Final Report of the Guide Study (CS6534)

Student Name: HUANG Kunlun

Student Number: 57878689

Topic of Proposal: Physical layer key generation for wireless sensor networks

Supervisor: Prof. XU Weitao

November, 2023

TABLE OF CONTENTS

TABLE OF CONTENTS I

CHAPTER 1 INTRODUCTION 1

1.1 Background.......... 1

1.2 Objectives 3

CHAPTER 2 RELATED WORK.......... 4

2.1 Theoretical Foundations 4

2.1.1 Physical Layer Key Generation 4

2.1.2 LoRa Physical Layer Protocol.......... 5

2.2 Existing Solutions 6

2.2.1 CFR-Based Physical Layer Key Generation.......... 6

2.2.2 RSSI-Based Physical Layer Key Generation.......... 6

2.2.3 Chaos-based Physical Layer Key Generation 7

2.2.4 Deep-Learning-Based Physical Layer Key Generation 7

2.2.5 Quantum-Based Physical Layer Key Distribution.......... 7

2.3 Advantages and Disadvantages.......... 8

2.4 Relationship with Proposed Solution 9

CHAPTER 3 PROJECT OUTLINE AND METHODOLOGY 11

3.1 Project Overview 11

3.2 Methodology 11

3.2.1 LoRa Physical Layer Emulation and Analyst 11

3.2.2 Arduino Shield featuring LoRa technology.......... 11

3.2.3 Arduino's UNO Programmable Board and IDE 12

3.2.4 Deep-in-Building and Outdoor Test 13

3.2.5 Data Extraction and Analyst.......... 14

3.2.6 Physical Key Generation 16

3.2.7 Attempt on Decentralization of Key Distribution.......... 16

CHAPTER 4 SYSTEM MODELING AND STRUCTURE.......... 17

4.1 Communication Architecture 17

4.2 LoRa End Nodes Modeling and Algorithm.......... 18

4.3 Physical Layer Key Generation Modeling and Algorithm……………………….21

REFERENCES…………………………………………………………………25

CHAPTER 1 INTRODUCTION

1.1 Background

Network security is a recent scorching topic and affects everyone's vital interests. Mature PKI-based password management systems such as SSL and TLS are regarded as an excellent way to solve the problem of encrypted communication in computer networks. However, in the field of the Internet of Things, due to issues such as node computing performance and the difficulties in key management in large amount of nodes, PKI systems are not suitable in it. According to Gartner[1], there will be over 43 billion IoT devices connected to the internet by 2023, and the IoT market size will reach US$1.4 trillion. The rapid development of the IoT is mainly due to the following trends. Also, the MIIT suggests that the IoT is a future development trend that will have a profound impact on all industries[2]. The security of the communication channel in the Internet of Things is of utmost importance, given the critical role of IoT. Ensuring the security of IoT communication channels will be a highly significant and crucial subject. To address this problem, and based on previous research, some physical layer key generation methods have been proposed. Physical layer key generation is a technique for generating cryptographic keys from the physical characteristics of a wireless communication channel. This can be done by exploiting the unique features of the channel, such as its fading characteristics, noise properties, and path loss. Physical layer key generation is particularly attractive for wireless sensor networks (WSNs) because it does not require any prior shared information between the nodes, and it can be implemented with relatively low computational overhead. Physical layer key generation has recently been a research hotspot in academia and industry[3]. However, existing research often focuses only on traditional wireless communication technologies such as Wi-Fi, ZigBee, and 5G Radio Access Network. However, in the area of Low-Power Wide-Area Networks (LPWAN)[4], the long communication distance, low power consumption, and low transmission rate bring new research challenges to physical layer key generation. LoRa, a widely used LPWAN communication technology, is a physical layer modulation method defined by Semtech Corporation and based on the Chirp spread spectrum technology, which achieves a reception sensitivity of -148 dBm, a small data rate (0.3-50kbps) in exchange for a high communication range (3km in urban areas and 15km in suburban areas) and low power consumption (battery-powered

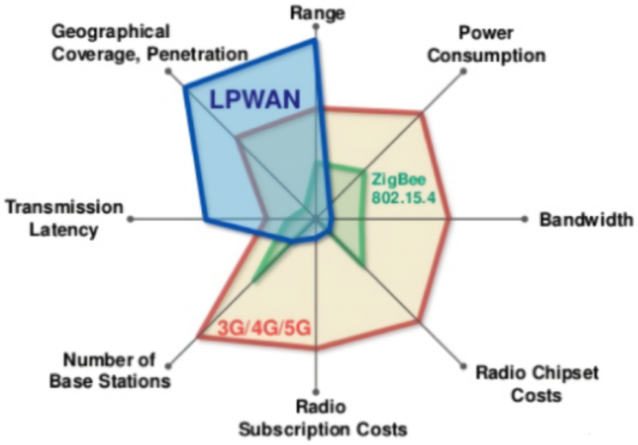


Figure 1.1 LPWAN vs. Other wireless communication technologies[4]

operation for up to 10 years under certain conditions)[5]. It provides a reliable connectivity solution for low-power IoT devices, especially for communication or data interaction with outdoor wireless sensor networks. Despite many advantages, the biggest problem is that physical layer encryption for LoRa is not well standardized. Because it is a type of wireless communication tech that has to broadcast its signal, it makes the communication information easy to capture and monitor. Thus, physical layer encryption for LoRa communication is a significant topic. In short, physical layer key generation for LoRa is still under development, but it can potentially revolutionize the security of LoRa-based networks. It can be used to secure a wide range of applications, including smart cities, industrial automation, and the Internet of Things.

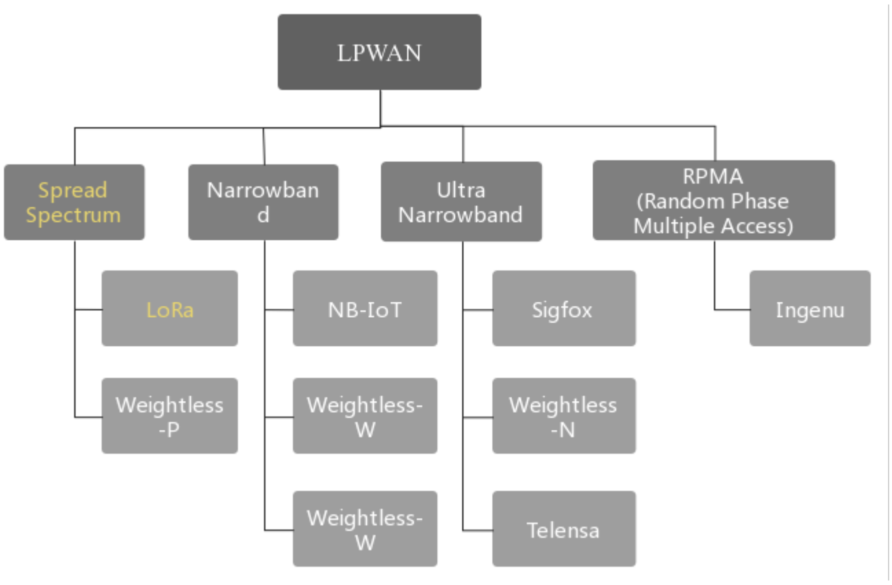


Figure 1.2 LPWAN Classification

1.2 Objectives

The specific research objectives are:

1. Understand the characteristics and limitations of LoRa communication technology, including its communication range, power consumption, data rate, and noise immunity.

2. Select and configure suitable LoRa modules and development boards to realize the LoRa communication protocol stack and basic data transmission functions.

3. Implement a physical layer encryption algorithm for LoRa communication to protect the privacy and confidentiality of data. The algorithm should suit LoRa's low data rate and long communication distance characteristics and have strong anti-interference and anti-jamming capabilities.

4. Test and evaluate the performance of the encryption algorithm via multiple scenario experiments and simulations. Compare the performance with existing key generation algorithms for LoRa physical layer and analyze its pros and cons.

CHAPTER 2 RELATED WORK

According to the previous research, some methods of key generation based on the physical layer of the LoRa communication protocol were proposed. However, Physical layer key generation for LoRa is still under development, but it has the potential to revolutionize the security of LoRa-based networks. It can be used to secure a wide range of applications, including smart cities, industrial automation, and the Internet of Things. Physical layer key generation for LoRa is a promising technique for establishing secure communication links in LoRa-based networks. Physical layer key generation exploits the randomness and reciprocity of the LoRa channel to extract a secret key shared by the communicating devices. This key can then be used to encrypt and decrypt their communication, ensuring confidentiality and integrity.

2.1 Theoretical Foundations

2.1.1 Physical Layer Key Generation

Physical layer key generation is the way to generate a shared secret key between wireless devices by exploiting the reciprocity of the random fading channel, in which wireless devices measure highly correlated wireless channel characteristics (e.g., channel impulse responses or received signal strengths) and use them as shared random sources to generate a shared key. In theory, in a rich multipath scattering environment, a passive attacker who is more than a half-wavelength away from the legitimate users will obtain uncorrelated channel measurements and thus cannot infer much information about the generated key. The physical layer key generation mechanisms do not require expensive computation and have the potential to achieve information-theoretic security in the sense that the secrecy of the generated key is not dependent on the hardness of a computational problem but relies on the physical laws of the wireless fading channels.[3] Generally, physical layer key generation applies the following five steps to generate a key: channel probing, randomness extraction, quantization, information reconciliation, and privacy amplification, illustrated in Fig.2-1.

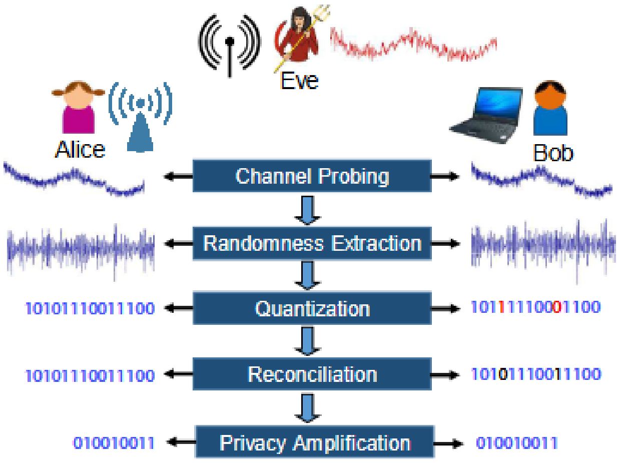


Figure 2.1 Secret key generation model[3]

2.1.2 LoRa Physical Layer Protocol

LoRa adopts the Chirp Spread Spectrum (CSS) modulation mechanism, which provides anti-interference and long-range communication capability. As shown in Figure 2-2(a) and (b), a base symbol in LoRa physical layer protocol (PHY) is a chirp with frequency linearly increasing with time. The start frequency (fstart) of a symbol represents the encoded information. A LoRa symbol has two segments with a sharp frequency drop, as shown in Figure 2-2(c). And the RSSI (Received Signal Strength Indicator) is a relative measurement that helps us determine if the received signal is strong enough to get a good wireless connection from the transmitter. Since LoR supports bi-directional communication, The RSSI is an important measurement for both gateways and end devices. And RSSI is measured in dBm and its value is a negative form. The closer the RSSI value is

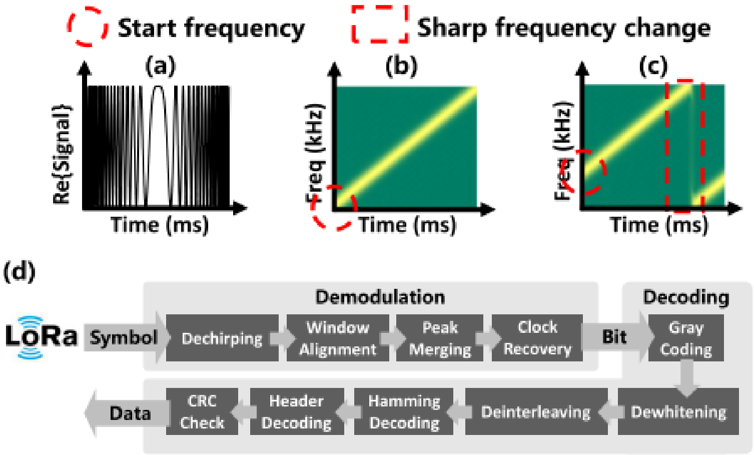
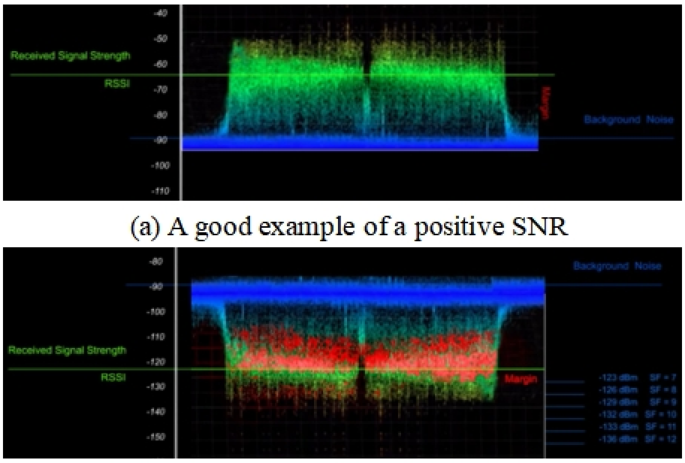


Figure 2.2 (a) Real part of a base up-chirp symbol. (b) Base up-chirp symbol. (c) Shifted symbol. (d) Complete procedure of LoRa PHY[6]

to zero, the received signal is stronger, the good examples for positive and negative have shown in Figure 2-3 (a) and (b)[7]. And the following factors mainly influence the RSSI: Path loss, Antenna gain, Cable/connector loss.



(a) A good example of a positive SNR

(b) A good example of a negative SNR

Figure 2.3 example of a positive/negative SNR

2.2 Existing Solutions

2.2.1 CFR-Based Physical Layer Key Generation

One common approach to Physical layer key generation for LoRa is to use the fine-grained channel frequency response (CFR). Y. Peng et al. proposed a secret key generation method from CFR for OFDM TDD systems[8]. Moreover, H. Luo et al. proposed a Channel Frequency Response-Based Secret Key Generation Scheme in In-Band Full-Duplex MIMO-OFDM Systems[9]. The CFR is also a unique character- istic of the LoRa channel that depends on the distance, orientation, and other environmental factors. By estimating the CFR at both ends of the communication link, the devices can generate a secret key that is shared only between them.

2.2.2 RSSI-Based Physical Layer Key Generation

Another approach to physical layer key generation for LoRa is to use the received signal strength indicator (RSSI). The RSSI is a measure of the strength of the received signal. S. Jana et al. evaluated the effectiveness of secret key extraction for private communication between two wireless devices from the received signal strength variations on the wireless channel between the two devices[10]. After that, W. Xu et al. employed a number of signal processing techniques to improve key generation rate significantly and proposed a novel compressive sensing-based reconciliation framework to reduce the mismatch rate[11]. The RSSI is also affected by the distance and other environmental factors. By exchanging RSSI measurements, the devices can generate a secret key that is shared only between them.

2.2.3 Chaos-based Physical Layer Key Generation

One another emerging technique for physical layer key generation in wireless networks is the use of chaos theory. Chaos theory, which is a branch of mathematics, can be applied to generate randomness in wireless communication links. The chaotic behavior of the wireless channel can be exploited to generate a random bit sequence that is used as a secret key shared only between the communicating devices. Yahya M. Al-Moliki et al. proposed a chaotic key creation approach, that is introduced by including the position-sensitive and real-valued channel state information of the VLC channel[12]. The chaotic nature of the wireless channel ensures that the generated key is random and difficult to predict, making it difficult for adversaries to eavesdrop on the communication link.

2.2.4 Deep-Learning-Based Physical Layer Key Generation

A related approach to physical layer key generation for wireless network is the use of machine learning algorithms. Deep learning or machine learning algorithms can be used to analyze the wireless channel characteristics and identify patterns that can be used to generate a secret key. For example, by using supervised learning algorithms, the devices can analyze the wireless channel behavior and identify features that are unique to each communication link. These features can then be used to generate a unique secret key that is shared only between the devices. X. Zhang et al. implemented a Deep-Learning-Based Physical-Layer Secret Key Generation for FDD Systems[13].

2.2.5 Quantum-Based Physical Layer Key Distribution

Using quantum mechanics principles for physical layer key distribution is also a promising approach for wireless networks. Quantum mechanics principles can be applied to generate truly random keys using the quantum states of particles. The devices can use quantum states of particles to generate random bits that are used as secret keys for encrypting and decrypting their communication. Using quantum keys ensures that the key is truly random and cannot be predicted or intercepted by an adversary. However, several technical limitations, such as channel losses and transmission distance, should be overcome before it comes to practical application.

2.3 Advantages and Disadvantages

Firstly, chaos theory for physical layer key generation in wireless networks has several strengths and weaknesses. On the one hand, the chaotic behavior of the wireless channel ensures that the generated key is random and difficult to predict, making it difficult for adversaries to eavesdrop on the communication link. This approach provides high security and randomness, which is essential for secure communication. However, on the other hand, using chaos theory for physical layer key generation may be challenging in practical implementation. The chaotic behavior of the wireless channel may be difficult to characterize and control in practical environments, making it difficult to ensure consistent and reliable key generation. Also, chaos theory may be more complex mathematical than other key generation techniques, making it challenging to implement and scale in resource-constrained environments. Secondly, using quantum-based principles for physical layer key distribution in wireless networks has several strengths and weaknesses. On the one hand, quantum keys ensure that the key is truly random and cannot be predicted or intercepted by an adversary, providing high security and randomness. This approach is auspicious for ensuring secure communication in wireless networks. However, on the other hand, the quantum-based is typically designed for optical fiber or free-space optical communication, not for RF wireless communication like LoRa. The use of quantum mechanics principles may be challenging in practical implementation. Generating quantum keys requires using specialized quantum devices and technologies that may not be widely available or practical for resource-constrained environments. Additionally, quantum communication is weaker to practical imperfections than classical communication, making it difficult to ensure consistent and reliable key distribution under practical conditions. Thirdly, the Deep-learning-based method offers promising security and key establishment advantages without pre-shared secrets. However, it also faces challeng- es related to data quality, model vulnerabilities, and resource constraints. It is not a standard or suitable choice for LoRa technology due to resource constraints, data availability, real-time requirements, complexity, and the presence of established security measures within the LoR protocol. Implementing Deep-Learning-Based LoRa networks would require overcoming these challenges and aligning with the specific requirements and limitations of LoRa technology. Such three methods fundamentally differ in operating principles, resource requirements, environmental sensitivity, standardization, and security considerations. Typically, the hardware performance of LoRa nodes, including the CPU and memory specifications limitation, is designed for Low-Power and Battery-Operated, which means they are not providing such calculation performance. Thus, no matter in theory or industrial area, most research on the Physical-Layer Secret Key Generation for LoRa are based on their wireless channel characteristics as very fast and efficient to implement in LoRa Node.

2.4 Relationship with Proposed Solution

In the context of the guide study, the primary focus revolves around the inherent characteristics of the Internet of Things. Given the inherent constraints imposed by the limited computational capacity of LoRa nodes and a spectrum of attributes, including communication speed and error rates, I have elected to adopt a widely acknowledged approach within the industry. This approach, rooted in the Received Signal Strength Indicator principles, serves as the foundation for the generation of physical-layer cryptographic keys. Moreover, my methodology draws inspiration from established practices documented in pertinent literature. To enhance the quality of the raw signal data, I intend to implement a series of preprocessing techniques, encompassing bit quantization and signal demodulation. The present study and research direction exhibit significant potential, particularly within the realm of the Internet of Things and the nuanced field of physical-layer key generation facilitated by LoRa technology. Preprocessing Methodologies: In the context of managing RSSI data, the employment of advanced preprocessing techniques assumes paramount importance. Strategies such as bit quantization and signal demodulation should be explored in depth, and their application should be calibrated to optimize precision and accuracy. An inclusive and thorough approach to validation should be adopted, spanning a spectrum of operational contexts that encapsulate the diversity of real-world IoT deployment scenarios. Rigorous validation across an array of settings, encompassing both indoor and outdoor environments, as well as scenarios with varying degrees of obstruction, will ensure the adaptability and efficacy of the proposed method. Furthermore, I have meticulously devised a comprehensive validation strategy tailored to the real-world deployment scenarios characteristic of LoRa technology. The testing plan includes but is not limited to the potential critical use case for LoRa technology, which is in the field of environmental monitoring. By doing so, I endeavor to substantiate the practical applicability of the theoretical framework I have laid out. Also, the methodical planning and meticulous experimental design are the cornerstones of empirical validation. The experimental framework should be meticulously devised to yield a substantial corpus of empirical evidence, reflecting the manifold operational contexts under scrutiny.

CHAPTER 3 PROJECT OUTLINE AND METHODOLOGY

3.1 Project Overview

In this project, I am supposed to mainly implement each LoRa node hardware with a programmable Arduino shield and a Dragino's LoRa expansion board for Arduino and then program the communication protocol with the Arduino IDE software with the LoRa KIT. Including the following: The implementation of LoRa nodes' communication, such as signal sending and receiving, coding and decoding, The extraction and analysis of the physical layer information between the communication process, The implementation of the LoRa key generation and the test on it. For the current interim progress, I have already implemented the following:

1.LoRa Physical Protocol emulation,

2.The design of each programmable Arduino shield for LoRa communication,

3.The logical implement of the programmable shields for LoRa,

4.Deep-in-Building Test Data Collection.

3.2 Methodology

3.2.1 LoRa Physical Layer Emulation and Analyst

Before the design and development in real vibe, the understanding and simulation of the LoRa physical layer could be significant since it has its characteristic, which is different from the standard communication protocol. The LoRaPHY KIT is for LoRa Physical Protocol emulation[6]. Through this KIT, the payload of LoRa can be simulated, especially, and the encryption information can be analyzed and visualized before and after.

3.2.2 Arduino Shield featuring LoRa technology

Typically, LoRa uses SX127x chipset, as shown as Fig 3-2, by Semtech, and an Arduino Shield features LoRa technology based on an Open source library. This shield allows users to send data and reach extremely long ranges at low data rates. It provides ultra-long-range spread spectrum communication and high interference immunity while minimizing current consumption. It is on the Semtech SX1276/SX1278 chip, and it targets

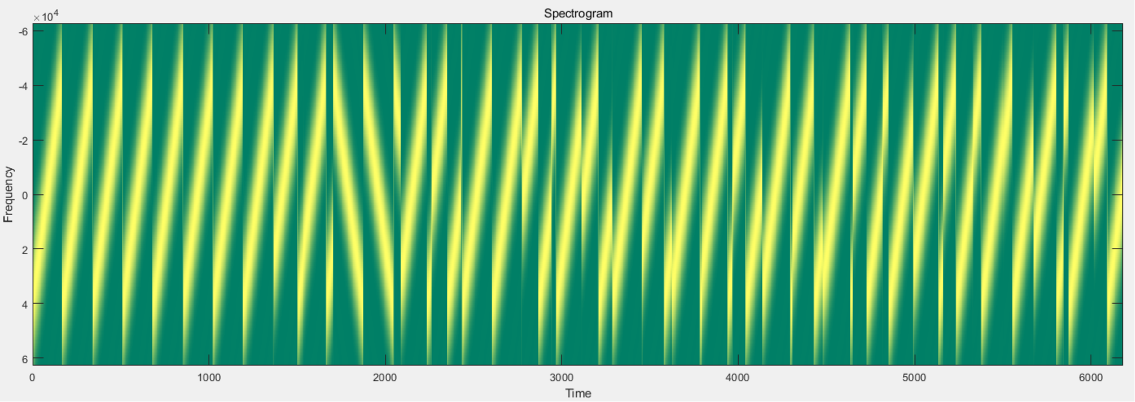


Figure 3.1 A emulated LoRa physical layer payload of '[78 22 43 44 12]'

professional wireless sensor network applications such as irrigation systems, smart metering, smart cities, and building automation. Moreover, using the SX127x LoRa technique, the shield can achieve a sensitivity of over -148dBm using a low-cost crystal and bill of materials. The high sensitivity combined with the integrated +20 dBm power amplifier yields an industry-leading link budget, making it optimal for any application requiring range or robustness. By using these components, the LoRa communication protocol can be well physically implemented and is a powerful and adaptable tool that can support the realization of the project.

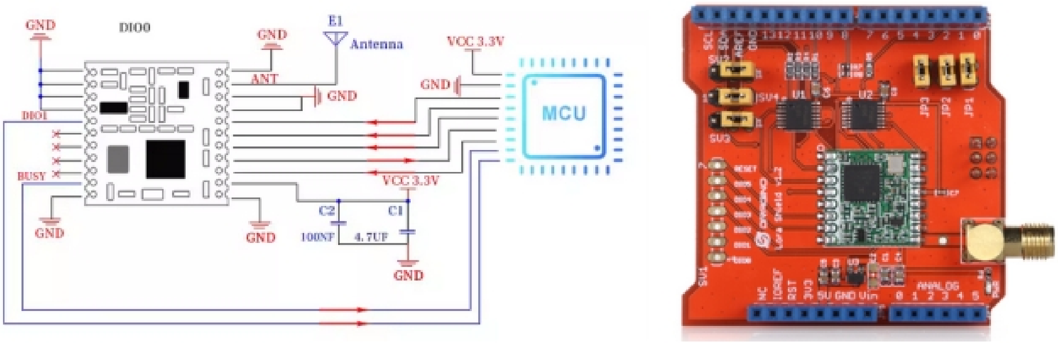


Figure 3.2 The SX127x schematic diagram and physical representation

3.2.3 Arduino's UNO Programmable Board and IDE

The Arduino Uno is a popular microcontroller board widely used in the maker and electronics communities for various projects and prototypes. Although the Arduino Uno itself does not possess native LoRa capabilities, it can be effectively employed as a control and interface unit to facilitate seamless integration with external LoRa modules or chipsets, shown as Fig 3-4. The integration process entails meticulous wiring connections between the Arduino Uno and the LoRa module, encompassing essential pins such as those for power (3.3V or 5V), ground (GND), and serial communication (TX and RX). Voltage levels are diligently considered to ensure compatibility between the Arduino Uno and the LoRa module. In addition to the essential connections, supplementary connections, such as those related to module reset or configuration, may be required depending on the module's specifications. To facilitate effective communication with the LoRa module, the Arduino IDE is utilized. Relevant libraries that optimize interactions with the LoRa module are installed, ensuring the Arduino Uno is aligned with the specific requirements of the chosen module. The subsequent phase entails the creation of Arduino code tailored to configure and orchestrate communication with the LoRa module. This code encompasses fundamental tasks such as module initialization, parameter configuration (e.g., frequency, spreading factor, bandwidth), and the transmission or reception of LoRa data packets. The intricacies of the code are contingent upon the precise LoRa module chosen and the intended application. Upon the completion of the code development, it is compiled and uploaded to the Arduino Uno via the ubiquitous USB interface, positioning the controller as a command and control center for the LoRa module. By using this set of components, it is possible to effectively implement the logic for sending and receiving data between various LoRa nodes, as well as extracting features such as RSSI from the communication channel. Additionally, it enables the implementation of logic for physical layer encryption.

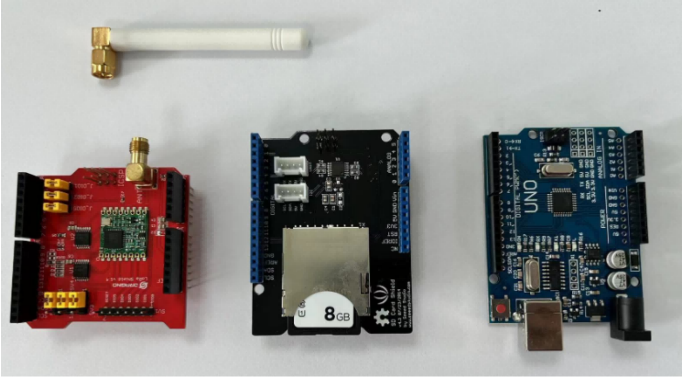


Figure 3.3 The SX127x and Arduino's UNO Programmable Board

3.2.4 Deep-in-Building and Outdoor Test

The comprehensive evaluation of LoRa communication systems necessitates the execution of rigorous deep-in-building and outdoor tests, each designed to address distinct aspects of performance and coverage within varying environmental contexts. The principal objective of the deep-in-building test for LoRa communication is to scrutinize the signal propagation characteristics within the intricate confines of interior structures, encompassing walls, floors, and architectural elements. The outdoor test regimen for LoRa communication entails a meticulous assessment of the system's performance attributes in open-air or outdoor settings, characterized by the absence of indoor obstructions. At the same time, this testing plan is combined with a potential critical use case for LoRa technology in the field of environmental monitoring[14]. Currently, most environmental monitoring data is transmitted using mobile networks, which can be costly and dependent on the infrastructure of telecommunication operators. In remote areas like the wilderness or deep in the mountains, there is often a lack of cellular signal coverage, even though environmental monitoring activities frequently need to be conducted in these areas. Therefore, this testing plan encompasses outdoor fixed monitoring stations, central monitoring facilities, and mobile monitoring stations for environmental monitoring. It is expected to effectively carry out a series of tests, including deep-in-building and outdoor tests.

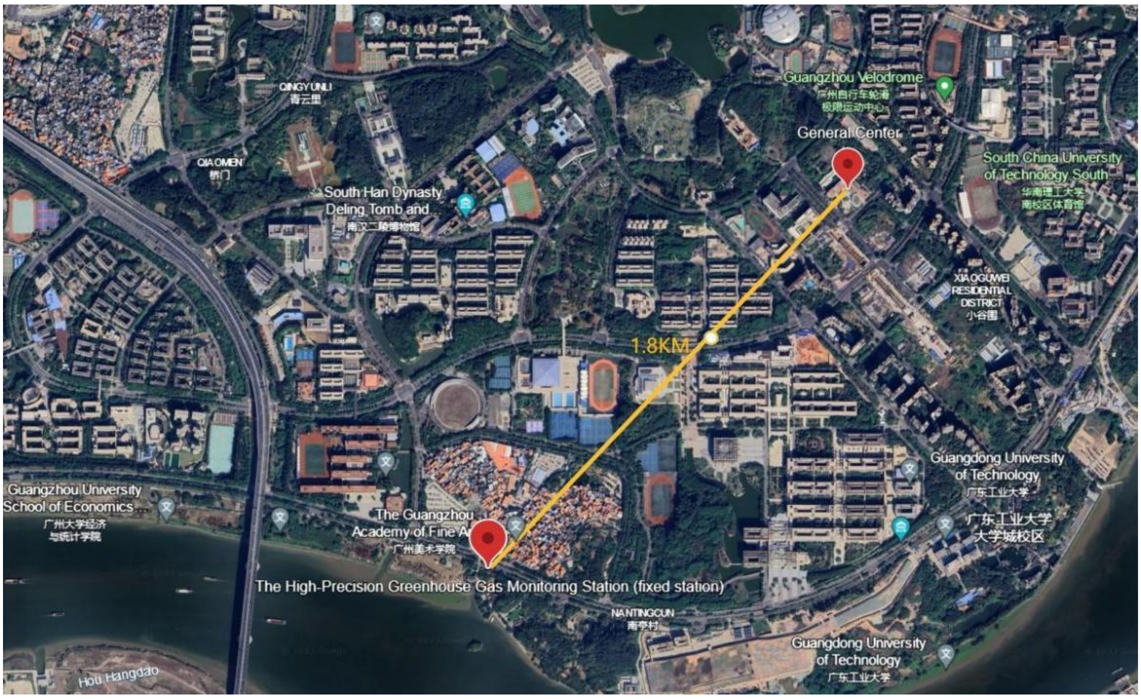


Figure 3.4 Outdoor test between an fixed High-Precision Greenhouse Gas Monitoring Station and the General Center (1.8km)

3.2.5 Data Extraction and Analyst

By testing in various scenarios, a series of test data can be obtained. The next step is to extract and preprocess the data. In particular, the analysis of feature data from the signals. Through the analysis of this data, we can understand the characteristics of LoRa

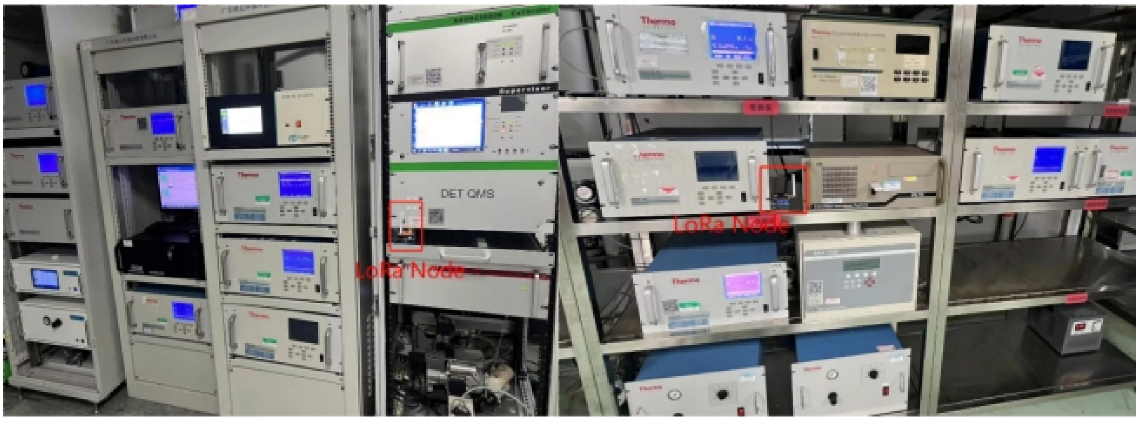


Figure 3.5 Deep-In-Building Test at General Center

signal features in different scenarios and under different electromagnetic background interferences. Also, the Savitzky-Golay filter will be applied to the raw signal datasets. The filter is a commonly used technique in signal processing for smoothing or filtering noisy data, especially in the field of spectroscopy and chromatography. The primary goal of the Savitzky-Golay filter[15] is to remove noise from a signal while preserving the important features, such as peaks and valleys. It works by fitting a polynomial to a small window of data points and then using this polynomial to estimate the value of the central data point. This process is repeated for all data points, effectively smoothing the signal. After

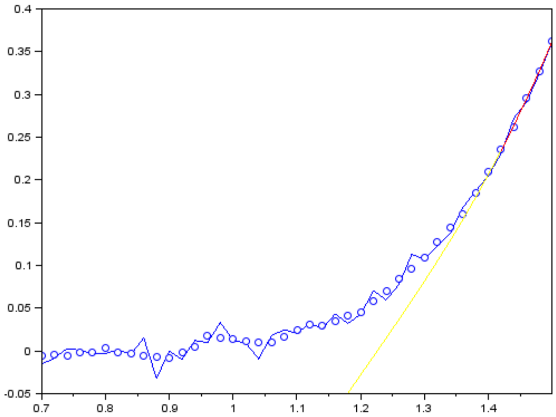


Figure 3.6 Savitzky-Golay filter

the data extraction and preprocessing, the analyst can begin to explore the characteristics of LoRa signal features in different scenarios and under different electromagnetic background interferences. This analysis will help to determine the optimal communication parameters and signal processing methods for specific applications. And Based on the results of these tests, the analyst can provide decision-makers with data-driven insights on how to optimize LoRa communication systems for different environmental contexts. This information is crucial for companies and organizations to make informed decisions about how to design, deploy, and maintain LoRa communication networks that are reliable and efficient for their specific applications.

3.2.6 Physical Key Generation

Physical layer key generation is a method for generating a shared secret key between wireless devices by exploiting the reciprocity of the random fading channel. And wireless devices measure highly correlated wireless channel characteristics and use them as shared random sources to generate a shared key. In this section, the primary focus is on the physical layer key generation based on relevant algorithms. This includes the processing of signal feature data using methods such as Bit-Quantization and reconciliation. Additionally, it involves comparing the key matching rates and key generation speeds for different parameters values. Some related research achievements in Lora physical layer encryption will be utilized in this chapter.

3.2.7 Attempt on Decentralization of Key Distribution

After testing the physical layer key generation algorithms, considering the communication characteristics of IoT and the limited memory and computational capabilities of LoRa nodes, an attempt will be made to propose a decentralized physical layer generation scheme based on node calculations.

CHAPTER 4 SYSTEM MODELING AND STRUCTURE

4.1 Communication Architecture

The architecture mainly describes Alice and Bob as two LoRa nodes that need to generate the same communication key. It starts with one of them initiating the key generation communication (a random LoRa message with a sequence number). When the other party (Bob) receives the request, it immediately replies with a message with a sequence number. The pair of two LoRa packets that in the communicate with each other are both marked with RSSI. These tags are mainly generated by the strength of the signal via the communication. Since wireless message transmission can be regarded as extremely fast propagation, which is close to the speed of light, the RSSI values that the two nodes can obtain in this communication process are close. Or, to be more specific, if LoRa messages are generated with the same pattern, the trend of RSSI values obtained at both ends over a period of time is close. In addition, since RSSI is mainly affected by signal strength, in the real world, it is generally due to changes in the surrounding environment, such as changes in the communication distance between the two nodes, the appearance of some obstacles between the nodes, the weather, or interference from other signals, etc. For the two nodes, the impact is the same, especially for the physical communication channel formed by the two nodes. Therefore, the trend of RSSI values obtained by the two nodes is a good physical layer feature factor. As shown in the figure 4-1, after n interactions, n pairs

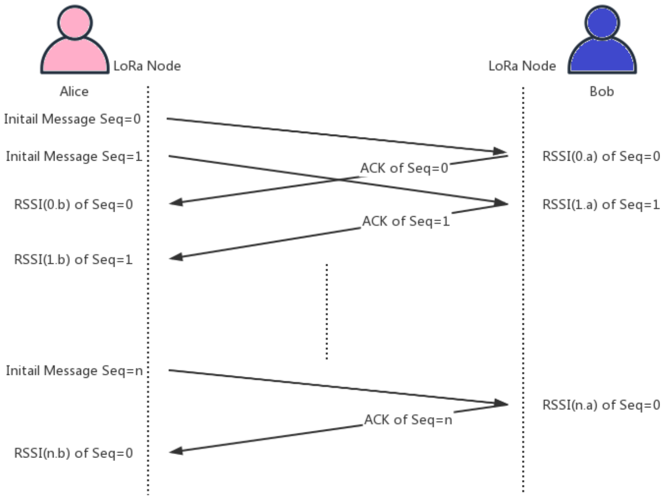


Figure 4.1 Communication Architecture

of RSSI values can be obtained. Then, through a certain algorithm, the n pairs of RSSI values can be modulated and used as factors to generate a key. In this way, the two nodes can obtain a common physical layer communication key. However, in actual use, due to interference from the actual environment, such as physical obstruction, channel blocking, and frequency band interference, the LoRa communication packets cannot always be received as expected. In particular, due to long-distance transmission, LoRa physical layer messages are not reliable transmissions, so packet loss needs to be considered. In addition, according to this communication model, the biggest challenge to generating a pair of RSSI is still Bob's reply message. This is because Bob's reply is based on the premise of receiving Alice's message, but Alice may not be able to receive Bob's reply.

4.2 LoRa End Nodes Modeling and Algorithm

As mentioned in the previous text, LoRa nodes are half-duplex devices. In order to implement the mentioned functional model, a specific algorithm is required to enable LoRa nodes to both listen to messages and transmit messages after receiving information. One way to implement this functional model is to use a polling approach. In this approach, LoRa nodes periodically poll other nodes to see if they have any information to send. If the polling detects that other nodes have information to send, the LoRa node will stop listening and transmit a message.Another way to implement this functional model is to use an event-driven approach. In this approach, LoRa nodes register an event handler that will be triggered when they receive information from other nodes. When the event handler is triggered, the LoRa node will stop listening and transmit a message. Specifically, in the polling approach, LoRa nodes can use the following steps to implement the aforementioned functional model:

- Set a polling interval, such as 100 milliseconds.
- Within the polling interval, LoRa nodes will send a query message to other nodes.

If LoRa nodes receive a response message from other nodes, the LoRa nodes will stop listening and transmit a message. In the event-driven approach, LoRa nodes can use the following steps to implement the aforementioned functional model:

- Register an event handler that will be triggered when they receive information from other nodes.
- In the event handler, LoRa nodes will stop listening and transmit a message.

Both of these algorithms have their own advantages and disadvantages. The advantage of the polling approach is that it is simple to implement, but it can reduce the LoRa node's receiving sensitivity. The advantage of the event-driven approach is that it can improve the LoRa node's receiving sensitivity, but it requires additional overhead for the event handler. The purpose of this algorithm is to establish a two-way communication channel over LoRa. The "SEQ" message is a special message that is used to initiate a communication session. The onReceive() function can be used to process any incoming messages, such as acknowledgment messages or data messages. The loop starts by checking the millis() function to see how long it has been since the last message was sent. If it has been longer than the interval period, then the loop sends a SYN message using the sendMessage() function. The loop then calls the onReceive() function to parse for any incoming packets. If a packet is received, the onReceive() function will be called with the parsed packet data. The sendMessage algorithm encapsulates a message for LoRa communication. It constructs a packet containing destination and sender addresses, a message ID, payload length, and the actual message. The LoRa packet is then sent, and the message ID is incremented for the next message. This function is designed for a larger program implementing LoRa communication. The onReceive algorithm for Alice processes incoming LoRa packets, extracting and verifying header information, checking the recip-

Algorithm 4.1 Algorithm for Alice

```
Initialize variables:
$lastSendTime \leftarrow 0$
$interval \leftarrow$ user-defined interval
function LOOP
    if millis() − lastSendTime > interval then
        $message \leftarrow$ "SEQ" $\leftarrow n$
        SENDMESSAGE($message$)
        print "Sending" + $message$
        $lastSendTime \leftarrow$ millis()
        $interval \leftarrow$ random(2000) + 1000                ▷ 2-3 seconds
    end if
    parse for a packet, and call ONRECEIVE(parsePacket())
end function
```

Algorithm 4.2 Algorithm for sendMessage

```
function SENDMESSAGE(outgoing)
    Initialize $message$
    Add $message \leftarrow$ Payload $\leftarrow outgoing$
    Send $message$
    $msgCount \leftarrow msgCount + 1$
end function
```

Algorithm 4.3 onReceive Algorithm for Alice

```
function onReceive(packetSize)
    if packetSize = 0 then
        return
    end if
                                        ▷ Read packet header bytes:
    Initialize $incoming$
    while LoRa.available() do
        $incoming \leftarrow incoming$ + (char)LoRa.read()
    end while
    if incomingLength $\neq$ incoming.length() then
        print "error: message length does not match length"
        return
    end if
    if $recipient \neq$ localAddress $\wedge$ $recipient \neq 0xFF$ then
        print "This message is not for me."
        return
    end if
                        ▷ Print message details for this device or broadcast:
    print $message$
end function
```

ient address, and providing detailed information about the received message, including sender and recipient addresses, message ID, length, RSSI values, and Snr. The algorithm for bob continuously parses for incoming LoRa packets and invokes the onReceive function, passing the result of as the packet size parameter. The onReceive algorithm for Bob processes incoming LoRa packets, extracting and verifying header information, checking the recipient address, providing detailed information about the received message, and then sending a response message containing the received signal strength indication (RSSI) using the sendMessage function.

Algorithm 4.4 Algorithm for Bob

```
Initialize variables
function loop
    parse for a packet, and call onReceive(parsePacket())
end function
```

Algorithm 4.5 onReceive Algorithm for Bob

```
function onReceive(packetSize)
    Algorithm 4.3 onReceive Algorithm for Alice
    $outgoing \leftarrow incoming.SEQ$
    print $sendMessage(outgoing)$
end function
```

4.3 Physical Layer Key Generation Modeling and Algorithm

After the LoRa end node setup, we can process the physical layer key generation by the following steps:

Algorithm 4.6 RSSI-Based Key Generation Algorithm

```
Require: $X_A$ and $X_B$ : the original sample RSSI list
  Initialize variables
  $X'_A$, $X'_B$ ← Savitzky–Golay filter($X_A$, $X_B$)
  repeat
      $K'_A$ ← Quantization($X'_A$)
      $K'_B$ ← Quantization($X'_B$)
  until $K''_A \leftarrow Reconciliation(K'_A) = K'_B$
  $K_{Share}$ ← Privacy_Amplification($K''_A$, $K'_B$)
  return $K_{Share}$                              ▷ Secret Key for Alice and Bob
```

Sampling: Alice and Bob exchange a number of probe and response packets to sample the RSSI channel by the mentioned in 4.2 LoRa End Nodes Modeling and Algorithm.

Signal processing: Alice and Bob apply outlier detection and Savitzky–Golay filter, which is mentioned in 3.2.5, to reduce the discrepancies caused by the environmental noise. They then use linear interpolation to construct missing data points. The Savitzky–Golay algorithm is a method for smoothing and differentiating data. Given a set of data points y_i at positions x_i for $i = 1, 2, \dots, n$, the smoothed data $\hat{y}_i$ is computed using the convolution operation:

$$\hat{y}_i = \sum_{j=-k}^{k} c_j \cdot y_{i+j}$$

where k is the half-width of the smoothing window and c_j are the Savitzky–Golay coefficients. The coefficients can be computed by solving the linear system of equations:

$$\begin{bmatrix} S_0 & S_1 & \dots & S_{2k} \\ S_1 & S_2 & \dots & S_{2k+1} \\ \vdots & \vdots & \ddots & \vdots \\ S_{2k} & S_{2k+1} & \dots & S_{4k} \end{bmatrix} \begin{bmatrix} c_{-k} \\ c_{-(k-1)} \\ \vdots \\ c_k \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

where $S_j = \sum_{i=1}^{n} x_i^j$. The solution to this system provides the coefficients c_j for the convolution operation.

Multilevel Quantization: Alice and Bob convert the RSSI values to bit strings by employing multilevel quantization technique.

Algorithm 4.7 Multilevel Quantization

```
function QUANTIZATION(sampleArray, numBitsPerSample, alpha)
    Variables Initailzation
    for $i \leftarrow 1$ to $M$ do
        levelBase[$i$] ← offset
        levelTop[$i$] ← levelBase[$i$] + stepSize
        offset ← offset + stepSize + gbandSize
    end for
    decimalValArray ← []
    validIndices ← []
    for $i \leftarrow 1$ to sampleArrayLength do
        for $j \leftarrow 1$ to $M$ do
            if sampleArray[$i$] = minVal then
                decimalValArray[$i$] ← 1                ▷ Decimal assignment starts from 0
                validIndices[$i$] ← $i$
                break
            else if sampleArray[$i$] = maxVal then
                decimalValArray[$i$] ← $M - 1$
                validIndices[$i$] ← $i$
                break
            else if levelBase[$j - 1$] ≤ sampleArray[$i$] ≤ levelTop[$j - 1$] then
                decimalValArray[$i$] ← $j - 1$
                validIndices[$i$] ← $i$
                break
            end if
        end for
    end for
    Variables Updating
    for $i \leftarrow 1$ to decimalValArrayLen do
        bitString.extend(format(decimalValArray[$i$], f′0{numBitsPerSample}$b'$))
    end for
    return bitString, validIndices
end function
```

The function uses M-ary quantization to represent the input samples with a specified number of bits per sample, considering guard bands to reduce quantization errors. The resulting bit string can be used for further processing or transmission.

Reconciliation: Alice and Bob use a CS-based reconciliation method to correct the bit mismatches between their bit strings. The Reconciliation algorithm is commonly used for solving sparse linear regression problems, where the $l1$ regularization encourages sparsity in the solution. The function iteratively refines the solution by updating the active set and adjusting the solution vector based on the direction and step size. The algorithm terminates when the convergence criterion is satisfied or after a specified number of iterations. The algorithm then computes the active set, which is the set of indices of the elements in the solution that have the largest residual correlations. The active set is computed by finding the elements in the solution that have a residual correlation that is within $1e-5$ of

Algorithm 4.8 Reconciliation Algorithm

```
function RECONCILIATION($A, y$)
    $Variables\ Initailzation$
    for iter_idx ← 1 to iter_times do
        c ← $A^T \cdot (y - A \cdot x)$
        lambda_max_idx ← argmax(|c|)
        lambda_max ← |c[lambda_max_idx]|
        act_set ← where(|c − lambda_max| < $1e - 5$)
        state ← zeros vector of size $m$
        state[act_set] ← 1
        $R$ ← $A$[:, act_set]$^T \cdot A$[:, act_set]
        $d$ ← pinv($R$) · sign($c$[act_set])
        gamma ← 1000
        for idx ← 1 to $m - 1$ do
            if state[idx] then
                my_id ← where(act_set = idx)
                tmp ← max(0, $-x$[idx]/$d$[my_id])
            else
                av ← $A$[:, idx]$^T$ · ($A$[:, act_set] · $d$)
                tmp1 ← max(0, (lambda_max − $c$[idx])/(1 − av))
                tmp2 ← max(0, (lambda_max + $c$[idx])/(1 + av))
                tmp ← min(tmp1, tmp2)
            end if
            if tmp > 0 then
                gamma ← min(tmp, gamma)
            end if
        end for
        $x$[act_set] ← $x$[act_set] + (gamma · $d$)[0]
        if $\|y - A \cdot x\|_2 < 1e - 6$ then
            break
        end if
    end for
    return $x$, iter_idx
end function
```

the maximum residual correlation. After Reconciliation, performs computing the bitwise exclusive-Or (XOR) of two binary arrays, bits_a (Bits of Alice after Quantization) and mismatch, and then casting the result to an integer array. This is a useful operation for error correction, as it can be used to recover the original bits from a corrupted set of bits, provided that the number of errors is less than half the number of bits.

bits_recover = np.logical_xor(bits_a, mismatch.reshape(len(mismatch))).astype(int)

Key Extraction: Alice and Bob extract a secure key from the reconciled bit strings using a shared secret key or a one-way hash function. Considering that the computing power of LoRa nodes is limited, especially in the case of IoT nodes without AES hardware acceleration, Chacha20(RFC 7539)[16], a symmetric encryption algorithm that is more suitable for embedded hardware, is chosen here. Chacha20 has a simpler process than the traditional AES algorithm and has achieved better performance such that in the absence of a dedicated accelerator[17-18].

REFERENCES

[1] Gartner Research. Cross-industry insight: Iot market opportunities and top spend use cases [EB/OL]. (2023-06-08)[2023-06-08]. https://www.gartner.com/en/documents/4432199.

[2] Ministry of Industry and Information Technology. Iot 13th five-year plan (2016-2020) years [S]. Beijing: The 13th five-year plan, 2016.

[3] Zeng K. Physical layer key generation in wireless networks: challenges and opportunities [J/OL]. IEEE Communications Magazine, 2015, 53(6): 33-39. DOI: 10.1109/MCOM.2015.7120014.

[4] Anciaux L. LPWAN - 5 letters in the heart of the iot revolution[N/OL]. IOT Factory, 2018-01-19 [2018-01-19]. https://iotfactory.eu/lpwan-5-letters-in-the-heart-of-the-iot-revolution/.

[5] LoRa Alliance. LoRaWAN® Regional Parameters RP002-1.0.2[S]. Fremont, CA: LoRa Alliance, 2022.

[6] Xu Z, Tong S, Xie P, et al. From demodulation to decoding: Toward complete lora phy understanding and implementation[J/OL]. ACM Trans. Sen. Netw., 2023, 18(4). https://doi.org/10.1145/3546869.

[7] The Things Netowrk. RSSI and SNR[EB/OL]. 2022. https://www.thethingsnetwork.org/docs/lorawan/rssi-and-snr/.

[8] Peng Y, Alexandropoulos G C, Wang P, et al. Poster: Secret key generation from cfr for ofdm tdd systems over fading channels[C/OL]//9th International Conference on Communications and Networking in China. 2014: 660-661. DOI: 10.1109/CHINACOM.2014.7054386.

[9] Luo H, Garg N, Ratnarajah T. A channel frequency response-based secret key generation scheme in in-band full-duplex mimo-ofdm systems[J/OL]. IEEE Journal on Selected Areas in Communications, 2023, 41(9): 2951-2965. DOI: 10.1109/JSAC.2023.3287610.

[10] Jana S, Premnath S N, Clark M, et al. On the effectiveness of secret key extraction from wireless signal strength in real environments[C/OL]//MobiCom ’09: Proceedings of the 15th Annual International Conference on Mobile Computing and Networking. New York, NY, USA: Association for Computing Machinery, 2009: 321–332. https://doi.org/10.1145/1614320.1614356.

[11] Xu W, Jha S, Hu W. Lora-key: Secure key generation system for lora-based network[J/OL]. IEEE Internet of Things Journal, 2019, 6(4): 6404-6416. DOI: 10.1109/JIOT.2018.2888553.

[12] Al-Moliki Y M, Alresheedi M T, Al-Harthi Y. Chaos-based physical-layer encryption for ofdm-based vlc schemes with robustness against known/chosen plaintext attacks[J/OL]. IET Optoelectronics, 2019, 13(3): 124-133. https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-opt.2018.5072. DOI: https://doi.org/10.1049/iet-opt.2018.5072.

[13] Zhang X, Li G, Zhang J, et al. Deep-learning-based physical-layer secret key generation for fdd systems[J/OL]. IEEE Internet of Things Journal, 2022, 9(8): 6081-6094. DOI: 10.1109/JIOT.2021.3109272.

[14] Botero-valencia J, Castano-Londono L, Marquez-Viloria D, et al. Data reduction in a low-cost environmental monitoring system based on lora for wsn[J/OL]. IEEE Internet of Things Journal, 2019, 6(2): 3024-3030. DOI: 10.1109/JIOT.2018.2878528.

[15] Savitzky A, Golay M J E. Smoothing and differentiation of data by simplified least squares procedures.[J/OL]. Analytical Chemistry, 1964, 36(8): 1627-1639. https://doi.org/10.1021/ac60214a047.

[16] Nir Y, Langley A. Request for comments: number 7539 ChaCha20 and Poly1305 for IETF Protocols[M/OL]. RFC Editor, 2015. https://www.rfc-editor.org/info/rfc7539. DOI: 10.17487/RFC7539.

[17] Velea R, Gurzău F, Mărgărit L, et al. Performance of parallel chacha20 stream cipher[C/OL]// 2016 IEEE 11th International Symposium on Applied Computational Intelligence and Informatics (SACI). 2016: 391-396. DOI: 10.1109/SACI.2016.7507408.

[18] De Santis F, Schauer A, Sigl G. Chacha20-poly1305 authenticated encryption for high-speed embedded iot applications[C/OL]//Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017. 2017: 692-697. DOI: 10.23919/DATE.2017.7927078.