Security in the Internet of Things – a LoRa study

Récoules Frédéric

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE

INSTITUT **N**ATIONAL DES **S**CIENCES **A**PPLIQUEES

TOULOUSE

Electrical & Computer Engineering Department

GRADUATION PROJECT

Security in the Internet of Things

LAAS CNRS –

7 avenue du Colonel Roche

31077 Toulouse Cedex 4

Recoules Frédéric

5I – IL

June 2016

ABSTRACT

The Internet of Things is a dream which is becoming true. The technology is now sufficiently advanced to make the first application a reality.

There can be no doubt that this future endless pool of data floating on the air will draw attention of the so-called hackers. It is very alarming to realize we could give them the key to act in the physical world.

There is too much at stake to just stand by. The scientific community has to work with the industry to make sure the security will not be left in favor of profit.

In this report, I present the work of my final project of the engineer school. It was a small contribution to the project to make the IoT secure.

As an attacker, I tried to misuse the LoRa network. In order to do so, I use the new possibilities offered by a new kind of reconfigurable hardware called USRP. I fear, this kind of cheap equipment could make the attacks easier.

But, as I am not a malicious person, my ultimate goal was, when I found a vulnerability, to proposed solutions to fix it.

ACKNOWNLEDGMENTS

I would like to take this opportunity to thank all the people who have contributed in some way to this report, particularly Ms. DRAGOMIRESCU and Mr. ALATA who offered the internship to me. I thank them for supervising my work, for their attention and for the time they awarded for me.

I am most especially thanking Mr. MABILLE for the quality of its work and the clarity of its presentations.

I also thank all the members of the MINC and TSF teams who were always ready to help me and support me.

Then, I would like to thank all the members of the LAAS CNRS who, directly or indirectly, have lent their helping hand in this venture.

Finally, I would like to thank Ms. HUGUET who helped me and who gave me all her attention.

TABLE OF CONTENTS

Abstract iii

Acknownledgments iv

Table of contents v

Glossary vi

Chapter 1. Introduction 1

Chapter 2. Contex and objective 3

Chapter 3. The LoRa physical layer 5

3.1. The data set 5

3.2. Prototyping the software demodulator 7

Chapter 4. The LoRaWAN protocol 9

4.1. Theoretical analysis 9

4.2. Validation in controlled environment 10

4.2.1. Setting up of a private network 10

4.2.2. Realization of attacks 12

Chapter 5. Conclusions and perspectives 15

Appendix A : LAAS CNRS 17

Appendix B : LoRa modulation diagram 18

Appendix C : LoRa modulation documentation 19

Bibliography 23

List of Illustrations 25

GLOSSARY

FFT: the Fast Fourier Transform is an algorithm to quickly compute the Fourier Transform of a signal. It has a time complexity of $o(nlog(n))$ instead of $o(n^2)$.

LoRa: it is the abbreviation of long range.

M2M: it is the abbreviation to machine to machine. It is used to name a process that does not need the man supervision.

MINC: one of the team of the LAAS which is working on the wireless communication development.

RF: it is the abbreviation of radio frequency domain.

SDR: it is the abbreviation for software defined radio. A specialized kind of application to do RF processing.

TSF: one of the team of the LAAS which is working on the fault-tolerance and the security in the IT system.

USRP: the Universal Software Radio Peripheral is a hardware device which is used as a radio front end for a host computer. It is completely reconfigurable and it could be used for many means.

WAN: it is the abbreviation of wide arena network. It is a kind of network using kilometric scale wireless communication technology.

CHAPTER 1. INTRODUCTION

A man as normal as possible is always seeking ways of making its life easier.

Internet of Things could be the end state of an age-old dream of mankind. As the result of progress in a lot of domains, electronic, radiofrequency, algorithm and big data, the IoT have begun to come true.

It is still in its infancy, but experts are confident and they expect a wide deployment in the coming years. It seems that its future is bright as they provide for no fewer than 50 billion objects by 2020.

We are in 2030, the present has exceeded all expectations. All my gear is connected. They send a lot of information in a cloud which now rules every aspect of my life. It will be seven years I have not turned a doorknob as the doors automatically detect my authorization and open. But, this morning, the garage door slammed shut on the front of my car for no apparent reason. It was a malicious person who hacked my home. It was a real shock for me as it compromised the work I have done for over fourteen years.

This scenario takes a pessimistic point of view, I agree, but it is not improbable. Recent debates over the Sigfox network suggest that security is not really the essential priority of object suppliers. However, they show it still is a major point of concern for customers and I will assure you, we are not the firsts trying to address the problem.

I am sure you have understood, we must take care of security aspect long before the wide deployment take place and so it was too late.

So, it is surprising or even troubling, to see there are not yet so many works related to that. And that is why my tutors opened a topic to study this new kind of network.

The security in the IT is conversely a well-known topic but, because of the nature of the objects, we must not reuse its finding as it is and we need a whole new approach to study them.

Especially because there is not case of related attacks, we have limited experience in how hackers should proceed. Moreover, the emergence of a new kind of cheap reconfigurable RF hardware could be in their favor.

In this report, I will present the result of my work in the teams of LAAS CNRS.

Firstly, I will talk about the context and the objectives which were set for me.

Then, I will present a use case of what may be done with an USRP in the hand of a curious person or of someone with malicious intents.

Afterward, I will present the approach I have used to investigate the LoRa networks. I studied the state-of-the-art and I compared it with my own analysis. Then I needed to formulate and validate my hypothesis by implementing the attacks I found in a real environment.

Finally, I will conclude by talking about my future work to complete my internship and the future prospects in the world of research.

If you are interested by some technical details, I turn your attention to the appendix selection.

CHAPTER 2. CONTEX AND OBJECTIVE

It is a turning point for the Internet of Things. The idea is not new but it is just starting to be possible. The LAAS is very involved in the research of this domain. An example of its work is the smart home project driven by the team ADREAM. It is not the only team working on it. The concern about the IoT is now the security.

However it involves several domains and so, no team is more qualified than others to work on it. My internship is a collaborative effort between the two teams MINC and TSF. Following the success of the previous experience with Mr. MABILLE, a INSA intern too, my tutors decided to carry on in this way.

It is a very big topic. To make the internship possible, we had to reduce its scope.

So we decided to focus to the Wide Area Network of sensor because they are a big part of the IoT and so far the most representative.

More specifically, we focused on the LoRa technology for several reasons.

Firstly, between Sigfox and LoRa, the two leading technologies, the first one was already being investigated and this is evident with the recent debate.

Then, as practical, there is the fact that the team already has a working set of LoRa chips as Mr. MABILLE worked on it and his work was a great starting point for mine. On the Figure 1 and 2, you can see the custom nodes he made with a Semtech SX1272 chip and a microcontroller LPC1768.

Finally, there are more and more people who plan to use it for their future applications. Mention can be made, for example, of the two operators Bouygues Telecom and Orange who recently announced the deployment of a LoRa based network to cover France. We also need to talk about The Thing Network, a crow founded citywide network in Amsterdam, which now trying to spread to the other cities.

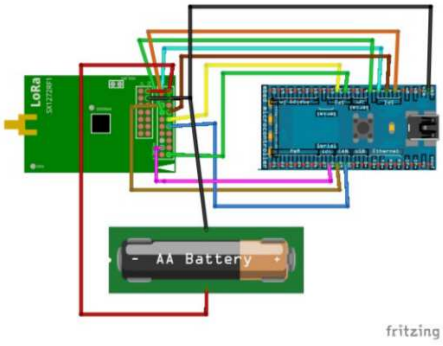


Figure 1. Circuit diagram of a LoRa node

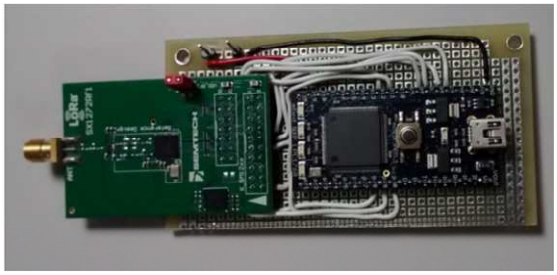


Figure 2. Picture of a LoRa node

We were very curious of knowing the specific features of the LoRa technology. The modulation technique was still ununderstood and for my first task, I was asked to demystify it.

They gave me an USRP as you can see it in the Figure 3. In the box, it contains a complete RF chain front-end and an FPGA, allowing us to use in many RF applications. My task aimed both to enrich our knowledge by understanding the modulation and to proof that the USRPs are effective to be used as potential attacker equipment.

Figure 3. Picture of the USRP NI 2900

My second task was to make a study on the mac protocol used by the LoRa network.

As I was not, from my formation, an expert in security, I had to learn by myself a lot of theories. Making the state-of-the-art helped me too to learn the basics of network security.

Also, I needed to understand how the protocol is working in order to find some weak points which could be used by an attacker.

The last task of my internship that is remaining is to test the hypothesis on an operational network to validate or invalidate them. The LAAS does not have yet a network to play with it. With the help of my new knowledge and my work, I needed to set up our own network in order to try to break it.

CHAPTER 3. THE LORA PHYSICAL LAYER

Due to the nature of radio communication, wireless cannot really get along with security. As it is subject to eavesdrop and jam, the first task allotted to me was to study the physical layer of LoRa.

However, the technology LoRa use a special kind of modulation called chirp spread spectrum, which can appear a little bit obscure for the non-specialists.

This modulation presents some interesting characteristics which make it more resilient to traditional wireless perturbations. I was surprised and challenged by its capability to well resist to a hand-made white noise jammer.

Moreover, there is no so far another way to receive or emit LoRa frames than the use of a chip of manufacturers.

It could be seen as the so called "security through obscurity" because it results that an attacker will face difficulties, for a variety of technical reasons, to eavesdrop communications.

I think it was not what it was intended.

On one hand, I hope we are all aware the attackers do not worry about this kind of details as they demonstrate their resourcefulness for reverse-engineering operations.

On the other hand, the overall process of modulation is explained in published patent and so, everyone may try to understand it. In fact, there is already a project which tries to demodulate LoRa frame with a cheap Software Defined Radio but it lacks of results.

At this point, I tried to understand the modulation process.

3.1. The data set

In order to validate a future algorithm, I needed a set of well-known records. It should be as much as possible exhaustive, but as it cannot cover all the combinations of parameters, I focus on a subset of the most used ones.

To discriminate the impact of the payloads in the resulted modulated signal, I recorded for each set of parameters a payload of zeros as reference and all the payloads obtained by a single bit flip from this reference.

Before starting the analysis, I made sure it is a repeatable process and so sending multiple times the same payloads with the same parameters will always produce the same signal.

I used a Universal Software Radio Peripheral NI 2900 driven by the open source software GNURadio.

To make the recording operation easier and less time consuming I developed a custom dedicated module to GNURadio to allow record only the interesting part of the signal, the frames, and discard the samples between them.

The sampling method

Thanks to the work of my predecessor, I already knew what a LoRa signal is like in the frequency domain. With the basics of signal processing theory, I know sampling is no trivial matter.

Due to the specific nature of chirp, there were some questions about the role of the carrier frequency. I was not sure that the fact to remove it will not alter the signal and so if I could only work with the base band signal. These doubts were supported by the fact my first results widely differed from what it was expected. As the result, I have taken some time to study the problem from a theoretical point of view and to convince me that the records are correct.

With the benefit of hindsight and the help of mathematical theory, it appears that these strange results were due to a misinterpretation of negative frequencies in the computation of the FFT.

3.2. Prototyping the software demodulator

It was a major investment of my internship. In fact, it was close to the reverse-engineering of a LoRa chip as if I was an attacker.

The knowledge of the patent made it more accessible, but it should be noted that all the stages of the process are not necessary implemented as they was described.

I did not even know the output of the intermediary stages. Each step was developed with the more flexible code to easily tweak the way it works. I could not know with certainty, before ending the overall process with success, what were the intermediary solutions. It required a high level of rigor to not be lost in all the opened options.

Bit pattern

Interpreting a flow of bits as close as the hardware work gives a lot of freedom. By the way, all the numerical values could be interpreted with different conventions.

I learn to my cost that the simple fact of reversing the bit order could lead to the solution or to a seeming like solution…

I developed some script to help me to reduce the set of the option by checking some constraint which cannot be violated.

As the result of rigor, intuition and, of course, some lucks, I came with a set of scripts able to detect, synchronize, demodulate and decode the content of a recorded frame.

I used GNU Octave as the top level language to write the script. It is an interpreted language which shares the same syntax as MatLab with the benefit to be free.

It is specialized in the computation of mathematics. It has a high support for the complex number and Fast Fourier Transform. It saves us to keep track the type and the precision of numeral values.

It does not need any compilation and so, save a lot of time while manually tweaking some stages.

It meets exactly the needs for a fast prototyping which serve only as a proof that the demodulation can be surely computed in software.

If you are curious, you will find in the annex B an overview of the LoRa modulation process.

CHAPTER 4. THE LORAWAN PROTOCOL

4.1. Theoretical analysis

The LoRaWAN protocol is promoted to be secure. As far as I could judge, it has been designed to take care about a lot of threats that the researchers have pointed out through the survey of literature.

But as it is flexible enough to operate over a wide range of hardware equipment, it gives a lot of freedom on the implementation while guaranteeing the interoperability of the devices.

It makes it even harder to examine the protocol from a security point of view. In addition to understanding the expected functionality of the system and how a malicious actor could misuse it, we must consider that the regular users are likely to harm themselves.

For example, a device owner can decide to use its own encryption algorithm to encode its data. As a side effect, it may therefore decide to send plain text data to save its valuable energy. This kind of deviant behavior is the best way to shoot itself in the foot for anyone who are not aware of security or law.

From the state of the art, I made a list of potential vulnerabilities we could find in a wireless sensor network. The analysis of the LoRaWAN specification and the discussion with my tutors allowed me to append some new kind of potential vulnerabilities to the list.

These vulnerabilities must be described by a security analysis to estimate the credibility, the scope and the damage it can cause. This allows us to sort them by the priority to validate them.

However, at this point, this is highly speculative.

4.2. Validation in controlled environment

The next step of my work was to verify, for each potential vulnerability we found, if they are or not true.

This is an essential stage, in order to test our hypotheses and it is a requirement for the work is being valued.

4.2.1. Setting up of a private network

Obviously, we cannot use a public network to make our experimentation.

It will not allow us to finely measure the impact of what we tried to do and I am not even talking about the legal qualification. But, in fact, the chief barrier is there is not yet a public LoRaWAN network which covers the LAAS.

I was given the task to set up a private network with the available equipment in order to practice the tests.

Upgrade of the nodes

The chips were flashed to send any payload by the host computer request. To be compliant with the LoRaWAN protocol, I needed to change the firmware.

Hopefully, there are a lot of available libraries for building a LoRaWAN node.

I firstly tried to adapt the code of LoRaMAC developed by IBM. Its advantage was it made an abstraction of the hardware and it is easily transferable to any device. Unfortunately, it is based on real time event driven engine and an intrinsic problem of the microcontroller make the use of timer unstable.

I left it because I had not enough time to try to debug the problem.

Then I used the LoRaWAN-lib. It was developed to work on a set of well-known chip and so, it requested a little more adaptation to work on our nodes but afterward, it has shown a great stability.

The nodes are now ready to send data to an application through a LoRaWAN network.

Configure the back-end

Unlike the node libraries, there is not a lot of available source code of the server. I use the sources of The Things Network but they are in constant evolution. This is showing the early stage of the technology.

By default, it was configured to connect to the community network, so, it had to be adjusted to meet the needs of a private network.

Build of the gateway

The protocol is only working as a node to gateway communication. It is organized in a star-of-stars topology. In our case, we need at least one gateway. Due to hardware capability, a node cannot be used as a gateway compliant with LoRaWAN. We need a specialized hardware called "concentrator" which is able to simultaneously listen on different RF channels.

We ordered one, but we knew there was no chance to receive it before the end of the internship.

Then we decided to try to emulate the hardware with an SDR. It would be the first "nearly" fully functional LoRa software support.

Thanks to my previous work, we knew it was theoretically possible. But we should take into account that many constraints have been added.

To be easily connected with the USRP, there could not have been a better choice than to use GNURadio. So, I must rewrite the algorithm in c++ and design the application to use the GNU framework.

The major difference is we wanted to move on from an offline record processing to a real time continuous data flow processing. So a sample must be treated once and only once as there is no more history. All the algorithms must keep track their state and became state machines.

Then, to emulate a concentrator, we need to be able to listen on several narrow band channels. With only one antenna, it means to increase the bandwidth until you cover all your channels. But it has the drawback, by increasing the sample rate, to significantly increase the required computational performance. It required a lot of optimization to make it practicable.

The laptop where I worked, can hardly handle a bandwidth of 8 MHz, but this is what it is needed to properly cover the legacy channels. To reduce the CPU load, I added a stage of focus. When a frame is detected, it moves the RF center frequency in the center of the channels

and greatly reduces the sample rate to 500 kHz for example. It gains enough in performance to run on my laptop.

The last step was to make it compatible with the back-end architecture of the network by making it able to handle the UDP packet with the expected format.

We can see in the Figure 4 the flow diagram composed of the block I made.

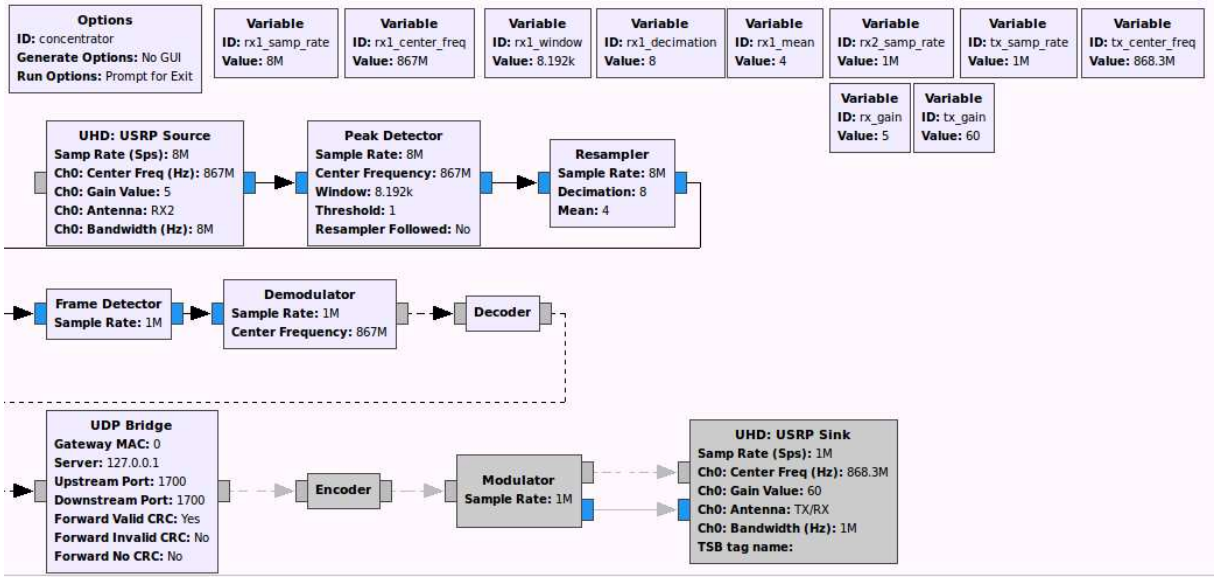


Figure 4. Complete flow diagram of the SDR concentrator

The result is still perfectible but it works as expected. It is able to correctly handle bidirectional communication between the end-node and the user application.

4.2.2. Realization of attacks

It is undoubtedly the most interesting part of the internship as it is the conclusion of the research I made so far.

Unfortunately, I would not be able to talk a lot about it as I did not spend yet enough time on it. It is because it required the private network to be set up and I have just finished it.

However, I took advantage of my knowledge in the demodulation process and in GNURadio to develop a new kind of radar jammer.

It has some attractive characteristics as it automatically scan the channels and activate when it detect a frame and if it was installed on the path between the transceiver and the receiver, it does not need to emit a signal stronger than the one it received.

Desynchronization

The preamble is the most important part of a LoRa signal. The receiver is very flexible and it is able to balance offset in frequency or timing with the transceiver.

By playing in canon a look like LoRa signal during the synchronization, it adds a non-existing bias that the receiver will erroneously try to cancel.

It has the advantage to be nearly invisible from an external point of view because it never floods the RF channel.

In general, there is no way to counter jamming except to find and deactivate his source. We must at least be detect we are victim of it. In this case, the only way I found is the gateway must keep track the number of reception with which it lost the synchronization. If this counter exceeds a threshold, the gateway must alert the network.

I am quite sure a lot of back-end implementation does not care about the number of failed receptions whereas this information is stored in the gateway firmware.

CHAPTER 5. CONCLUSIONS AND PERSPECTIVES

My work was mixed between reflection process and technical implementations. It is still some work to do and I am eager to put all the time I have left to finish my work. For now, my main realization is I made, as a proof of concept, a software LoRa concentrator which could be used both as part of the network gateway and as a hacker tool.

I am proud to have participated, even if it was little, to a greater goal.

I really appreciated to come out of my chosen field and to learn a little bit more about other domains. I made tremendous progress in the understanding of the signal processing and security and it was exactly what I expected when I was looking for an internship.

However, my work is not the main point of my internship. I would like to work for the research and I would apply to become a doctoral student. And there is a chance for me to remain in the LAAS community for the year to come.

Even if this is not the case, I have benefited from the presence of the doctoral students. More than never, I followed them in their daily life. I was at the scientific meeting. I could not have got a better idea of what looks like a thesis. I now have a better understanding of what are the challenges of the research.

That is how I really perceive this internship.

APPENDIX A : LAAS CNRS

The Laboratory of Analysis and Architecture of Systems is a CNRS research unit. It is one of the 34 Carnot institutes.

Located at Toulouse, it is associated with the five founding members of the University of Toulouse including the Institut National des Sciences Appliquées and the University Paul Sabatier.

It has approximately seven hundreds of employees for twenty research groups.

Its research activities are mainly focus on the Information Sciences and Technologies but they are not limited to this.

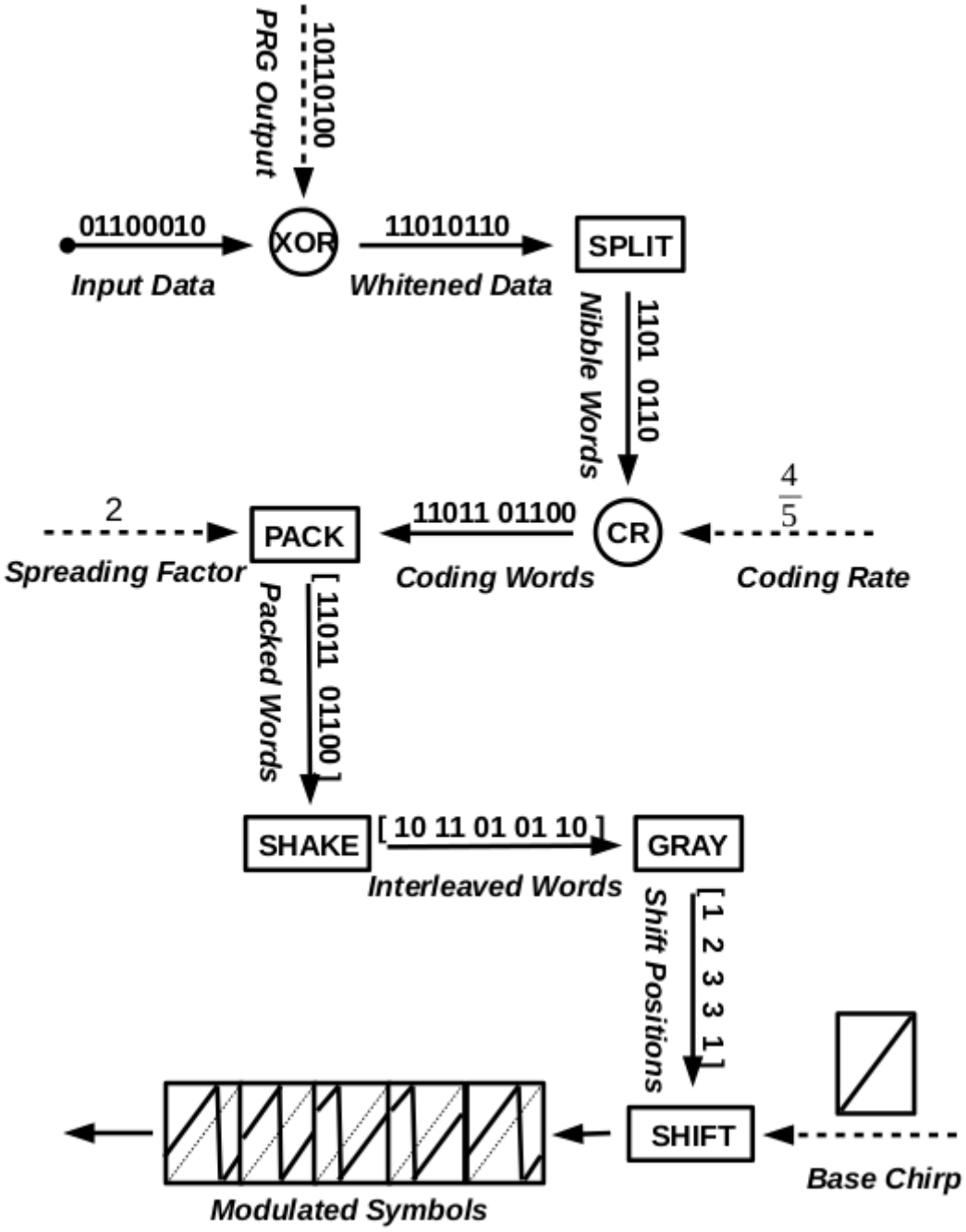
Scientific research is distributed into height main departments:

- Critical Infomation Processing
- Networks and Communications
- Robotics
- Decision and Optimization
- MicroNanosystems RF and Optics
- Nano-Engineering and Integration
- Energy Management
- MicroNanoBio Technologies

In addition, there is also three transversal axes defined across the domains:

- Adream: Architecture for Dynamic Resilient Embedded Autonomous Mobile Systems
- Alive: Interaction with the living
- Synergy: System for smart energy management

APPENDIX B : LORA MODULATION DIAGRAM



APPENDIX C : LORA MODULATION DOCUMENTATION

On the transmitter side, the physical modulation of LoRa can be decomposed in six steps.

Whitening

In the first place, the module multiplies the input data with a sequence of pseudo-random bits. This method is inherited of standard modulations like Frequency-Shift Keying to avoid a long string of homogeneous bits.

There is a Linear Feedback Shift Register (LFSR) which generate the pseudo-random string. On a SX1272, it is based around the 8-bit polynomial $x^8+x^4+x^3+x^2+1$. For each byte of data, it applies the XOR operator between the bits of data and those of the LFSR flip-flop. The clock of the LFSR ticks only one time between the bytes. The seed of the polynomial is 0xFF, so, at the initial stage, each flip-flop is set to '1'.

The first outputs of the LFSR arranged into bytes are: 0xFFFEFCF8F0E1C2...

Note: There is a need to be careful with the order of the bits. The module uses a little endian like convention. From this point, the input binary data need to be interpreted from left to right by the polynomial $1+x+x^2+x^3+x^4+x^5+x^6+x^7$, the perfect reverse of the common order.

Splitting

The module splits the data in nibble words. As the common is to send bytes overs the network, there are two nibbles by byte of data sent. Notice that the first nibble will contains the Least Significant Bits (LSB).

Error-correcting code

The module transforms each nibble into a code word. The number of bits of the resulted code word depends of the selected Coding Rate (CR). The module offers four CR. There are the parity bit $4/5$, the Hamming code $4/6$ and $4/7$ and the extended Hamming code $4/8$.

Packing

Depending of the Spreading Factor (SF), the module packs the nibbles into groups. The size of a group is equal to SF. If there are not enough code words to fill a group, the module adds padding words.

Interleaving

The module transposes a group of SF code words of length $4/CR$ into $4/CR$ symbol words of length SF.

The interleave is diagonal such as the bits of a code word are not at the same position in the symbol words.

Let define $C_{i,j}$ the bit j of the code word i and $S_{j,i}$ the bit i of the symbol word j. The relation is:

$$S_{j,\,\mathrm{mod}(i+j,SF)} = C_{i,j} \quad \text{for} \quad 0 \leqslant i < SF \quad \text{and} \quad 0 \leqslant j < 4/CR$$

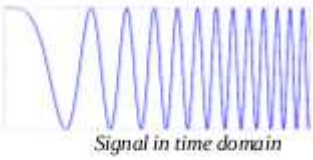
An example for SF equal to 7 and CR = 4/5 is the matrix:

	S_0	S_1	S_2	S_3	S_4
6	$C_{6,0}$	$C_{5,1}$	$C_{4,2}$	$C_{3,3}$	$C_{2,4}$
5	$C_{5,0}$	$C_{4,1}$	$C_{3,2}$	$C_{2,3}$	$C_{1,4}$
4	$C_{4,0}$	$C_{3,1}$	$C_{2,2}$	$C_{1,3}$	$C_{0,4}$
3	$C_{3,0}$	$C_{2,1}$	$C_{1,2}$	$C_{0,3}$	$C_{6,4}$
2	$C_{2,0}$	$C_{1,1}$	$C_{0,2}$	$C_{6,3}$	$C_{5,4}$
1	$C_{1,0}$	$C_{0,1}$	$C_{6,2}$	$C_{5,3}$	$C_{4,4}$
0	$C_{0,0}$	$C_{6,1}$	$C_{5,2}$	$C_{4,3}$	$C_{3,4}$

Matrix of interleaving

Shifting

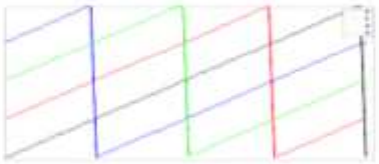
The module interprets the symbol words as a Gray coded numbers. They will determine the circular time-shift amounts of signal periods. The base symbol, as unmodulated or zero valued symbol, is the up-chirp. It is a signal in which the frequency increases linearly with time.



Signal in time domain

Instantaneous frequency in time domain

The symbol words have a length of SF bits. There is a total of 2^{SF} shift positions.



Possible shift positions of chirp for SF equal to 2

Demodulation

On the receiver side, the demodulation is the reverse sequence of the modulation process. The first step is the only one that should be considered.

Unshifting

Receiving the signal composed of multiple modulated chirps, the receiver have to determine the time-shift amount of each symbol. This supposes a fine symbol time synchronization. Assuming the synchronization is done, the receiver multiplies the signal with an own generated down-chirp signal.

The expression of a linear chirp signal is given by the formula $A\,e^{i2\pi(f_0+\frac{\Delta f}{2T}t-\frac{\Delta f}{2})t}$ where A is the amplitude, f_0 the central frequency, Δf the bandwidth and T the period.

Multiplying a τ delayed up-chirp $A_u e^{i2\pi(f_u+\frac{\Delta f}{2T}(t-\tau)-\frac{\Delta f}{2})(t-\tau)}$ with a down-chirp $A_d e^{i2\pi(f_d+\frac{\Delta f}{2T}t-\frac{\Delta f}{2})t}$ give the following signal $A_u A_d e^{i2\pi((f_u+f_d-\frac{\Delta f}{T}\tau)t+(\frac{\Delta f}{2T}(\tau+1)-f_u)\tau)}$. This signal has only one harmonic $f_u+f_d-\frac{\Delta f}{T}\tau$.

As the k^{th} time-shift, the delay τ has the value $-k\frac{T}{2^{SF}}$. The measured frequency of the resulted signal is $f_u+f_d+\frac{\Delta f}{2^{SF}}k$.

The receiver is able to determine the value of the position k because all the other elements are well known and constant.

Note: It is preferable to use an Fast Fourier Transform such as $\frac{\Delta f}{2^{SF}}$ is a multiple of the scale in frequency.

Physical frame

A LoRa frame is composed of three parts which are the preamble, the header and the payload. These three parts are not modulated from the same way.

Preamble

A frame begins with a sequence of an implicit number of unmodulated symbols to allow the receiver to lock on the the frequency and the timing of the transmitter. The length of this sequence is chosen long to lower the duty cycle of the receiver.

After them, it will be followed by two modulated symbols which have the value of the synchronization word.

The preamble is finished by two down-chirp unmodulated symbols.

Header

The header is composed of twenty bits of non-whitened data. It is composed with the length of the payload into two bytes, one bit for the presence of a payload CRC, the Coding Rate into three bits and a height bits header CRC.

0-7	8	9-11	12-20
P. Length	P. CRC	CR	H. CRC

The preamble is always followed by height symbols. They are the first group of data, encoded with a reduced set of positions (SF - 2) and with a coding rate of 4/8 .

This group can contain 4 * (SF – 2) bits. If the Spreading Factor is higher than seven, the module start storing nibbles of data into this first group.

Payload

The payload is composed of modulated symbols with the full set of position. If the payload contain a CRC, it is appended after the data and is not whitened.

BIBLIOGRAPHY

Jean-Marie Dilhac 2009, "*Une introduction aux télécommunications*".

Christophe Escriba, Jean-Yves Fourniols 2010, "*Systèmes électroniques analogiques : Amplification, filtrage et optronique*".

Stefan Schmidt, Holger Krahn, Stefan Fischer and Dietmar Wätjen, "*A Security Architecture for Mobile Wireless Sensor Networks*".

Chris Karlof, Naveen Sastry, David Wagner, "*TinySec: A Link Layer Security Architecture for Wireless Sensor Networks*"

Devesh Jinwala, Dhiren Patel and Kankar Dasgupta 2009, "*FlexiSec: A Configurable Link Layer Security Architecture for Wireless Sensor Networks*".

Bijoy Kumar Mandal, "*A Security Architecture for Wireless Sensor Networks Environmental*".

Robert Miller, "*LoRa Security, building a secure LoRa Solution*"

LIST OF ILLUSTRATIONS

Figure 1. Circuit diagram of a LoRa node 3
Figure 2. Picture of a LoRa node 4
Figure 3. Picture of the USRP NI 2900 4
Figure 4. Complete flow diagram of the SDR concentrator 12