2021 13th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT) | 978-1-6654-0219-4/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICUMT54235.2021.9631602

A Combined Attack Scenario to Exploit the Join Procedure of LoRaWAN

1st Duc Tran Le
University of Science and Technology – The University of Danang
Danang, Vietnam
letranduc@dut.udn.vn

2rd Truong Duy Dinh
Posts and Telecommunications Institute of Technology
Hanoi, Vietnam
duydt@ptit.edu.vn

3th Van Dai Pham
Dept. of software engineering and computer science
Bonch-Bruevich Saint Petersburg State University of Telecommunications
Saint-Petersburg, Russia
fam.vd@spbgut.ru

4nd Ruslan Kirichek
Dept. of software engineering and computer science
Bonch-Bruevich Saint Petersburg State University of Telecommunications
Saint-Petersburg, Russia
kirichek@sut.ru

5th Egor Filin
Dept. of software engineering and computer science
Bonch-Bruevich Saint-Petersburg State University of Telecommunications
Saint-Petersburg, Russia
filin.ed@mail.ru

6th Alexander Shestakov
Dept. of Intelligent Automation and Control Systems
Bonch-Bruevich Saint-Petersburg State University of Telecommunications
Saint-Petersburg, Russia
alexandr.shestakov01@yandex.ru

***Abstract*—In the LoRaWAN network, end devices must initiate the join operation by sending packets to the Network Server. This process contains several flaws that attackers can use to disrupt network functioning. We will study, analyze, and compare several techniques for exploiting vulnerabilities that are typically conducted during the join procedure in LoRaWAN. Additionally, we propose considering a possible attack scenario, which may occur when the attacker combines some attack techniques to exploit the join procedure.**

***Index Terms*—jamming attack; replay attack; Over-The-Air Activation; LoRaWAN**

I. Introduction

Nowadays, the Internet of Things (IoT) plays a critical role in various spheres of life, including agriculture, industry, and education [1], [2]. IoT enables the flow of data between various sensors, actuators, electrical devices, and network devices. Bluetooth, Wi-Fi, and Zigbee [3], [4] are still utilized in IoT. However, they are limited in cost, energy consumption, and, most importantly, the ability to carry data across great distances. That is why LoRa [5] and LoRaWAN [6] were created. Fig. 1 presents a statistic analyzed by Statista about the number of Low Power Wide Area Networks (LPWAN) connections by technology worldwide from 2017 – 2023. It shows that the portion of LoRa becomes more significant in comparison with other technologies [7]–[9].

With LoRa technology, we can transmit data from a distance up to several kilometres without power amplifier circuits. Additionally, LoRa offers advantages in terms of battery life optimization, ease of deployment, and resistance to interference.

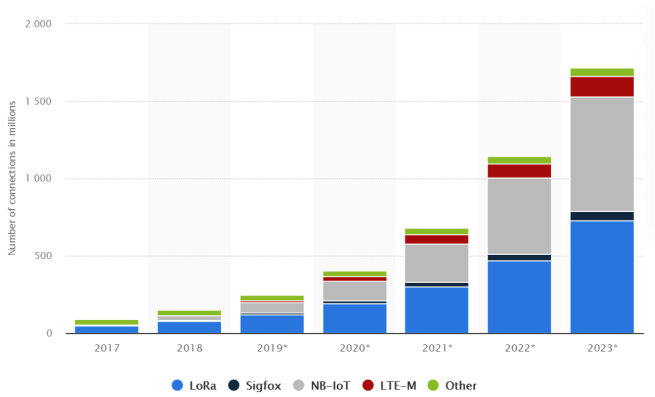

Fig. 1: Number of LPWAN connections by technology worldwide from 2017 – 2023

Apart from the benefits, LoRaWAN has a number of disadvantages, one of which is security concerns. Several studies are now being conducted on this subject. Other researchers' studies on this subject focus exclusively on the attacks and the methods taken to prevent them. Apart from providing an overview of attacks on the LoRaWAN network, this paper compares various attacks and proposes an attack scenario based on combining different forms of attacks. Following that, we compare several known techniques for mitigating the impact of these attacks during the join phase.

The rest of this paper is organized as follows: Section II overviews the security issues in LoRaWAN. In Section III, we shortly explain the join procedure. In Section IV, we analyze the jamming attack, and Section V presents the replay attack.

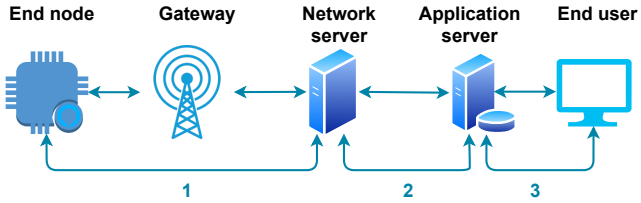

Fig. 2: LoRaWAN network deployment

Section VI presents a combined attack scenario. In Section VII, we compare some existing solutions, and we conclude this paper in Section VIII.

II. Security Issues in LoRaWAN

LoRaWAN employs two distinct layers of security: a network layer and an application layer. The network layer is in charge of validating end device access to the network. By contrast, the application layer protects end-user data from network operators. To increase LoRaWAN security, Advanced Encryption Standard (AES) encryption is used in conjunction with key exchange using the IEEE EUI64 identifier [10]. However, it still includes security flaws that attackers can exploit to gain access to and destroy networks and steal sensitive data [11].

In LoRaWAN, there are three processes of communication and data transmission (Fig. 2): (1) Communication between the end devices and Network Server; (2) Communication between the servers; (3) Communication between the Application Server and the user.

Each procedure has its own set of security flaws [12]. Since processes (2) and (3) utilize other networks such as LTE and Ethernet, security considerations in these processes are unrelated to LoRa Technology. As a result, this paper will focus only on the security implications of the communication process between end devices and Network Servers.

The current version of the LoRaWAN specification significantly improved the protocol's functionality and fixed many previously reported security issues. However, as the analysis in [13] indicates, there are still several security threats. LoRaWAN-based systems are vulnerable to distributed denial of service (DDoS) attacks directed at wireless signal transmissions [14]. Additionally, data transmitted via the LoRaWAN network is not entirely encrypted [15], allowing information leakage or communication between servers to be hacked via man-in-the-middle (MITM) attacks.

The following are the major contributions of this paper:

- analyzing and comparing the most common forms of attacks that occur during the join procedure;
- illustrating a scenario in which an attacker employs multiple attack types;
- Analyzing and comparing a variety of potential strategies for mitigating the effects of these types of attacks.

III. Join Procedure in LoRaWAN

Each end device is manufactured with two distinct root keys: an application key (AppKey) and a network key (NwkKey).

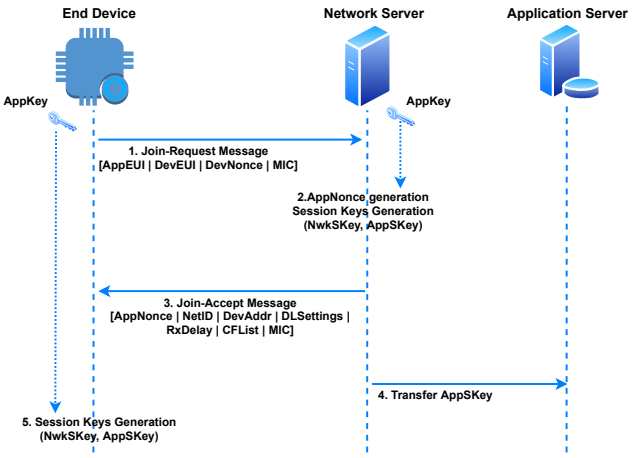

Fig. 3: Join Procedure in LoRaWAN

The session keys (application session key (AppSKey) and network session key (NwkSKey)) are generated using this pair of keys. Each end device has a unique AppSKey. Only the end device and the application server are aware of it. The NwkSKey is unique to each end device and is only known to the end device and the Network Server.

When a new LoRa-end device is added to a LoRa network, it should go through an activation process. Through this process, both session keys are shared between the end device and the Network Server. Currently, LoRa provides the following two types of activation methods: Activation-By-Personalization (ABP) and Over-The-Air Activation (OTAA), called Join Procedure. Because in the ABP method, an end device can belong to a particular LoRa network without performing a join procedure under certain conditions, we will not recall it now. The readers can find more details about ABP in [12].

The join procedure consists of two messages exchanged between the end device and the Network Server, namely join-request and join-accept. The end device sends the join-request message to the Network Server. The Network Server responds to the join-request message with a join-accept message if the end device is permitted to join a network. Fig. 3 depicts the join procedure. **Join Procedure Steps:**

– **Join-Request Message**: It is composed of an end device identifier (**DevEUI**), an application identifier (**AppEUI**), and a 16-bit nonce (**DevNonce**). They are formatted in the IEEE EUI-64 address space standard. The DevNonce is a random sequence number that begins with zero when the device is powered on and increments with each join-request made by the end device. A DevNonce value shall never be used in conjunction with another AppEUI value. The Network Server records the end device's most recent DevNonce value and ignores join-requests if DevNonce is not incremented.

A message integrity check (**MIC**) value of join request, which is calculated by end device and an AppKey, is preshared between the end device and Network Server. It should be noted that the join-request message is not encrypted (for the versions before version 1.0.3). It can be transmitted using any data rate

and following a random frequency hopping sequence across the specified join channels.

– **Authentication and Session Keys Generation**: The Network Server examines whether the end device is permitted to join the network or not to depend on the DevNonce at the time of reception. If the Network Server detects that the DevNonce in the join request has been used earlier, it considers that the message is invalid and the join procedure will fail. If the message is legitimate, the Network Server uses the MIC value to authenticate the end device. The Network Server generates a NwkSKey and an AppSKey if the end device passes the authentication. **AppNonce** is a unique random counter number generated by the Network Server and sent back to the end device. **NetID** is a 24-bit field network identifier to separate addresses of geographically duplicated LoRa networks. The Network Server can freely determine the other bits of NetID. NwkSKey and AppSKey are performed as follows:

$$NwkSKEY = AES128_{encr}(AppKey, \\ 0x01|AppNonce|NetID|DevNonce|pad_{16})$$

$$AppSKey = AES128_{Encr}(AppKey, \\ 0x02|AppNonce|NetID|DevNonce|pad_{16})$$

where pad_{16} – padding data to minimize the packet-sniffing attacks.

– **Join-Accept Message**: A join-accept message contains **AppNonce**, **NetID**, **DevAddr**, **DLSettings**, **RxDelay**, and **CFList**. The **DevAddr** is a 32-bit identifier of the end device within the current network. **DLSettings** contains several values related to the downlink configuration. **RxDelay** is a delay between the transmission and reception process. **CFList** is an optional field that is about channel frequencies. Finally, the whole join-accept message is *encrypted* with the AppKey.

– **Transfer AppSKey**: Because the AppSKey is intended to protect connections between the end device and the Application Server, it should be sent from the Network Server to the Application Server. Because the LoRa specification does not specify when or how AppSKey should be exchanged with the application server, it is implementation-dependent. It may be a critical component and hence should be included in the joining procedure.

– **Session Keys Generation**: The end device decrypts the join-accept message and generates session keys using the retrieved parameters after receiving it.

As can be observed from the examination of LoRaWAN's operation above, LoRaWAN is incompatible with various threats. Indeed, the Network Server may be a web server, and a denial-of-service attack could be conducted to block all communications with this web server. Another attack vector could be based on the properties of AppNonce, in which the end device does not register in the join-accept message (while DevNonce in join-request message shall be registered to avoid replay attack). An attacker can register a fake join-accept message, and when the end device submits another join-request, the attacker can react with that fake registered message. Additionally, we can see that the NwkSKey and AppSKey are generated using AppKey. An inside attacker, such as a network administrator, can totally decrypt and change application data. As a result, the Application Server must have confidence in the Network Server's ability to maintain data integrity. The following section will discuss some common attack types that occur during the join procedure in detail.

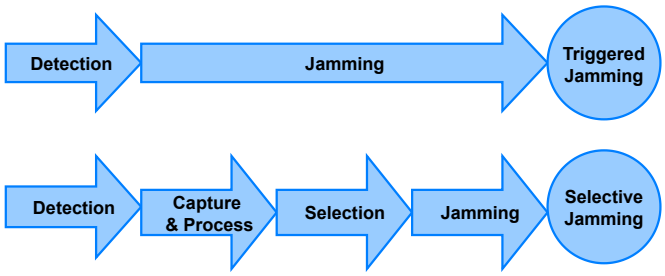

Fig. 4: Jamming attack techniques

IV. Jamming Attacks on Join Procedure

Because LoRaWAN is a wireless network, it cannot be protected against signal jamming attacks. Jamming is a type of denial of service attack that targets wireless channels. Its objective is to obstruct communication between the end device and the Network Server by producing a noise signal near LoRa end devices. Researchers recreated this attack scenario and described it in [14]. We will present two distinct strategies for attack jamming in this paper: triggered jamming and selective jamming. The flow of jamming attacks for each approach is depicted in Fig. 4.

A. Triggered jamming

To avoid concurrent transmissions, LoRa radio modules can scan a given channel to determine whether or not a LoRa transmission is currently in progress. However, attackers might leverage this capacity to identify activity on the channel and deploy a fake LoRa device to initiate a transmission jamming attack. This attack is well-suited for disrupting a large number of gateways, causing widespread disruption of operating applications and the network. It results in an increase in packet loss across the network. This type of attack is simple to execute because no modulation or decoding techniques are required.

B. Selective jamming

Jamming that is triggered will interfere with all devices operating at a given frequency, making it immediately detectable. To evade detection, attackers may use a different sort of jamming called Selective Jamming. Selective jamming interferes with only those devices or communications that have been previously selected. It is difficult to discern between incidental transmissions and deliberate attacks. However, due to the characteristics of LoRaWAN, attackers must perform a series of steps, including (i) detecting a LoRaWAN packet, (ii) initiating the packet's reception, (iii) aborting the packet's reception if the received data triggers the jamming policies;

and (iv) immediately jamming the channel if all configurations are set. Due to the large number of messages processed before jamming, this entire procedure increases the processing time and attack deployment. As a result, the chance of successive jamming using the selective jamming approach is smaller than the probability of triggered jamming after a specific interval.

V. Replay Attacks on Join Procedure

The replay attack is a type of attack in which the identical message transmitted by the end devices is recorded and replayed. As discussed previously, join-request packets are not encrypted (for versions before 1.0.3), allowing attackers to capture the packet's data and perform a replay attack. The attackers may gain legitimate access to the LoRaWAN network via the replay attack. They will transmit malware and engage in other damaging actions within that network, resulting in catastrophic consequences. From version 1.0.3, the join-request message is encrypted and recorded with a four-byte MIC.

Additionally, a DevNonce value is used to prevent a replay attack. However, these nonces are generated by a random-number generator with limited capabilities, such as a limited pool of numbers (each time a number from this pool is randomly selected, the probability of selecting the same number increases with time), which may result in the generator repeating itself after a certain amount of usage time. Additionally, certain types of jamming techniques (described below) can be used to deplete this DevNonce number pool rapidly.

To carry out a replay attack, we need to proceed in three stages: (i) install equipment to sniff packets in the vicinity of the target, (ii) analyze the packets and extract information such as AppEUI, DevEUI, DevNonce, (iii) perform the attack on the Network Server by sending continuous join-request messages. Under this attack, the Network Server will remove or do not respond to the requests from the real end devices. The authors in [16] pointed out that such an attack needs to take $\mathbf{T}_r$ days to take place.

$$\mathbf{T}_r[days] = \frac{\mathbf{N}_D + 1[\mathbf{DevNonce}]}{\mathbf{f}_J[\frac{\mathbf{DevNonce}}{days}]}$$

where $\mathbf{f}_J$ is the number of valid procedures involved every day on each end device, $\mathbf{N}_D$ is the number of DevNonce values previously used and hosted by the Network Server. In most cases, $\mathbf{T}r$ is a huge number, but still, there are exceptions, for example, when the device is reset when participating in the network.

In Tab. I, we compare these two types of attacks in the join procedure.

VI. A Combined Scenario to Exploit the Join Procedure of LoRaWAN

Although selective jamming attacks are challenging to be detected and their success rate is relatively high, the consequences just stop at the level of interference or destruction of the network operations rather than stealing information or spreading the malware. Meanwhile, replay attacks can do

TABLE I: A comparison between the jamming attacks and the replay attacks

Criteria	**Jamming Attacks**		**Replay Attacks**
	Triggered Jamming	***Selective Jamming***	
Consequence	Causing network congestion and denial of service on a specific channel in the vicinity of the interference-causing equipment	Only causes an impact on the selected target	Spreading the malware, causing loss and misleading information
Deployment	Easy to configure and perform. It is suitable when there are many targets	It is more complex than triggered jamming, but it is simpler than a replay attack to perform	It is complex, has higher technical requirements, and need more devices to perform
Success rate	High success rate	The success rate is lower than triggered jamming	The success rate is much lower than jamming attacks, and waiting time can be very long
Detection	Easy to be detected	Challenging to be detected. It is similar to conventional incidents	Difficult to be detected
Popularity	Very popular	Very popular	Relatively common

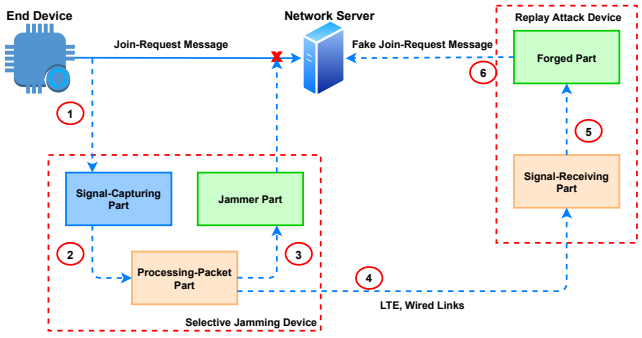

Fig. 5: An assumption on the combined attack

these things, but these attacks have a low success rate due to the rejection mechanism when reusing a DevNonce value. Therefore, if the attacker combines these two attack methods, it will be a highly capable attack option and may cause serious consequences. In the following part, we will present the assumption on the attack based on the combination of the two attack methods mentioned above. This attack is divided into two stages: the jamming stage and the replay stage. Fig. 5 describes these stages. Fig. 4.

Jamming stage: In this stage, the attacker will install a selective jamming device to prevent and capture the join-

request message from the target in the join procedure. This device consists of three parts:

- Signal-Capturing Part: it captures the signal from the target (1) and then transmits the packets to the Processing-Packet Part (2).
- Processing-Packet Part: it analyzes, decrypts, extracts the necessary information from the captured packets. If the packets come from the target, the Processing-Packet Part will trigger the operation of the Jammer Part (3) to interfere with the communication of that target, and then transfers the packets to the replay device (4) through LTE, Wi-Fi... on another frequency or even uses a wired network to avoid noise.
- Jammer Part: it is installed near the target. It emits the interference frequency to interrupt the transmission process to the Network Server of the target according to the control of the Processing-Packet Part.

Replay stage: After the jamming process and extracting the necessary information, a replay attack will be performed with a replay device. This device consists of two parts:

- Signal-Receiving Part: It receives information from the Processing-Packet Part, then it proceeds to create a fake packet containing the information in the join-request message of the target device. This part will send that fake packet to Forged Part (5), similar to the target end device.
- Forged Part: it sends a fake join-request message (6) to the Network Server to gain access to the network. The fake message will be accepted as a valid request, and the attacker's device can participate in the LoRaWAN network and take the next destructive steps.

VII. Comparison of Existing Solutions

In this section, we will compare the existing solutions for the attacks as mentioned above. The results of this comparison are presented in Tab. II.

VIII. Conclusion

In this study, we discussed and compared two common forms of attacks during the join procedure: Jamming Attacks and Replay Attacks. Additionally, we compared existing strategies for mitigating these types of threats.

Furthermore, we provided an assumption on the combination of these two types of attacks.

We conducted our investigation using LoRaWAN version 1.0.3. While doing this investigation, a new LoRaWAN version (version 1.0.4) with some improvements was released. As a result, some of the properties described in the paper may not apply to version 1.0.4. We are currently researching version 1.0.4 and will include any changes in our upcoming study.

Acknowledgment

The publication has been prepared with the support of the grant from the President of the Russian Federation for state support of leading scientific schools of the Russian Federation according to the research project SS-2604.2020.9.

References

[1] N. E. Oweis, C. Aracenay, W. George, M. Oweis, H. Soori, and V. Snasel, "Internet of things: Overview, sources, applications and challenges," in *Proceedings of the Second International Afro-European Conference for Industrial Advancement AECIA 2015*. Springer International Publishing, 2016, pp. 57–67.

[2] R. Kirichek, A. Vladyko, M. Zakharov, and A. Koucheryavy, "Model networks for internet of things and SDN," in *2016 18th International Conference on Advanced Communication Technology (ICACT)*. IEEE, Jan. 2016.

[3] P. Datta and B. Sharma, "A survey on IoT architectures, protocols, security and smart city based applications," in *2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. IEEE, Jul. 2017.

[4] A. Koucheryavy, A. Vladyko, and R. Kirichek, "State of the art and research challenges for public flying ubiquitous sensor networks," in *Lecture Notes in Computer Science*. Springer International Publishing, 2015, pp. 299–308.

[5] M. Bor, J. E. Vidler, and U. Roedig, "Lora for the internet of things," 2016.

[6] J. de Carvalho Silva, J. J. Rodrigues, A. M. Alberti, P. Solic, and A. L. Aquino, "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.

[7] A. Yastrebova, R. Kirichek, Y. Koucheryavy, A. Borodin, and A. Koucheryavy, "Future networks 2030: Architecture & requirements," in *2018 10th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*. IEEE, Nov. 2018. [Online]. Available: https://doi.org/10.1109/icumt.2018.8631208

[8] M. Al-gaashani, M. S. A. Muthanna, K. Abdukodir, A. Muthanna, and R. Kirichek, "Intelligent system architecture for smart city and its applications based edge computing," in *2020 12th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*. IEEE, Oct. 2020.

[9] R. Kirichek and V. Kulik, "Long-range data transmission on flying ubiquitous sensor networks (FUSN) by using LPWAN protocols," in *Communications in Computer and Information Science*. Springer International Publishing, 2016, pp. 442–453.

[10] L. Alliance, "White paper: A technical overview of lora and lorawan," *The LoRa Alliance: San Ramon, CA, USA*, pp. 7–11, 2015.

[11] M. Eldefrawy, I. Butun, N. Pereira, and M. Gidlund, "Formal security analysis of LoRaWAN," *Computer Networks*, vol. 148, pp. 328–339, Jan. 2019.

[12] I. Butun, N. Pereira, and M. Gidlund, "Security risk analysis of LoRaWAN and future directions," *Future Internet*, vol. 11, no. 1, p. 3, Dec. 2018.

[13] I. Butun, N. Pereira, and [et.], "Analysis of LoRaWAN v1.1 security," in *Proceedings of the 4th ACM MobiHoc Workshop on Experiences with the Design and Implementation of Smart Objects - SMARTOBJECTS '18*. ACM Press, 2018.

[14] E. van Es, H. Vranken, and A. Hommersom, "Denial-of-service attacks on LoRaWAN," in *Proceedings of the 13th International Conference on Availability, Reliability and Security*. ACM, Aug. 2018.

[15] M. Vanhoef and F. Piessens, "Advanced wi-fi attacks using commodity hardware," in *Proceedings of the 30th Annual Computer Security Applications Conference*. ACM, Dec. 2014.

[16] S. Zulian, "Security threat analysis and countermeasures for lorawan join procedure," 2016.

[17] E. Aras, N. Small, G. S. Ramachandran, S. Delbruel, W. Joosen, and D. Hughes, "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, Nov. 2017.

[18] S. Na, D. Hwang, W. Shin, and K.-H. Kim, "Scenario and countermeasure for replay attack using join request messages in LoRaWAN," in *2017 International Conference on Information Networking (ICOIN)*. IEEE, 2017.

TABLE II: A comparison between the jamming attacks and the replay attacks

Solution	Description	Attack type	Advantages	Disadvantages
Creating dense LoRa networks with overlapping coverage regions [17]	Deploying LoRaWAN end devices within the range of multiple gateways makes the jamming hard to be performed because the jammers are challenging to prevent the end devices to connect to a gateway. The jamming will become more complicated when using this technique because the attacker must identify all the end devices' gateways.	Jamming attacks, combined attacks	No need to modify the application level	High cost, inefficient for jamming attacks that have an influence range wider than the coverage.
Optimizing the channel hopping usage [17]	LoRa devices hop between multiple channels when sending messages to reduce collisions. The more channels used, the more complex the jammer has to be, as it needs to listen on all of those channels	Jamming attacks, combined attacks	The built-in mechanism in LoRa specification, easy to configure	High cost, high energy consumption
Using higher spreading factor (SF) in Chirp Spread Spectrum (CSS) techniques [13]	The higher SFs require higher dB differentials between the jammer and target message.	Jamming attacks, combined attacks	Increasing jamming process time and limiting multiple transmissions of the jammer.	High cost, high energy consumption
Analyzing the transmission rate [17]	Performing traffic analysis and profiling (at the gateway or server level). If the sending rate of the end device is aware, the Network Server can identify abnormal traffic patterns	Jamming attacks, replay attacks, combined attacks	Only need to apply at the Network Server and it can react accordingly to the unplanned changes	Difficult to determine traffic patterns and it is inappropriate when the deviation in the transmission time of the packets is small.
Encrypting the packet using the XOR algorithm [18]	Using a token generated from the previous session keys to XOR with a join message to create a new encrypted packet. Only the end device and Network Server know this token and the attacker could hardly get the token to decode.	Replay attacks, combined attacks	Simple, high-security efficiency	Need more time to encode and decode the data at the end device and Network Server