MWR Labs Whitepaper

LoRa Security
Building a Secure LoRa Solution

Robert Miller

1. Introduction to this paper

This paper aims to provide independent analysis and guidance around the security of the Long Rang (LoRa) solution and its Long Range Wide Area Network (LoRaWAN) protocol. It is aimed at organisations thinking of using, or actively developing LoRa solutions and should provide readers with clear guidance about how LoRa secures data as well as its limitations that must be considered by developers and users.

The LoRaWAN protocol like many of its rivals offers encryption and secure methods of provisioning end devices (Nodes). However these features should not be blindly trusted by developers and users as they do not defend every possible attack against their solution, and their effectiveness will be governed by the developer's implementation.

Given the wide range of applications using LoRa, many attacks discussed in this paper may not be realistic or even possible for a particular solution. It is important that this guide is not taken as a complete guide for your solution; ultimately there may be attacks unique to your systems. Instead the topics discussed here should be used as guide to help you consider likely attacks against your solution.

Contents

2. Introduction to LoRa

LoRa is a Low Power Wide-Area Network (LPWAN) solution intended for systems that require the ability to send and receive low amounts of data over a range of many kilometres without high power costs. It uses the 868MHz and 900MHz ISM bands and is able to transmit over several kilometres depending on environment. LoRa is a spread spectrum solution which uses wide bandwidth to help protect against deliberate interference or environmental noise. According to LoRa's documentation, the network protocol used by LoRa (LoRaWAN), is capable of providing data rates from between 0.3kbps to 50kbps which varies based on required range and interference.

2.1 LoRa Network Components

The external assets of LoRa solutions are made up of Nodes and Gateways which communicate with a Network Server. Nodes are used to measure and sometimes to remotely control external systems. They are typically low powered and communicate wirelessly with one or many gateways. A Node is normally formed of a LoRa transceiver which is managed by a microcontroller. The microcontroller can send management commands to the transceiver to configure LoRa network settings, or to send and receive application data which the transceiver is responsible for delivering to the Network Server via the Gateways. Although Nodes can be listening at all times, it is standard for the Node to work in a "call then listen" configuration, whereby the Node will send data to the Network Server and then have short windows afterwards where it listens for data coming back from the Network Server.

Gateways are fewer in number, and transfer data from the Nodes back to the Network Server using standard IP connections. A LoRa solution therefore follows a "star of star" topology, where multiple Nodes talk to one or more gateways, which in turn talk back to a single Network Server. Gateways perform no security functionality themselves, but merely act as a conduit to relay data between Nodes and Network Server.

The Network Server is not so well defined but represents the edge of the systems that would store and parse the data sent from the Nodes. In several systems already deployed the Network Server is an Internet facing web service which the Gateways can connect to using for instance cellular networks.

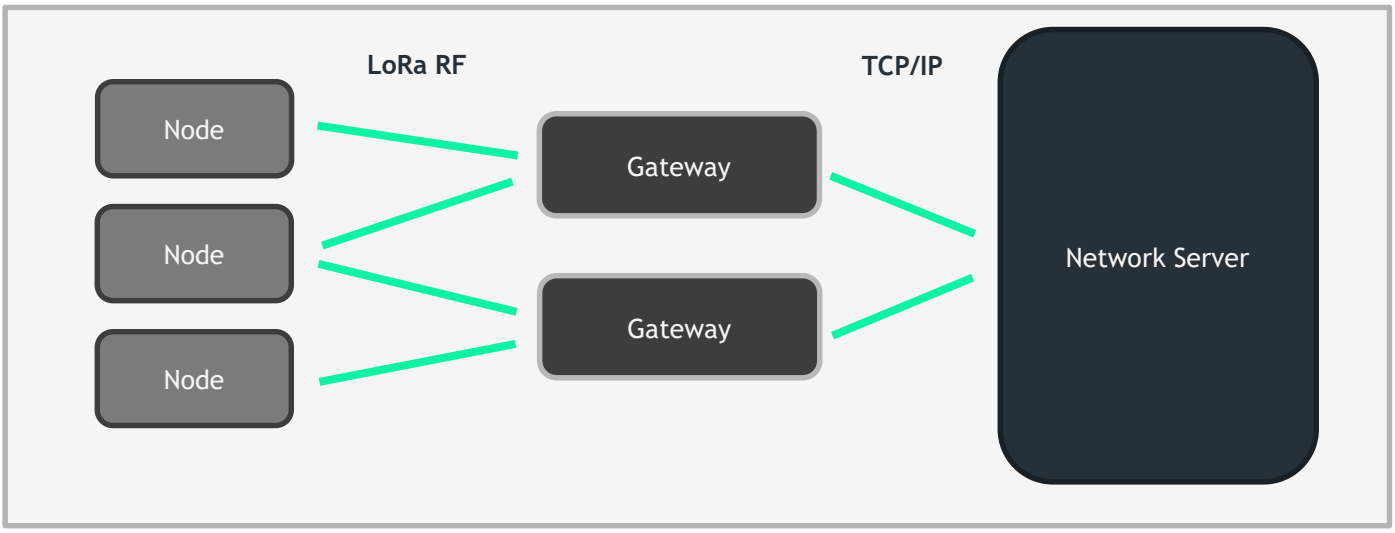

Diagram 1: A high level diagram of the "star of star" LoRa network topology

2.2 Network Stack

The Network stack described here is defined by the LoRaWAN specification:

Radio PHY layer

Preamble	PHDR	PHDR_CRC	PHYPayload	CRC

All LoRaWAN messages have a PHY layer containing a preamble (8 bytes of 0x34 for EU863-870MHZ ISM Band), plus a header and payload, each with CRCs.

PHYPayload

MHDR	MAC Payload	MIC

The PHYPayload starts with a MAC header followed by the MAC Payload and an integrity check value, which is covered in detail in Section 3 of this document. The MAC Header and Payload contain both the user's data, plus header information (such as the type of message being sent) and LoRa version information.

MAC Payload

FHDR	FPort	FRMPayload

The MAC payload contains a Frame-Header (holding the source and destination addresses plus frame counter), a Frame Port and the Frame Payload which holds application data.

The Frame Port is used to determine if the message is containing only MAC commands (where it is set to 0), or application specific data (where it should be set to the number for the relevant application).

3. Security features offered by LoRa

The LoRaWAN protocol provides both signing and encryption for parts of LoRaWAN packets. These are performed using symmetric keys known both to the Node and to the Network Server (and potentially to Application Servers located behind the Network server depending on requirements) and are distributed in one of two ways depending on how a Node joins the network

3.1 Joining a Node to a LoRa network

3.1.1 Over-The-Air-Activation (OTAA)

The first method by which Nodes are allowed to join a LoRa network is through OTAA. Here each Node is deployed with a unique 128-bit app key (AppKey) which is used when the Node sends a join-request message. The message is not encrypted, but is signed using this AppKey.

The Node sends the join-request message including its unique AppEUI and DevEUI values plus a DevNonce which should be a randomly generated two byte value. The AppEUI should be unique to the owner of the device. The DevEUI should be a globally unique identifier for the device.

These three values are signed with a 4 byte MIC which is produced using the following calculation:

```
mac=aes128_cmac(AppKey, MHDR | AppEUI | DevEUI | DevNonce)
MIC = mac[0..3]
```

The server should check the values and then re-calculate the MIC with the AppKey. If valid, the server may respond with a join-accept message within the receive windows of the Node. The Network server generates its own nonce value (AppNonce) and calculate the Node's two new 128-bit keys: the app session key (AppSKey), and the network session key (NwkSKey). These are calculated based on the values sent to it in the join-request message:

```
NwkSKey = aes128_encrypt(AppKey, 0x01 | AppNonce | NetID | DevNonce | pad16)
AppSKey = aes128_encrypt(AppKey, 0x02 | AppNonce | NetID | DevNonce | pad16)
```

The join-accept reply includes an AppNonce, an end-device address (DevAddr) along with configuration data for RF delays (RxDelay) and channels to use (CFList). A MIC is generated using the following calculation:

```
mac = aes128_cmac(AppKey, MHDR | AppNonce | NetID | DevAddr | RFU | RxDelay | CFList)
MIC = mac[0..3]
```

This data is sent back using the AppKey as an encryption key. In this way, the Node can use the AppKey to decrypt the data and then calculate AppSKey and NwkSKey key using the AppNonce.

3.1.2 Activation by Personalisation (ABP)

ABP differs from OTAA as the Nodes are shipped with the DevAddr and both session keys (NwkSKey and AppSKey), which should be unique to the Node. As the Nodes already have the information and keys they need, they can begin communicating with the Network Server without the need for join messages.

3.2 Protection of data sent over LoRa networks

Once a Node has joined a LoRa network, either through OTAA or ABP, all future messages will be encrypted and signed using a combination of NwkSKey and AppSKey. As these keys are only known by the Network Server and specific Node, there should be no way for another Node, or a man in the middle attack to recover the clear-text data.

3.2.1 Data Encryption

Encryption of messages is performed using AES128 in Counter mode (CTR). If the packet's FPort is set to 0 then the NwkSKey is used, otherwise the AppSKey is used. An important feature of all messages in LoRa is that the counters for sent (FCntUp) and received (FCntDown) messages are maintained by the Node and Network Server, and that these counters never repeat.

For encryption and decryption a keystream (S) is produced as follows:

```
i = 1..k where
k = ceil(len(FRMPayload) / 16)
Ai = (0x01 | (0x00 * 4) | Dir | DevAddr | FCntUp or FCntDown | 0x00 | i)
Si = aes128_encrypt(K,Ai), for i = 1..k
S = S1|S2|..|Sk
```

The keystream includes the FCntUp or FCntDown values, which should mean that the keystream never repeats in the Node's lifetime. The FRMPayload is then XOR'd with the keystream to encrypt or decrypt the data. Other data such as the FPort and FCNTUp are sent unencrypted.

3.2.2 Message Signing

The MAC Payload section of messages are signed to prevent manipulation of the cipher-text, or of other values such as the DevAddr, FCntUp or FCntDown values. The 4 byte Message Integrity Code (MIC) is calculated as follows:

```
Msg = MHDR | FHDR | FPort | FRMPayload
B0 = (0x49 | 4*0x00 | Dir | DevAddr | FCntUp or FCntDown | 0x00 | len(msg) )
mac = aes128_cmac(NwkSKey, B0 | msg)
MIC = mac[0..3]
```

3.3 Class B networks

Class B networks provide the functionality for Network Servers to send messages to Nodes without the need to receive a message from the Node first. Messages can be unicast (where a single Node is messaged), or multicast (where every Node is messaged with a single message). This functionality is achieved by having Nodes listen for messages in specific time windows.

For a Node to use class B functionality, it must synchronise its listening windows with the network. This is achieved through the Gateways producing a Beacon which includes its GPS coordinates and a time reference. These are produced simultaneously by all Gateways in the network.

Once a Node has received a Beacon, it can switch to class B and begin listening for incoming messages in specified listening windows. These windows are referred to as “ping slots”. The Node will send subsequent uplink messages with the Class B bit of the FCTRL field set to 1, letting the server see which Nodes are class B enabled.

The contents of the Beacon Payload sent by Gateways is shown below:

NetID	Time	CRC	GWspecific	CRC

The NetID is a three byte value used by the nodes to identify that it came from Gateways belonging to its network. It therefore needs to be unique to the particular LoRa network, otherwise if two LoRa networks send beacons with the same NetID, there could be erroneous actions taken by the nodes.

Time is a four byte value that represents the time in seconds since 00:00 1 January 1970.

The first CRC is either a one or two byte value that is the CRC-16 of the NetID and Time values.

The GWSpecific value takes the following format:

InfoDesc	Info

This allows the Gateway to send a range of information. The following values are listed in the LoRaWAN specification:

InfoDesc	Info
0	GPS coordinates of gateway’s first antenna
1	GPS coordinates of gateway’s second antenna
2	GPS coordinates of gateway’s third antenna
3:127	Reserved for Future Use
128:255	Reserved for custom network specific broadcasts

For sending GPS coordinates (InfoDesc 0,1 and 2), the format is 6 bytes where the first three bytes is for latitude, and the latter three bytes for longitude. They are written as 24 bit signed values.

3.3.1 Maintaining Class B Nodes

Class B Nodes may be expected to move and therefore require the Network Server to use different Gateways to contact them. This could be achieved through Nodes sending regular messages uplink messages so that the Network Server can see through which Gateway these messages arrive and update its routing tables accordingly.

Alternatively a more low power option is for the Node to listen for the regular Gateway Beacons and demodulate and inspect the beacon's content. If the strongest beacon contains different coordinates, then the Node should update the Network Server by sending a message which the Network Server can inspect to see through which Gateway the message arrived.

3.3.2 Multicast Messages

Multicast messages allow the Network Server to send a single downlink message to multiple class B Nodes simultaneously. For this to be possible, all recipient Nodes must share the same encryption keys. For this reason multicast messages are considered inherently less secure than unicast messages. They may therefore not include MAC commands.

The LoRaWAN specification does not describe the method for Nodes to attain these shared keys, but presumably the application of each Node would be able to update the LoRa transceiver with the new NwkSkey and AppSkey keys. The specification does mention that shared key distribution could also be achieved during "node personalization", but as the keys are normally based on nonces, and the keys themselves are never transferred, it is unclear how this would work in practice.

4. Attacks against LoRa systems

As can be seen by the provisioning and messaging security, it should be possible to use LoRa solutions securely to protect against man in the middle attacks affecting the confidentiality and integrity of data. LoRa also provides ways for developers to securely add new Nodes of their choosing to their LoRa network.

However other areas are left to the developers, which may lead to security vulnerabilities being introduced into particular Lora instances. This section looks at the responsibilities of the developer and describes attacks that could be performed if such vulnerabilities were introduced.

4.1 Weaknesses in Key Management

The use of symmetric encryption for security means that there must be at least two places where keys are stored. The Nodes and the Network Server.

4.1.1 Key Management in Nodes

For Nodes, they should only be storing keys that they require. It is likely given the range of hardware attacks available that an attacker could recover the AppKey, NwkSKey and AppSKey from a Node using for example side channel analysis. This attack uses the variations in power consumption or EM emissions from the transceiver during AES encryption to determine the key that must have been used. As an attacker with this key would be able to produce correctly signed and encrypted messages, the data coming from individual Nodes should therefore be assumed to be potentially untrustworthy.

The LoRa specification requires Nodes use keys that are unique to that particular device, and are random. If keys were to be shared cross-device, then it would be possible for an attacker to use a stolen key to intercept or spoof traffic from any other Node.

4.1.2 Key Management by Network Servers

The Network Server, and its supporting systems are ultimately responsible for the management of network and application keys. This means that the implementation of generating and storing keys could introduce vulnerabilities that undermine the security offered by LoRa.

Keys must be stored in a way that means although they are accessible to the Network Server for decrypting and verifying the signatures of messages, they are not open to be read or altered by unauthorised parties.

4.1.3 Key Usage by Network Servers

Another potential issue is how the Network Server performs its decryption and signature checking. Aside from introducing vulnerabilities in AES implementation, care should also be taken to perform checking the MIC signature before decryption of the message. If the FRMPayload is acted upon before the MIC is checked, then it would be possible for an attacker who had basic knowledge of the payload structure to manipulate the encrypted FRMPayload by flipping bits in the message. This could be performed without knowledge of the keys.

For example if the Node was sending the following JSON data as its payload:

```
{“ID”:”34”,”Temp”:”24”}
```

The cipher text may look like

```
750f7f9b6366b4228172fb36fdbe51a3dcc1a85d463d70
```

Because this is generated by XOR-ing the plain text with the keystream, it is possible to target specific bits to change. For example we would alter the Temp value by changing the bytes representing “24” (highlighted above). By changing 5d to 5a, the following plaintext is produced when decrypted:

```
{"ID":"34","Temp":"54"}
```

Checking the MIC is therefore vital to stop valid messages for a user being generated by an attacker.

4.2 Weaknesses in Key Generation

The LoRaWAN specification explicitly warns developers about generating secure network and application keys:

> *Each device should have a unique set of NwkSKey and AppSKey. Compromising the keys of one device shouldn‘t compromise the security of the communications of other devices.*
>
> *The process to build those keys should be such that the keys cannot be derived in any way from publicly available information (like the Node address for example)*
>
> **–LoRaWAN Specification V1.0**

Key extraction from Node devices is probable given that they will likely be physically outside of controlled environments. It is important therefore that the theft of keys from one Node does not compromise other Nodes in the system.

One potential vulnerability is where Nodes use Activation-By-Personalisation (ABP) for joining, but use keys derived by the Node based on features such as the device address. If this could be worked out through reverse engineering of one Node, then all other communications to any Node would then be compromised.

4.3 Data handling

Devices that are produced by a company but used in remote locations are often considered as being trusted devices. In fact their location puts them at risk of physical attack, ranging from theft through to tampering. Data received from any Node should not be considered safe, and therefore should be sanitised before use. An example of this may be where the Node is sending JSON data as shown in section 4.1.2. A mistake would be to assume that the JSON data is trustworthy as the encryption and signing was valid, and for example to place the results into a database, or use the data to form file names without first sanitising the data.

LoRa Node devices can be compromised in a number of ways. One example is if the LoRa Node used a Transceiver (such as the RN2483), which handles encoding, encrypting and transmitting the LoRa data.

The microcontroller does not know the encryption keys used by the LoRa network. Instead it would send data to the LoRa transceiver module which would encrypt, sign and transmit the data.

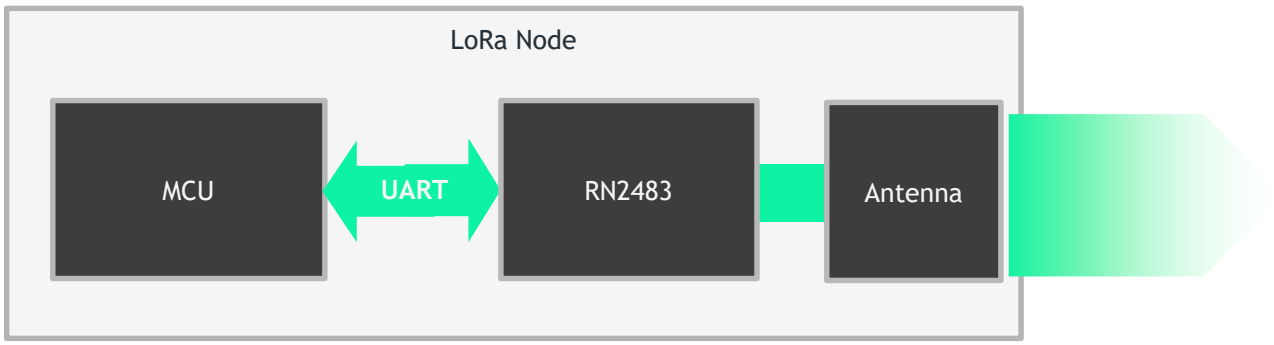

Diagram 2: A typical LoRa Node setup

An attacker with physical access to one of these devices could in theory replace the microcontroller or use the UART pins of the LoRa transceiver to start sending their own messages on behalf of the Node. The attacks would be dependent on the particular system. For example if the LoRa system is sending utility usage information then the attacker could falsify their usage data. Alternatively if the data was being send to include data that was being included within SQL statements by the server side components, then SQL injection could be possible.

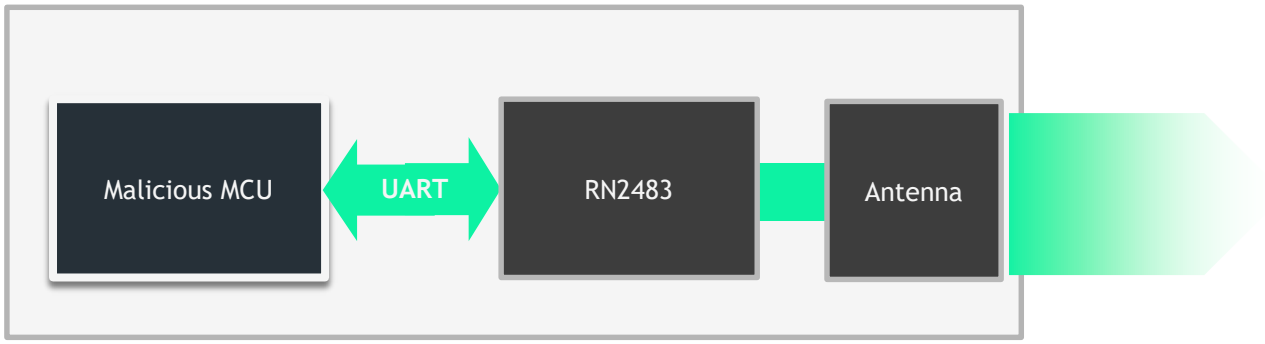

Diagram 3: Compromised LoRa Node with MCU replaced with an attacker programmed MCU

Whitelisting data should be simple for most LoRa solutions as the data being sent back from a Node is likely to be predictable. For example a Node measuring temperature should only send back characters 0-9. Any other characters encountered are an indicator of compromise.

4.4 Gateway Compromise

Gateways are expected to have an IP connection to the network server. For many this will be as simple as the gateway using a 3g dongle allowing Internet connectivity to the Network Server's Internet facing service. These devices may, depending on the service provider, be directly accessible over the Internet. In this case it is important that no services (such as management interfaces like SSH), are enabled, or are secured and can be updated when new security vulnerabilities are discovered.

For others though it may be considered an advantage to route Gateway traffic through to a private network through, for example, a VPN tunnel. In this case the physical security of the gateway should be

considered as an attacker with physical access may well be able to compromise the device and gain network access using the gateway's tunnel.

It would be important to harden the gateway so that tampering would lead to wiping of any access credentials, and also to harden the network that the gateways connect to, to minimize risk following compromise.

4.5 Internet Facing Components

Many solutions have made some components Internet facing, e.g. they can be accessed by anyone who knows the IP address, port and protocol that they use. Some LoRa solutions have made their Network Servers Internet facing so that they can be connected to by the Gateways.

This increases the risk of compromise, as Internet facing services are a common target for hackers. One risk is that gateway traffic could now be forged without the need of a compromised Node and therefore forgo the cost, as well as the bandwidth limitations that this vector causes. A possible attack would be for the MIC of packets to be brute forced (which would take around 2billion attempts to succeed given the MIC's 8 byte key space). Although infeasible over LoRa, a web service could be sent this amount of traffic.

Availability of the Network Server and Key management servers must also be resilient against Denial of Service (DoS) attacks. If the hosts were to become unresponsive, then all Nodes would stop being able to communicate. An example here may be a Network Server that was Internet facing with a web service interface. If an attacker could flood the web service with traffic, then it would stop being able to receive and communicate with the Gateways and ultimately the Nodes.

It is recommended that these components therefore restrict traffic through whitelisting IP addresses to only required systems, require authentication from the gateways and are subject to regular patching.

4.6 Counter management

The Network Server and Nodes are responsible for testing the integrity of the messages they receive, as well maintaining counters for messages (FCntUp and FCntDown).

If the Server or Node is not correctly checking counters then it would enable replay attacks, whereby the same Node message could be recorded and played back to the gateway multiple times. The effect would depend on the application, but could for example in a burglar alarm system be used to replay an "alarm disable" message.

Another issue with counters is that they are intrinsic to the security of LoRa's encryption. Encryption relies on generating a keystream which is then used to XOR with the plain-text to generate a cipher-text. If the same cipher-text was used twice and the attacker knew the plain-text for one of the messages, then they could calculate the plain-text of the other message.

An example of this may be if the solution does not increment its counters, or a Node can be forced to re-join a network, where it may reset its counters. An attacker with knowledge of plain text for one message could then XOR the known message with the cipher-text to recover this keystream. This would let them recover all further messages that used this same keystream. By incrementing the counters, the keystream will always be unique for every message, making this attack ineffective.

4.7 Attacks against class B networks

Class B networks allow for additional message types that in turn may create new opportunities for attackers in addition to those listed in previous sections.

4.7.1 Class B Beacons

Beacons are sent by Gateways which enable Nodes to synchronise their timing for downlink messages, and for Network Servers to know through which Gateway a particular Node can be contacted. These Beacons are not encrypted nor signed, and therefore represent both a source of information and a route to inject malicious data in to a system.

Gateways themselves may store sensitive information, such as APN/VPN configuration data and credentials. Their compromise could allow an attacker to use this data to connect to the Network Server themselves. The Beacons, containing GPS coordinates of each Gateway aid an attacker in locating these devices to carry out attacks listed in section 4.4 of this document.

Beacons can also be generated by an attacker using off the shelf tools. Nodes have no way of knowing whether the beacons it receives are malicious or genuine. If an attacker could generate a beacon that arrived before the official beacons, or at a higher signal strength, then the Node may well choose the malicious Beacon's data to act upon. According to the specification, Beacon data is sent to the application layer, which may risk data parsing vulnerabilities. Alternatively, if the Node uses the Time value in the Beacon, then this may lead to a denial of service state where listening windows for future Beacons and messages from the Network Server are no longer received.

Finally The LoRa specification states that Gateway Beacons allow for "network specific broadcasts". It should be kept in mind during the development of such broadcasts that as neither encryption nor signing is applied to such messages, they could be read or sent by attackers.

4.7.2 Multicast Messages

Multicast messages are used by a Network server to message multiple Class B Nodes simultaneously. For this to be achievable, all receiving Nodes must share the same network and application keys. The exact methodology for key distribution is left to the developer.

Multicast messages are considered less secure, as the compromise of the shared key from a Node would then allow an attacker to communicate to and from multiple Nodes and the Network Server. The attacker would likely use methods listed in section 4.1.1 of this report to attain the key out of a single Node.

Given that the specification states that MAC commands should not be sent using multicast, it is expected that a Node would be able to update its keys when expecting to receive a multicast message, and then back to the secure standard keys for all other operations. In this way the Network Server is still able to manage the network using MAC commands over unicast messages, and the resultant fallout of a multicast key compromise would then be limited to an attacker able to produce multicast messages.

5. Producing a Secure LoRa Solution

As can be seen from the list of attacks, weaknesses in the design and the implementation could lead to a LoRa system that was vulnerable to attack. However it is possible to build a LoRa solution that not only prevent but also can detect and respond to cyber-attack.

5.1 Prevention

Prevention of attacks requires not only the proper implementation of the LoRa standards on all components, but also the application of good security practices of all components that form the solution. For example producing good random keys for Nodes could be undermined if they are to be stored on a server that can be accessed by hundreds of employees. A "cryptolocker" style attack emanating from an employee's laptop that encrypts the App and Network keys used by the LoRa solution would render an entire system disabled.

Detailing security best practices for corporate IT networks is clearly a subject that is outside of the scope of this paper. However it would be a mistake to think only about LoRa solutions in terms of their embedded and wireless components. Generally the principle of least privilege should be applied. For example, if a Gateway does not need access to dozens of hosts through a VPN, then this should be reduced and then tested to prove that this is the case.

Another common mistake is to look at components in isolation. The customers may access their data through a web application which should be reviewed, but if other applications are running on the same host then their security would impact the LoRa solution's.

Prevention should also cover the reduction in damage for an attack. A good example of this is the LoRa standard's own advice to make sure that NwkSKeys and AppSKeys are unique per device. Although this does not prevent the compromise of one device's data, it does prevent the compromise affecting other Nodes.

5.1.1 Testing areas:

Node device review

This should look at how data is stored on the Node, what physical protections are in place (such as tamper detection), what interfaces are available and how the device will be updated.

Node RF review

This would look at the LoRa RF handling by Nodes, both in terms of how the protocol is parsed, how the Node handles authentication and deals with attacks such as replayed or spoofed messages.

Gateway review

The gateway may also be out of physical control, so should be reviewed from a local perspective by for example assessing the available interfaces and how it stores information such as configuration and credentials for connecting to the Network Server. Testing should also inspect how the gateway is parsing LoRa data and how updates are sent to gateways. The Gateway's communications with the Network Server should be reviewed to inspect the confidentiality and integrity of this channel

Network Server Review

The network server testing will vary depending on how it can be accessed by the gateway and internally for management. The interfaces should be reviewed to inspect how authentication is handled and data is parsed. The host's configuration should be reviewed and patch management investigated.

Backend/backhaul network review

The network managing the LoRa system from the Network Server on should be reviewed to make sure that an attacker cannot access the network without authentication and that if they were to gain access to the network, that they cannot access sensitive services or the data (Node data, customer data or key data).

Management systems review

The applications or processes used to manage the LoRa solution should be reviewed to make sure that they can only be accessed by authorised staff, and they correctly control and manage Nodes and Gateways to prevent for example unauthorised Nodes joining the network.

Customer accessibility (Smartphone app/web application)

Public facing systems that will be used by customers or users should be reviewed to make sure they restrict users only accessing data they have permission to view and edit. Platform specific testing such as web application or mobile app will need specialist attention to make sure that customers can safely use the solution.

Rollout and maintenance review

The individual components listed above may contain issues that would be missed by the above tests. For example the process for replacing a faulty Node may introduce vulnerabilities due to the interaction of several components and processes.

5.2 Detection and Response to Attacks

Detection of attacks mentioned in this paper should all be possible, but would require the inspection and reporting of identified attacks to staff. For example, attacks that seek to change the encrypted data (mentioned in 4.1.2 of this report), could be identified when the data is parsed by the back end servers. This would be a clearly unusual incident as normal deviations (caused by for instance RF interference) would have been detected by CRC checks.

Responding to attacks may be more complicated. Although the above scenario would give the responding team a device address, it is possible that the RF message was generated by a source other than the Node identified by the data (such as via a capture/replay attack using a software defined radio). Removing the Node or its keys may therefore remove a legitimate device from the network without stopping the attacker.

Other attacks that originate over server side components would be simpler to detect and respond to due to the relative maturity of intrusion detection solutions available for IT systems. With all IT systems, the monitoring effort should be reduced by first identifying the key assets of the system (for example the LoRa keys, management systems and customer data), minimising the access to these systems, and then

monitoring the few systems that have access to these assets. The hosts used for access can be restricted and audited, and the traffic sent from these systems to servers holding assets can also be monitored for indicators of compromise such as out-of-hours access or unusual requests.

6. Conclusion

LoRa and the LoRaWAN protocol allow secure solutions to be developed that protect the company and the end user from cyber-attacks. It should be clear to all developers of LoRa solutions however, that using LoRa does not guarantee security. Instead they should build LoRa solutions with the potential attacks in mind. Given that LoRa will form part of a complex IT solution means that security vulnerabilities are a likely occurrence during development. Similarly given that LoRa solutions are being used in systems ranging in use from home security through to monitoring and controller infrastructure, attacks and development of exploits against these systems are also likely.

A key area of concern for every solution will be around key storage. It is likely that developers will focus security testing on Nodes, and on customer portals. The key storage represents a single point of failure for an entire solutions as once this server or set of servers is compromised, the entire security around LoRa is undermined and attacker is free to intercept or spoof any message they wish. A simple phishing attack on staff could undermine the most expensive and otherwise well thought out secure LoRa solution.

Secure systems can be developed by understand LoRa's security features, but knowing that they are not a panacea to security. A secure solution can be developed by considering cyber-security at every stage. Knowing the different ways that a LPWAN solution can be attacked allows us to develop a system is built to defend, detect and respond to cyber-attacks. The alternative for many companies is to retro-fit security into their solutions. When a solution has many physical components like LoRa solutions, this process can be prohibitively expensive both in terms of cost and time.

6.1 About the Author

Robert Miller is a security consultant and researcher for MWR InfoSecurity. He heads the company's Operational Technology Security practice which helps clients understand and mitigate the risk of cyber-attacks in the industrial sector.

MWR InfoSecurity is an independent cyber-security consultancy which prioritises security research to help provide relative and innovative solutions to clients.