protocol, capturing potentially large and varied amounts of data would be a trivial matter for any attackers within proximity of the sending nodes. An attacker in this situation would need a device capable of sniffing LoRaWAN packets and storing them for later analysis, as well as being knowledgeable about the data types and formats utilised in LoRaWAN.

Collecting numerous LoRaWAN messages, though the payload is encrypted, is a great way to gather information about the network and the devices on it. After enough data has been collected, it may be possible to apply certain techniques to analyse the encrypted data, like machine learning applications to identify trends and regularities.

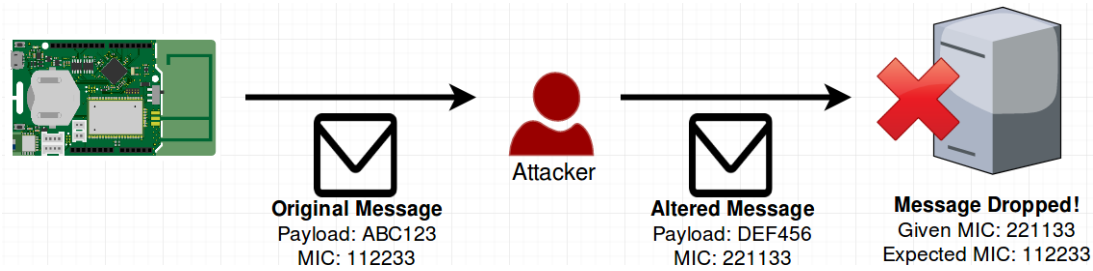

Fig. 3: Bit flipping failure attack due to MIC difference

Aside from the encrypted data, LoRaWAN packets leave much valuable information unencrypted, including frame counter values and device addresses. This data could be used to plot attacks and profile target devices.

Specifically, the data gathered by eavesdropping communications in this way could aid in choreographing DoS attacks in LoRaWAN. Having knowledge of message types, message arrival times, and sequence numbers, potentially enables an attacker to plan and execute such attacks. To understand more about how this could work, Section 4.6 outlines the operation of DoS attacks in LoRaWAN networks.

4.6 Threat Model: Denial of Service in LoRaWAN

Denial of Service is an area that appears grossly underrepresented in various research papers that outline attacks on LoRaWAN and similar LPWAN technologies. The main documented method of performing DoS attacks in LoRaWAN is via use of signal jamming and replay attacks. Aras *et al.* [30], focused on the effect that the Spreading Factor plays in jamming devices on a LoRaWAN network, but limited the attacks to a device sending on a single channel. The duty cycles used by a device in LoRaWAN will depends on whether the device uses single channel or multiple channels. If a single channel is used then the device can set to uses n time units for every m times unit where n<m, and the free slots are made available for other devices. When an end device operates on multi-channel, then in channel 1, the first n slots can be used for the first message and then use the (n+p) slots for the second message and so on for other channels for every m times unit where n=p and n or p < m. The main difference with other form of wireless communication say in WiFi radio communication is that WiFi radio works on carrier sense multiple access collision avoidance (CSMA/CA) where the participating devices competes for the channel and if collision occurs then an access attempt is made after a random amount of time. It means that it will be very hard to predict the channel access pattern of a device due to adoption of random access patten, so jamming the data generated by a particular device without jamming the entire channel will be next to impossible. Thus the nature of predictive jamming attacks on a particular device data generation in LoRaWAN and WiFi are different in nature. Taking such behaviour into account is important to ascertain how DoS attacks could be performed in real world production scenarios, as opposed to more controlled labs.

Jamming a channel continuously is easy and it will be easy to detect too if the channel is jammed continuously, however, if jamming is conducted only when the data is about to send or as long as the data is on transit then it will also jammed successfully, however, it will be very difficult to learn about the jamming. Thus, the jamming adopting predictive model will be smooth as well as very efficient and also act as anti-jammer detection proof. The aim of this paper is to study such a predictive jamming attack model in LoRaWAN.

Since LoRaWAN isn't based on traditional IP network principles, the process for carrying out these attacks, as well as their remedies, are less well established. Utilising eavesdropping attacks to gain an accurate profile on the target device is a method that could be used to great effect in determining the optimal window of opportunity for an attack. Since much valuable information is left unencrypted, including counters, airtime, channel, and device address, an attacker may build an accurate profile and tailor their DoS attack appropriately.

Replay attacks, as outlined in section 4.3, can produce a denial of service if executed correctly. Much work has gone into studying these attacks, but not other methods of DoS attacks. Despite the low message and data rate of LoRaWAN devices, it should still be possible to either overwhelm a gateway, especially an off the shelf indoor model, as well as blocking messages by sending data at the same time as a target device. Performing a DoS attack with this method instead of a replay attack also eliminates the need to reset the target device, an action unlikely to be achieved in practice if an attacker does not have physical access to the nodes.

As research has already been performed concerning the DoS capabilities of replay attacks and the OTAA activation method, but little to no research has been undertaken that experiments with ABP activated devices and more traditional DoS methods, the focus for the experiments in this work will be attempts at DoS attacks with ABP activated nodes.

4.7 Discussions

While many attacks and protocol overviews have already been discussed and analysed so far, due to insufficient varied research and results in this area, Denial of Service attacks used in conjunction with Eavesdropping attacks will be the focus in this paper from this point onward. The experiments will aim to build upon some of the work done by Aras [30], but with less emphasis on the Spreading Factor, and more on determining the ideal conditions for an attack, with focus on a more real world scenario, like the normal operational use of multiple channels. An attempt to implement a solution to such attacks will also be carried out.

This work is unconcerned with the contents of data such as the encrypted payload, as much research has already been carried out regarding this. Instead, it's the ample amount of data readable by anyone with a LoRaWAN enabled device that is of particular interest to this work, as this information is potentially of great consequence to the security of devices on the network.

To prove this passively collectable data can be used to great effect, scenarios will be presented in which regular, continuous DoS attacks will be launched, as well as more precise and targeted DoS attacks. The aim is to properly demonstrate the dangers of packet sniffing in a LoRaWAN environment, as well as show how easily DoS attacks may be launched, even by attackers with limited knowledge, thus proving the inherent weaknesses in the LoRaWAN protocol that still exist to be fixed.

It is predicted that the careful and thorough analysis of this passively gathered gateway data will enable the creation of a scenario in which a perfected, highly accurate and repeatable DoS attack may be carried out to great effect. It's also predicted that small payload sizes can change the outcome of an attack.

Based on all the currently analysed information, our quest is whether LoRaWAN is a secure LPWAN solution resilient to easily performed attacks. This can be further broken down into four subcategories:

- How easy is it to gather information on target devices, and how can this information be used to generate attack scenarios?
- How difficult are DoS attacks to perform based on the information gathered, and are there any particular factors that aid in generating better attack scenarios?
- Should the attacks be successful, what effective countermeasures may be implemented?
- Based on the results from the experiments and research, are LPWANs, specifically LoRaWAN, a viable solution to addressing the security concerns faced by IoT devices?

In an attempt to answer the above points, various attack scenarios are carried out in which one LoRaWAN enabled device attempts to block messages from the other. This will be achieved by use of a rogue gateway that will passively sniff data for later analysis. Messages will be sent at variable times and recorded, to identify exploitable patterns for use in the attacks. Codes are written in the Arduino IDE and uploaded to the LoRaWAN boards for execution. A further expansion and explanation on experimental setup and methods can be found in the following section.

5 Experimental Framework, Data Collection and Building the Predictive Model

5.1 Experimental Setup

The setup for this scenario involves two SODAQ ExpLoRer boards and an indoor LoRaWAN gateway. Much of the data are analysed through the use of the ThingPark wireless logger, as well as the internal logs kept by the indoor gateway. The gateway is connected via Ethernet to a basic home router, which then connects to the ThingPark network server. ThingPark is used as it provides extended functionality in analysing message types and data.

The gateway must be added manually to the ThingPark development environment before messages are forwarded to the wireless logger. Firewall ports 21, 22, and 2404 must be open for the gateway to contact the network server correctly. This indoor gateway will be used as the rogue gateway needed to capture and analyse LoRaWAN packets.

Since ABP is the method used for the activation of devices in this scenario, the SODAQ boards must also be configured for use in the ThingPark dashboard. The Device Address, Network Session Key, and Application Session Key are specified for the device, with these values being derived from the DevEUI in this instance, although any random value may be set for these without penalty. Spreading Factor 12 is used for all configured devices for these experiments.

The Arduino Integrated Development Environment (IDE) is utilised throughout this paper to upload the sketches to the devices. Many of these sketches consist of simple methods of pushing various data types to the network server. The appropriate session keys are configured in the code before compiling. The aim of these scenarios is to emulate the conditions in which a small scale LoRaWAN deployment would operate. This means no real restrictions, apart from the set locations of the nodes, and their proximity to the gateway, are imposed. Devices are also not restricted to any particular channels, in an attempt to produce a more natural scenario.

The first SODAQ ExpLoRer (E1) is acting as the attacker device in this setup, while the second device (E2) is taking the place of the target. E1 will attempt to block the communications of E2 by sending junk messages concurrently as the legitimate messages from E2. More detailed information about the hardware and software tools used in this scenario can be found in Figure 4 and Figure 5.

Tool	Information
SODAQ ExpLoRer	The SODAQ ExpLoRer is intended for use as a development tool to evaluate the practicality of utilising LoRaWAN in a development lab environment before moving to a production one.
LoRaWAN UfiSpace Pico Cell Wi-Fi Ethernet Gateway	Gateway used for home testing of LoRaWAN solutions. Detailed logs are easily exportable from the web interface.
Arduino IDE	IDE used to compile and upload sketches to the SODAQ boards. Many libraries are available to extend code functionality.
ThingPark Dashboard	Network management solution for LPWAN technologies.

Fig. 4: Tools overview

Fig. 5: Testing equipment, indoor gateway and SODAQ boards

5.2 Passive Data Collection

With the indoor gateway, it is possible to capture any LoRaWAN data transmitted within range of the device. By exporting this data into a log file and applying filters, an attacker may quickly pick out relevant information to apply to their attack scenario. The first step for an attacker is to identify the address of the device they wish to perform the DoS attack against, and then determine the common themes in transmission that may be exploited. Figure 6 shows a small sample of the data that the indoor gateway can collect, information like the device address of the sending device, channel information, spreading factor, and the FCnt (seq) value are unencrypted and readable by the attacker.

From these log files, an attacker could pick any device that reliably sends data at plottable time intervals and centre the denial of service around that. For example, analysis of these log files show the device with the address “002044BE” (E2) is a likely candidate due to not only the frequency of the sent data, but also the current frame counter value, which shows the device has been sending data for a reliable period up to this point. Knowing these details allows the attacker to create a reliable baseline for the operation of the target device, so that future predictions can be made which aid in the execution of the DoS attack.

When enough data has been passively collected and broadly inspected, the next stage in identifying the correct attack vectors is to do a much more thorough analysis to identify the exploitable trends in the device E2’s operation.

5.3 Building the Prediction Model by Analysis Packets

The first step in the LoRaWAN packet analysis is accurately determining the average time elapsed between each message sent by the target device E2. Having this information correct to the second not only provides the attacker with knowledge about the optimal window to attack in, but also enables the prediction of arrival times for all future messages.

```
17:23:06.689  (1616) [../lgw_x1.c:1067] PKT RECV tms=944666930 tus=3969838356 if=6 status=CRCERR sz=53 freq=868300
17:23:06.710  (1616) [../lgw_x1.c:1067] PKT RECV tms=944666951 tus=3969863452 if=4 status=CRCOK sz=53 freq=867900
17:23:06.710  (1616) [../lgw_x1.c:1091] PKT RECV data='40be442000005e000124c184cec3739b3895d55e61594ecbd4fd26044d5
17:23:06.710  (4576) [../main.c:3022] packet sent to LRC=0 lrrid=c00000f5 by order rssi=-23.000000 snr=12.500000
17:25:06.359  (1616) [../lgw_x1.c:1067] PKT RECV tms=944786597 tus=4089510740 if=3 status=CRCOK sz=53 freq=867700
17:25:06.360  (1616) [../lgw_x1.c:1091] PKT RECV data='40be442000005f000198c8170c5f877ea113ad570c8b0c03b7c068b5163
17:25:06.360  (4576) [../main.c:3022] packet sent to LRC=0 lrrid=c00000f5 by order rssi=-22.000000 snr=12.750000
17:25:24.176  (1616) [../lgw_x1.c:1067] PKT RECV tms=944804413 tus=4107356259 if=1 status=CRCERR sz=167 freq=86736
17:26:05.937  (7644) [../rttping.c:224] ping thd ppp0 st=15490 ok=0 ls=15490 nb=0 av=0 dv=0 mx=0
17:26:57.122  (8959) [../rttping.c:224] ping thd eth0 st=15490 ok=14975 ls=515 nb=5 av=23 dv=0 mx=24
17:27:06.009  (1616) [../lgw_x1.c:1067] PKT RECV tms=944906242 tus=4209157780 if=1 status=CRCOK sz=53 freq=867300
17:27:06.009  (1616) [../lgw_x1.c:1091] PKT RECV data='40be442000006000011acb46e2a03594f41b333d35b58de359b4b1937d0
17:27:06.009  (4576) [../main.c:3022] packet sent to LRC=0 lrrid=c00000f5 by order rssi=-23.000000 snr=11.000000
17:27:17.566  (1616) [../lgw_x1.c:1067] PKT RECV tms=944917799 tus=4220744803 if=2 status=CRCERR sz=227 freq=86750
```

Fig. 6: Sample indoor gateway log data

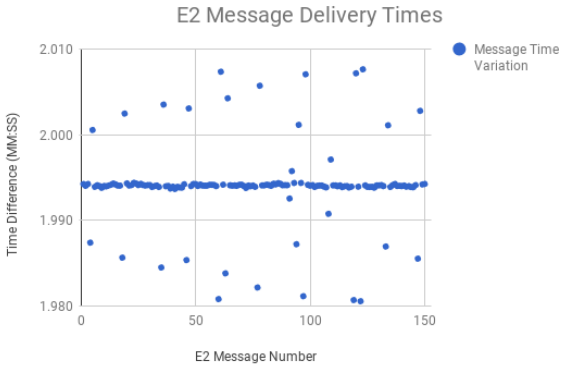

Fig. 7: Message delivery time differences for device E2

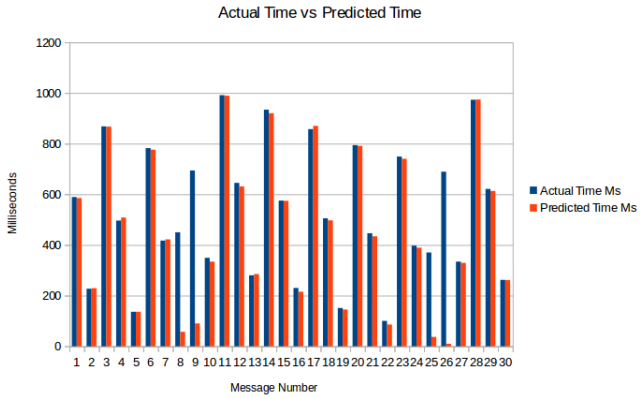

Fig. 8: Actual millisecond value vs predicted

The data in Figure 7 is produced from a sample of 150 packets from E2 collected by the indoor gateway. Aside from a few outliers, the average time between delivered packets for E2 lies at 1.9 minutes, or more precisely, 1.994. This works out as exactly 1 Minute 59 seconds 64 milliseconds. With this data, it's possible to accurately plot all the expected message delivery times for E2 from the last recorded message. This predicted data can then be compared against the actual message delivery times to ensure the reliability of the calculations as shown in the following equation.

$$PredictedTime = TargetLastMsg + AvgMsgTime \quad (1)$$

If the last recorded message that the rogue gateway picked up was at $19:02:48.945$, the simple formula above can be used to estimate the message delivery times for the device E2 for the next hour:

$$PredictedTime = 19:02:48.945 + 00:01:59.64 \quad (2)$$

Figure 8 shows the predicted message arrival time utilising the formula above, next to the actual message arrival time. The predicted messages have been calculated based on the last received message at $19:02:48.945$, all results in the graph focus solely on the millisecond value of the times.

Figure 9 shows a sample of the first 10 results from the graph above. Doing so gives more granular information and shows just how accurate the predictions were when compared against the actual message delivery times of E2. Although outliers still exist here with the values 393 and 396.

Predicted Message Arrival Time	Actual Message Arrival Time	Difference (ms)
19:04:48.585	19:04:48.589	004
19:06:48.229	19:06:48.227	002
19:08:47.867	19:08:47.868	001
19:10:47.508	19:10:47.496	012
19:12:47.136	19:12:47.136	000
19:14:46.776	19:14:46.782	006
19:16:46.422	19:16:46.417	005
19:18:46.057	19:18:46.450	393
19:20:46.090	19:20:45.694	396
19:22:45.334	19:22:45.349	015

Fig. 9: Sample of 10 values

6 Proposed Prediction based Jamming Model

With the analysis and the prediction model in the previous section, we performed two forms of attacks; the first is launched against E2 device and in the second attack, E1 is periodically reset.

6.1 *Testing an Attack form 1: Continuous Jamming Attack*

The first attack to be launched against the E2 device is a continuous DoS attack in which specific message times are not an issue. To perform this attack, the SODAQ device E1 will continuously stream data in the hope that a collision occurs to disrupt the messages sent from E2. The delay value in the Arduino sketch is set to a single millisecond, and the payload of the messages will be a consistent 40 bytes. This test takes place over an hour.

The results in Figure 10 show the apparent failure of device E1 to block any messages from E2 when utilising a continuous stream of messages. Despite the delay value for the code uploaded to E1 set to a millisecond, the device only managed to send 11 more messages than the target device E2. The reason behind the lack of messages from the device E1, is due to the duty cycle observed by the RN2483 wireless module on the SODAQ boards.

The data in Figure 11 can be analysed to show the time taken between messages during the continuous message stream over an

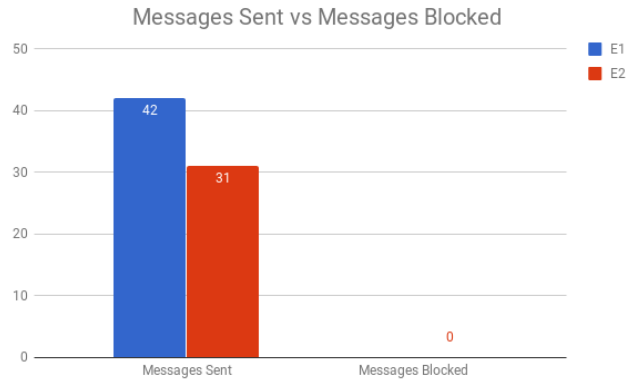

Fig. 10: Messages sent vs blocked over one hour for attack form 1

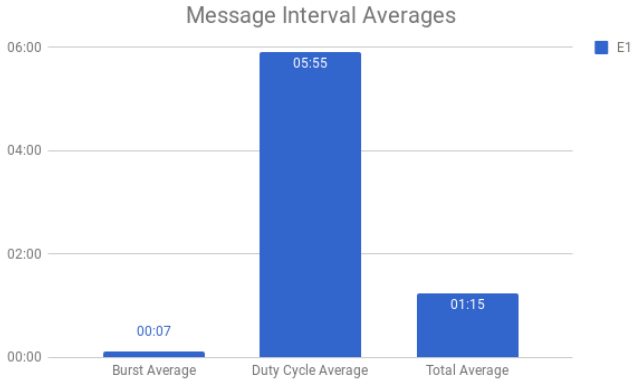

Fig. 11: Average intervals between messages (mm:ss)

hour. The graph shows that despite the delay value being set to a millisecond in the code, the average time between messages was 7 seconds. It can also be observed when the duty cycle limit is enforced, it takes an average of 5.55 Minutes before the E1 can resume sending burst messages.

For clarity, the burst average is the average time between messages before the duty cycle limitations are enforced, the duty cycle average is the time between bursts, and the total average accounts for all message delay times for E1 in this scenario. All values are in minutes and seconds.

This initial DoS test confirms E1 is incapable of initiating a continuous stream of messages without intervention. The next experiment for this section will involve the resetting of the device E1 when the duty cycle limit has been met, doing so will initiate a rejoin sequence to the network which will reset the timer on the device.

6.2 Testing an Attack form 2: Resetting the Device to increase sending rate

In this attack, E1 will be reset at the end of each legal message burst to avoid the wait needed to observe the duty cycle. Doing so will increase the amount of messages that E1 can send within the same 1 hour period. All other parameters, such as payload size, remain the same.

Figure 12 shows the number of messages sent by each device, as well as the amount of dropped messages for the target device E2. This graph shows a much greater success rate in blocking messages from E2 than the previous attack scenario, with 25.8% of all E2 messages dropped. While it would be simple to assume that the increased success rate of this attack is due to the large increase in sent messages by E1, the data gathered from the gateway logs displays an interesting trend pertaining to which messages are blocked and why.

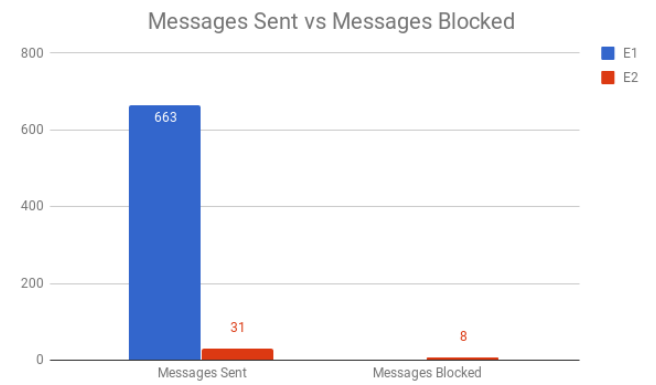

Fig. 12: Messages sent vs blocked over one hour for attack form 2

Direction	Timestamp	DevAddr	FPort	FCnt	AirTime	FCntUp
0	2018-03-28 17:32:49.114	002044BE	1	821	2.138112	821
0	2018-03-28 17:34:08.888	001FCECD	1	2	2.138112	2
0	2018-03-28 17:34:13.490	001FCECD	1	3	2.138112	3
0	2018-03-28 17:34:18.107	001FCECD	1	4	2.138112	4
0	2018-03-28 17:34:22.714	001FCECD	1	5	2.138112	5
0	2018-03-28 17:34:27.315	001FCECD	1	6	2.138112	6
0	2018-03-28 17:34:31.931	001FCECD	1	7	2.138112	7
0	2018-03-28 17:34:43.190	001FCECD	1	0	2.138112	0
1	2018-03-28 17:34:45.190	001FCECD	0	0	1.646592	-
0	2018-03-28 17:34:49.957	001FCECD	1	1	2.138112	1
1	2018-03-28 17:34:51.957	001FCECD	0	1	0.205824	-
0	2018-03-28 17:34:54.928	001FCECD	1	2	2.138112	2

Fig. 13: Message data sample from attack form 2

Figure 13 displays a sample of data from the gateway log during the experiment. From the preliminary calculations performed in section 5.3, it's known the device E2 (DevAddr 002044BE) sends data on average every 1.59.64 minutes. With this in mind, the expected delivery time of message 822 (FCnt) in the figure above would be around 17:34:48.754, due to the last recorded message from E2 being received at 17:32:49.114.

Instead, every time E1 is reset, 2 uplink and 2 downlink messages are sent to and from the device in relatively quick succession. It's these messages that have taken the slot in which message number 822 from E2 would have occupied. This trend is the same for all 8 dropped messages in this attack scenario, with the most frequently blocked messages being in between the first downlink and the subsequent uplink. This seems to indicate the initial data down and up messages that are exchanged at the start of the network connection of a node are the most effective at blocking communications from other devices.

With such useful time and sequence based information gleaned early on, the first attack launched to conduct continuous jamming to lead to DoS, presented an interesting finding and an unexpected results. The first continuous jamming attack didn't manage to block any of the messages sent by the target device E2. The reason for this poor performance was due to the attacker device E1 exceeding the duty cycle limitation for the 868Mhz ISM band and not able to understand how the other devices generated the packets. Subsequent research found the duty cycle of the 868Mhz ISM band is 1%, which results in a limited maximum transmission time of 36 seconds per hour, meaning the device can only "speak" for 36 seconds out of every operational hour [23]. Since both SODAQ ExpLoRer boards utilise the LoRaWAN certified RN2483 transceiver module*, there's no way to change the duty cycle limit on this particular hardware as the integrated LoRaWAN stack prevents a user from performing any action that violates the specification.

This posed an issue, as to perform a continuous jamming attack, because messages had to be streamed continuously to have any

*`https://support.sodaq.com/Boards/ExpLoRer/`

chance of blocking the target messages from E2. To overcome this in the second form of attack, the device was reset manually every time it reached the duty cycle limit for that session, as this was discovered as one way of overcoming the hard-coded duty cycle limitation. This however wasn't an ideal method of attack, as millisecond level accuracy cannot be guaranteed when limited to human reaction times.

Strategically resetting the attacker device allowed for the scenario to mostly ignore the duty cycle, and successfully block 8 messages from E2. While messages were successfully blocked for this attack, the number was still too low to be considered an effective means of DoS attack. Despite this, useful data was still gathered from this attack which formed the basis of the logic for the targeted DoS attack. It was found that all 8 of the successfully blocked messages in this attack fell between the first downlink message and the second uplink message destined for/from E1 upon every reset, meaning that these messages were the likely ones able to consistently block target messages.

6.3 Targeted Jamming for Denial of Service

From the testing carried out in the continuous DoS attacks, it was discovered it takes 8.1 seconds from E1 device restart, to the first uplink message being logged at the gateway in this particular scenario (6 seconds for restart operation + 2.1 seconds message airtime). This information coupled with the airtime of the first downlink message, and the airtime of the subsequent uplink message creates the basis for calculating the ideal time to initiate a restart on E1 to perform a more targeted DoS attack.

The values in Figure 13 show that the airtime for the first downlink message is 1.6 seconds, whereas the uplink after that displays as 2.1 seconds. By combining the total time from device restart to first uplink ($First_{up}$), the airtime of the first downlink message ($First_{down}$), and the airtime of the second uplink message ($Second_{up}$), it's possible to calculate an accurate window (AttackWindow) in which to launch the targeted DoS attack. The formula for this is shown below.

$$AttackWindow = First_{up} + First_{down} + Second_{up} \quad (3)$$

Using the values in Figure 13, the attack window is calculated as (8.1 + 1.6 + 2.1) = 11.8. With the attack window calculated, the formula from Section 5.3 used to gather the predicted target message time can be incorporated to figure out the exact time to initiate a restart on E1, so that the expected message from E2 occurs at the same time at the initial downlink/uplink messages from E1. The process for working out the attack time is shown below.

$$AttackTime = PredictedTime - AttackWindow \quad (4)$$

With the attack time calculated, this allows the attacker to perform a targeted DoS attack with high accuracy. The results from an attack which utilised these formulas are shown in Figure 14.

As the Figure 14 shows, all messages that E2 sent were blocked by the targeted messages from E1, producing a perfect Denial of Service. The implications of these results are worrying for LoRaWAN device owners, as anyone with a capable device and gateway may collect the data needed to choreograph such an attack.

These results show that using the defined formula, it is not only possible to block every message from the target device, but also allows the attacker to have granularity in being able to choose precisely which messages they wish to block at any time.

The formulas proved effective, as all 31 messages sent by E2 in the space of an hour were able to be blocked. These results not only confirm the first downlink and subsequent uplink messages play an important role in these attacks, but also that there are sufficient flaws

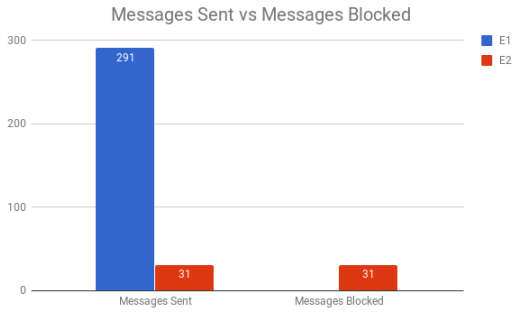

Fig. 14: Messages Sent vs Blocked over 1 Hour for Targeted Attack

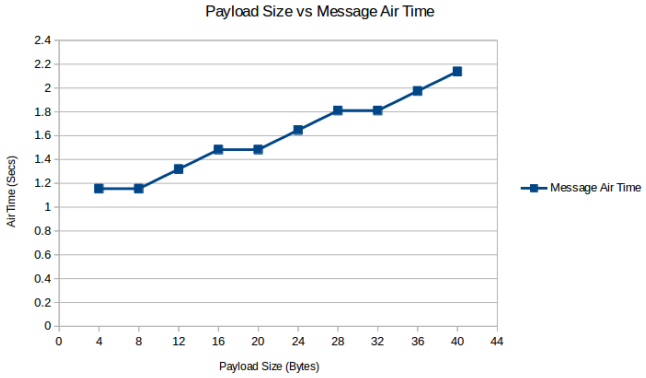

Fig. 15: Message payload size vs air time for E2

in the design of the LoRaWAN protocol. With the gateway unable to receive messages from any other device while it's transmitting to the attacker, it shows this method of calculating the attack times to ensure the initial downlink and uplink messages coincide with the arrival of any legitimate messages can be used to great effect.

It should be noted the first targeted DoS attack had both the target and attacker using payloads of 40 bytes (39 bytes with null terminating byte). Having a larger payload size for both devices increased the air time of all messages, which made the attack much easier to carry out. Having more air time for messages means they take longer to transmit, this way, if it takes the target longer to transmit a message to the gateway, it increases the chances that the communication will be cut off by the arrival of the first downlink message for the attacker. This gives a greater margin of error for the attack, as the attacker does not have to be as precise as if the message sizes differed dramatically.

Since this was a more controlled lab environment, messages were uniform, and devices were stationary and within line of sight of the rogue gateway, which is identified as being one of the larger weaknesses with this study and implementation of attacks. In production environments, payloads are more likely to often differ, devices may be moving/mobile, and may be considerable distances away from gateways. All these factors would play a major role in affecting the attack scenario and present new challenges to an attacker wishing to target these devices.

6.4 Targeted Jamming with Variable Payload Size

So far, both devices have had a payload size of 40 bytes. Having a more variable payload size changes the airtime of the messages as shown in Figure 15. This change in message airtime is likely to have a distinct change in the attack window needed to perform a successful DoS attack. Despite this, the same formula used in all previous experiments should still result in accurate and consistent results.

Attacker Total Reset/First Message	Attacker First Downlink Air Time	Attacker Second Uplink Air Time	Target Average Message Interval
00:00:07.348	00:00:01.640	00:00:01.160	00:01:58.997

Fig. 16: Selected values for 4 byte attack

As Figure 15 shows, the change from a 40 byte payload to a 4 byte payload almost halves the message air time. Less message air time means a shorter window of attack for the attacking device, which could lead to more messages being able to slip through. Using the formulas defined in the previous sections, it is still possible to calculate the attack times for the 4 byte messages from E2. Figure 16 shows the air time values for E1, and the average time between messages for E2 (analysed from 100 messages). The total time from E1 reset to the first message being received is also shown.

When the payload of both the attacker and target device was reduced to 4 bytes (3 bytes with a null terminating byte) the airtime for a 4 byte payloads was approximately 1 second less compared to when the payload was 40 byte. Despite the relatively small difference, the results backed up the theory about the impact of the payload size in Section 4.7, with the 4 byte payload targeted DoS only managing to block 16 out of the total 31 messages from E2. Results showed that not only did the 1 second less air time make a huge difference when attempting to block messages, but also that the predicted message delivery times had to be constantly recalculated throughout the attack, as they were consistently 1 to 3 seconds off. Such results are in stark contrast to those found in the previous section, as one would perhaps expect messages with a much lower payload size to vary a lot less than those with a 40 byte payload.

While it was predicted the smaller payload size would create an impact on the results, it was never theorised that it would have this type of impact. However, the attack could still be deemed as a success, as over 50% of the messages were able to be blocked, but the amount still able to be transmitted is a much higher number than could ever have been guessed. A potential remedy to this would be to collect a far larger sample size of messages to see if the average message interval time can be averaged out to a greater accuracy.

Out of all the attacks and experiments, these results were the most illuminating, and shows the weaknesses of prediction model used, as well as the interpretation of some results. The attack is more efficient when there's greater message airtime because interjecting data to cause error by one device upon another is higher when the data size is larger. An element of human error must be factored into these results as well, since the devices relied on human interventions to reset the devices, which is unsuited to scenarios that require millisecond accuracy.

It's possible that with a more machine perfect, and fully automated attack system that could continuously read the last message arrival time and update the attack times accordingly, that the results for lower payload messages, like the ones experimented with in the previous section, would exhibit a greater amount of messages blocked.

As the table 16 shows, values are slightly lower here than in the other sections due to the lower payload size. Using these values, it's possible to attempt another targeted DoS attack to see whether the reduced payload sizes/air times have any real effect on the results. The results from this attack are shown below in Figure 17. Both devices are utilising a payload size of 4 bytes in this scenario.

Despite this, the attack still managed to block 51.6% of the messages sent by E2. This shows the attack was still mostly successful, but it's a far cry from the results seen in Section 6.3. Because of this, it's important to design your LoRaWAN IoT solution to be resilient against such attacks, as the next section attempts to outline.

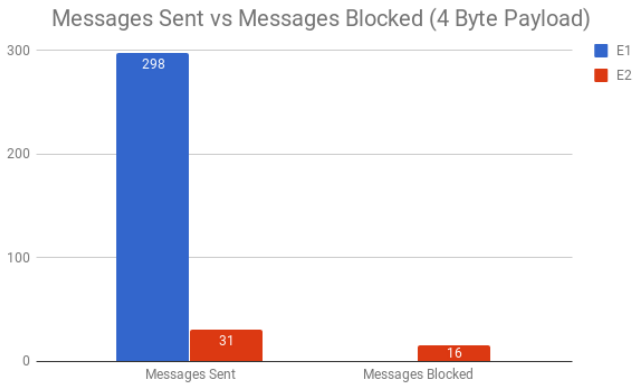

Fig. 17: Messages sent vs blocked over one hour for targeted attack (4 byte payload)

Predicted Message Delivery Time (MM:SS.ms)	Actual Message Delivery Time (MM:SS.ms)	Difference (SS.ms)
21:48.520	BLOCKED	N/A
23.47.517	23:46.446	01.071
25:46.514	25:44.918	01.596
27:45.511	BLOCKED	N/A
29:44.508	29:41.850	02.658
31:43.505	31:40.310	03.195
33:42.502	BLOCKED	N/A
35:41.499	BLOCKED	N/A
37:40.496	37:35.689	04.807
39:39.493	BLOCKED	N/A

Fig. 18: Predicted delivery times vs actual delivery times during 4 byte attack

7 Reflection on Success or Failure of Attacks and Mitigation Technique

As has been proven in the previous sections, it's possible to effectively block messages by analysing and averaging trends observed in the logs of the rogue gateway. The targeted DoS attacks have relied on the intervals between target device messages being fairly regular in their transmission, but should the delays between messages be more random and unpredictable in nature, as has already been demonstrated by the results in Section 6.3 and Section 6.4, it's likely that the targeted DoS attack will prove useless.

To test this, the target device E2 has had its code modified so that the delay between messages is a random value up to 4 minutes. With this change, it should be much harder for an attacker to identify trends in message transmission in which to apply the formula to plot out the best attack time. A sample of the collected message times with the random delay value are shown below in Table 19.

The delivery time of packets are different depending on the duty cycle of each communicating devices and when two or more device operates on a same channel and the starting time of the duty cycles are same then a data collision will occur and packet will get loss. However, if the starting time and ending times of duty cycles of two or more devices does not overlap then even if the devices use the same channel, data collision will not occur, thus no loss. Thus no two devices should transmit at the same time and utilize overlapping units in the same channel to deliver the packets successfully. As per things connected policy in "The Things Network" to maintain fair access, uplink airtime for a device is only 30 seconds per day and the downlink messages to 10 messages per day. So, the aim is to understand when are those times when a device is active. If random access attempt is used, then it becomes impossible to conduct a predictive jamming model to block packets. If a device uses multi-channel for

Message Arrival Time	Difference From Last Msg
17:08:00.156	00:01:33.602
17:08:14.497	00:00:14.341
17:12:10.329	00:03:55.832
17:15:38.513	00:03:28.184
17:19:04.577	00:03:26.064
17:19:41.334	00:00:36.757
17:20:18.434	00:00:37.100
17:22:24.906	00:02:06.472
17:24:27.175	00:02:02.269
17:28:30.954	00:04:03.779
17:31:57.559	00:03:26.605

Fig. 19: Sample of 10 messages with random delay

transmitting data in such a way that first data uses channel 1 and the following next data uses different channel then, it will be very hard to blocked or jammed all the data because the jammer needs to know pattern of channel hopping and the respective slots used in each channel. However, since the header information about the source and the destination are generally not encrypted, it will not be hard to learn the slots and channel used by a device. So, conducting a predictive jamming for a device using multi-channel is also possible.

Given the more erratic nature of the message delay times, it's impossible to calculate an accurate average time between the target messages for use with the attack formula. Due to this, it's also impossible to perform the same type of targeted DoS attack here performed in previous sections. Since a 1-3 second message interval difference was enough to skew results in the previous section, such significant and varied delays, as demonstrated here, is more than enough to completely eliminate the precise time window needed to perform these targeted DoS attacks. The interesting fact is that when a predictive jamming is conducted, it is not even necessary to conduct a jamming for the whole data length, because jamming for any fraction of the data length will lead to erroneous data and dropping of the packet by the gateway. When a predictive based jamming is used, it may be hard to detect if a jammer is on action, however, continuous generation of erroneous packets can lead to a suspicion of a jamming activity by the gateway. So, a mitigation technique could be an observation by the gateway to check the rate of loss of packet due to abnormal erroneous nature.

Due to the unique way in which LoRaWAN operates, and LPWANs in general, the typical countermeasures used to stop DoS attacks on IP networks can't be applied here. Because of this, measures such as the random delay demonstrated above have to be implemented as a proof of concept at the least, even if the end result turns out to not be particularly practical in a production environment. Further discussions and conclusions on all the results presented in this report so far will be discussed in detail in the following sections.

8 Conclusions

IoT technology is adopted by different sector including medical (heath monitoring, device monitoring and management), smart house/building, smart city and emergency systems to mention few areas. The main reasons of adopting such technology is due to ease in deployment, flexibility, ease in data collection and so on. However, security is the area which is least addressed in IoT technology because the end devices are low powered and have low computation and memory power, on the other hand introducing security dimension leads to complexity in design, needs high computation power and compromised battery life which make it prone to security attacks. In this work, it is found that in LoRaWAN, resetting of a device leads to higher data delivery rate and increases the chances of blocking the medium access rights of neighbour device. By understanding the Airtime and uptime of a device E2, an attack window can be created to perform a predictive jamming attack using actual data by other device E1 to increase the chances of blocking packet delivery of E2. It is also found that when the payload size of a packet is small, the chances of blocking packets reduces. In order to conduct a successful attack, it is vital to understand the payload of a packet, duty cycle time and its duration. However, when the duty cycle time and duration of the slot occupancy changes with a varying payload size of a packet makes it hard to make an accurate prediction to block packets. It will make it harder to attack if different devices operates in different channels.

In future, we intend to study the impact on prediction model when the number of devices in the network increases and it will be interesting to study the prediction model when medium access pattern is random and duty cycle and utilization of the slots changes. Moreover, in future a predictive model for jamming a device using multi-channel will also be designed.

9 References

1 Index, C.V.N.: 'Global mobile data traffic forecast update, 2016–2021 white paper', *Cisco: San Jose, CA, USA*, 2017,

2 Kim, J., Lee, J., Kim, J., Yun, J.: 'M2M service platforms: Survey, issues, and enabling technologies', *IEEE Communications Surveys & Tutorials*, 2013, **16**, (1), pp. 61–76

3 Meulen, R.v.d.: 'Gartner says 8.4 billion connected "Things" will be in use in 2017 up 31 percent from 2016', *Gartner Letzte Aktualisierung*, 2017, **7**, pp. 2017

4 Da.Xu, L., He, W., Li, S.: 'Internet of things in industries: A survey', *IEEE Transactions on industrial informatics*, 2014, **10**, (4), pp. 2233–2243

5 Tanczer, L., Brass, I., Elsden, M., Carr, M., Blackstock, J.J.: 'The United Kingdom's Emerging Internet of Things (IoT) Policy Landscape', *Tanczer, LM, Brass, I, Elsden, M, Carr, M, & Blackstock, J(2019) The United Kingdoms Emerging Internet of Things (IoT) Policy Landscape In R Ellis & V Mohan (Eds), Rewired: Cybersecurity Governance*, 2019, pp. 37–56

6 Talwana, J.C., Hua, H.J. 'Smart world of internet of things (iot) and its security concerns'. In: 2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData). (IEEE, 2016. pp. 240–245

7 Iskhakov, S., Meshcheryakov, R., Iskhakova, A., Bondarchuk, S. 'Analysis of vulnerabilities in low-power wide-area networks by example of the lorawan'. In: IV International research conference" Information technologies in Science, Management, Social sphere and Medicine"(ITSMSSM 2017). (Atlantis Press, 2017.

8 Moganedi, S., Mtsweni, J. 'Beyond the convenience of the internet of things: Security and privacy concerns'. In: IST-Africa Week Conference. (IEEE, 2017. pp. 1–10

9 Mansfield.Devine, S.: 'Weaponising the internet of things', *Network Security*, 2017, **2017**, (10), pp. 13–19

10 Constantin, L.: 'Hackers found 47 new vulnerabilities in 23 iot devices at def con', *CSO Website*, 2016,

11 Kolias, C., Kambourakis, G., Stavrou, A., Voas, J.: 'Ddos in the iot: Mirai and other botnets', *Computer*, 2017, **50**, (7), pp. 80–84

12 Teng, C.C., Gong, J.W., Wang, Y.S., Chuang, C.P., Chen, M.C. 'Firmware over the air for home cybersecurity in the internet of things'. In: 2017 19th Asia-Pacific Network Operations and Management Symposium (APNOMS). (IEEE, 2017. pp. 123–128

13 Khan, M.A., Salah, K.: 'IoT security: Review, blockchain solutions, and open challenges', *Future Generation Computer Systems*, 2018, **82**, pp. 395–411

14 Grammatikis, P.I.R., Sarigiannidis, P.G., Moscholios, I.D.: 'Securing the internet of things: challenges, threats and solutions', *Internet of Things*, 2019, **5**, pp. 41–70

15 Tomasin, S., Zulian, S., Vangelista, L. 'Security analysis of lorawan join procedure for internet of things networks'. In: 2017 IEEE Wireless Communications and Networking Conference Workshops (WCNCW). (IEEE, 2017. pp. 1–6

16 Eldefrawy, M., Butun, I., Pereira, N., Gidlund, M.: 'Formal security analysis of LoRaWAN', *Computer Networks*, 2019, **148**, pp. 328–339

17 Alharam, A.K., El.madany, W. 'Complexity of cyber security architecture for iot healthcare industry: A comparative study'. In: 2017 5th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW). (IEEE, 2017. pp. 246–250

18 Bui, D.H., Puschini, D., Bacles.Min, S., Beigné, E., Tran, X.T.: 'AES datapath optimization strategies for low-power low-energy multisecurity-level internet-of-things applications', *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2017, **25**, (12), pp. 3281–3290

19 Tsai, K.L., Huang, Y.L., Leu, F.Y., You, I., Huang, Y.L., Tsai, C.H.: 'AES-128 based secure low power communication for LoRaWAN IoT environments', *IEEE Access*, 2018, **6**, pp. 45325–45334

20 Tsai, K.L., Leu, F.Y., Chang, S.W., Lin, J.Y., Luo, H.T. 'A LoRaWAN Based Energy Efficient Data Encryption Method'. In: International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing. (Springer, 2019. pp. 493–502

21 You, I., Kwon, S., Choudhary, G., Sharma, V., Seo, J.: 'An enhanced LoRaWAN security protocol for privacy preservation in IoT with a case study on a smart factory-enabled parking system', *Sensors*, 2018, **18**, (6), pp. 1888

22 Bardyn, J.P., Melly, T., Seller, O., Sornin, N. 'IoT: The era of LPWAN is starting now'. In: ESSCIRC Conference 2016: 42nd European Solid-State Circuits Conference. (IEEE, 2016. pp. 25–30

23 Adelantado, F., Vilajosana, X., Tuset.Peiro, P., Martinez, B., Melia.Segui, J., Watteyne, T.: 'Understanding the limits of LoRaWAN', *IEEE Communications magazine*, 2017, **55**, (9), pp. 34–40

24 de Carvalho.Silva, J., Rodrigues, J.J., Alberti, A.M., Solic, P., Aquino, A.L. 'LoRaWAN-A low power WAN protocol for Internet of Things: A review and opportunities'. In: 2017 2nd International Multidisciplinary Conference on Computer and Energy Science (SpliTech). (IEEE, 2017. pp. 1–6

25 LoRa.Alliance. 'LoRaWAN What is it? A Technical Overview of LoRa and LoRaWAN LoRa Alliance'. (, 2015

26 Na, S., Hwang, D., Shin, W., Kim, K.H. 'Scenario and countermeasure for replay attack using join request messages in LoRaWAN'. In: 2017 International Conference on Information Networking (ICOIN). (IEEE, 2017. pp. 718–720

27 Miller, R.: 'LoRa security: Building a secure LoRa solution', *MWR Labs Whitepaper*, 2016,

28 'LoRaWAN Security'. (, . [Accessed: 2018-03-18]. `https://www.thethingsnetwork.org/docs/lorawan/security.html`

29 Lee, J., Hwang, D., Park, J., Kim, K.H. 'Risk analysis and countermeasure for bit-flipping attack in lorawan'. In: 2017 International Conference on Information Networking (ICOIN). (IEEE, 2017. pp. 549–551

30 Aras, E., Small, N., Ramachandran, G.S., Delbruel, S., Joosen, W., Hughes, D. 'Selective jamming of LoRaWAN using commodity hardware'. In: Proceedings of the 14th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services. (ACM, 2017. pp. 363–372