LoRa Water Meter Security Analysis

UnicornTeam, 360 Security Technology

Who we are

The 360 Security Research Innovation Alliance consists of three teams. Each team has raised many brilliant researchers in the corresponding key areas.

UnicornTeam focuses on wireless security, and they assess the security of anything using radio technologies, from small items such as RFID/NFC, wireless sensors, drones, to large systems such as GPS, cellular networks, satellite communication systems.

RocTeam focuses on hardware development for defense and attack. They have built many hardware security products.

PegasusTeam focuses on WiFi security area. They designed ‘SkyScan’ , a commercial wireless intrusion prevention system.



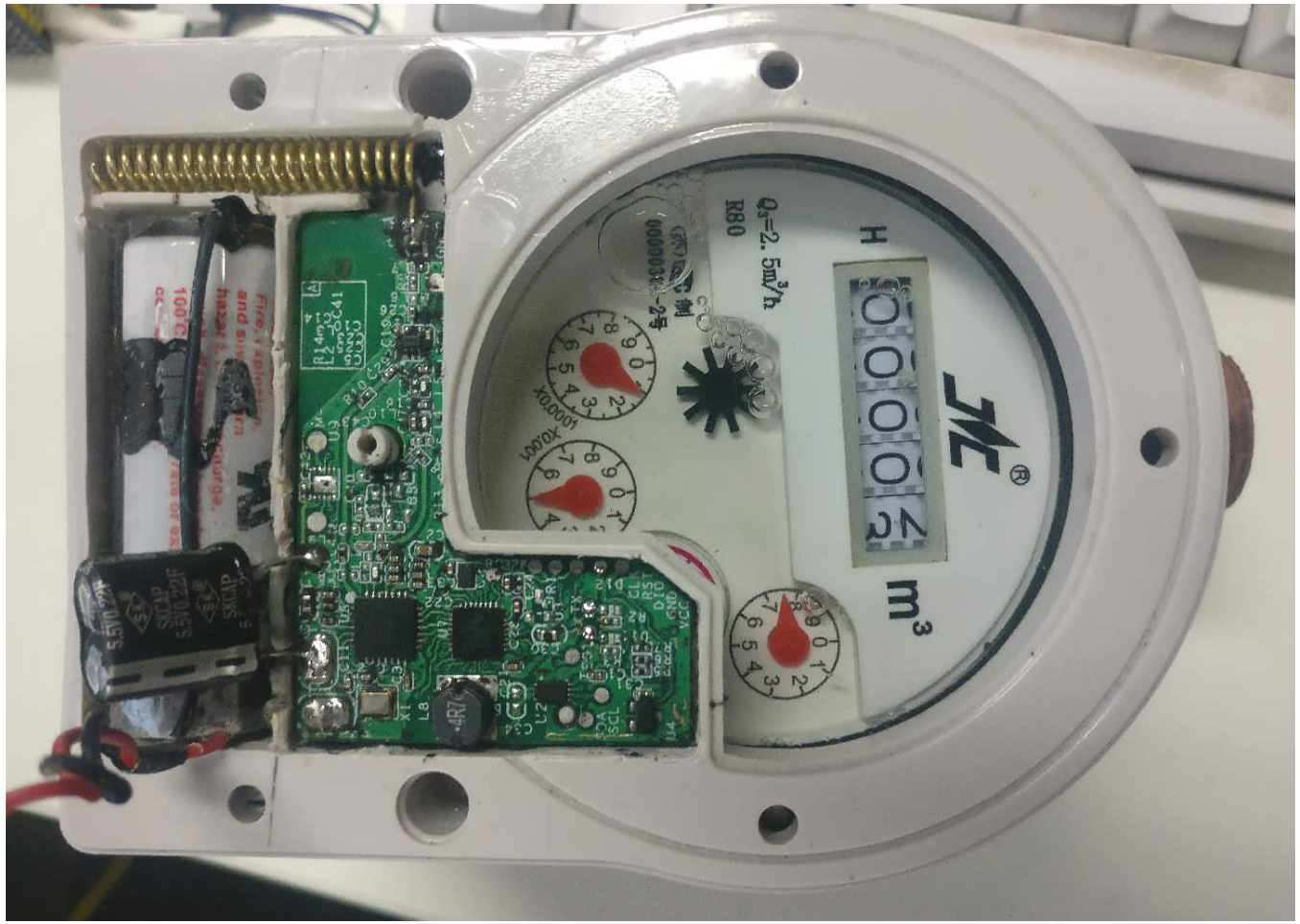
Unicorn.360.com/en

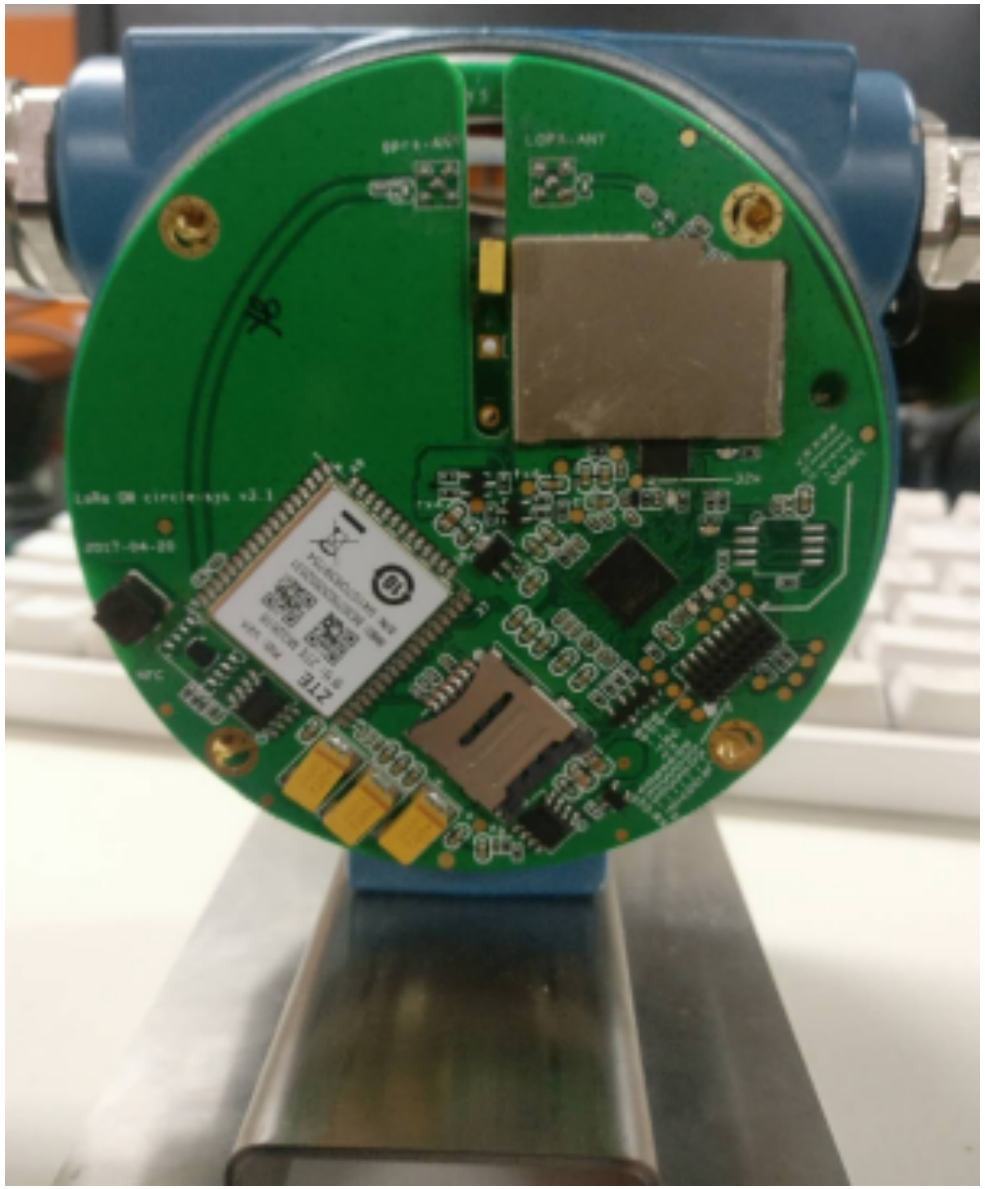
Real life Scenarios





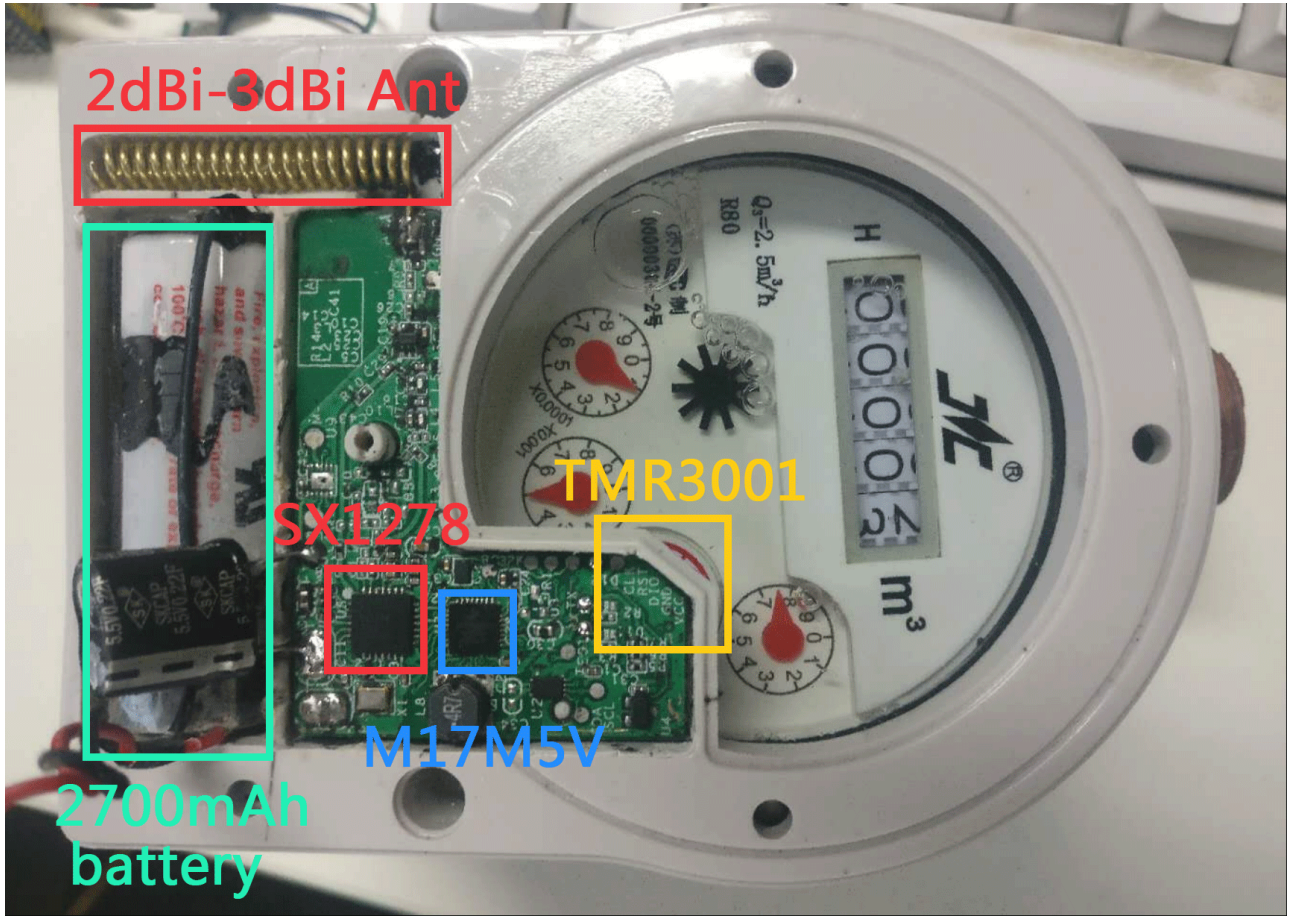
What does it look like?





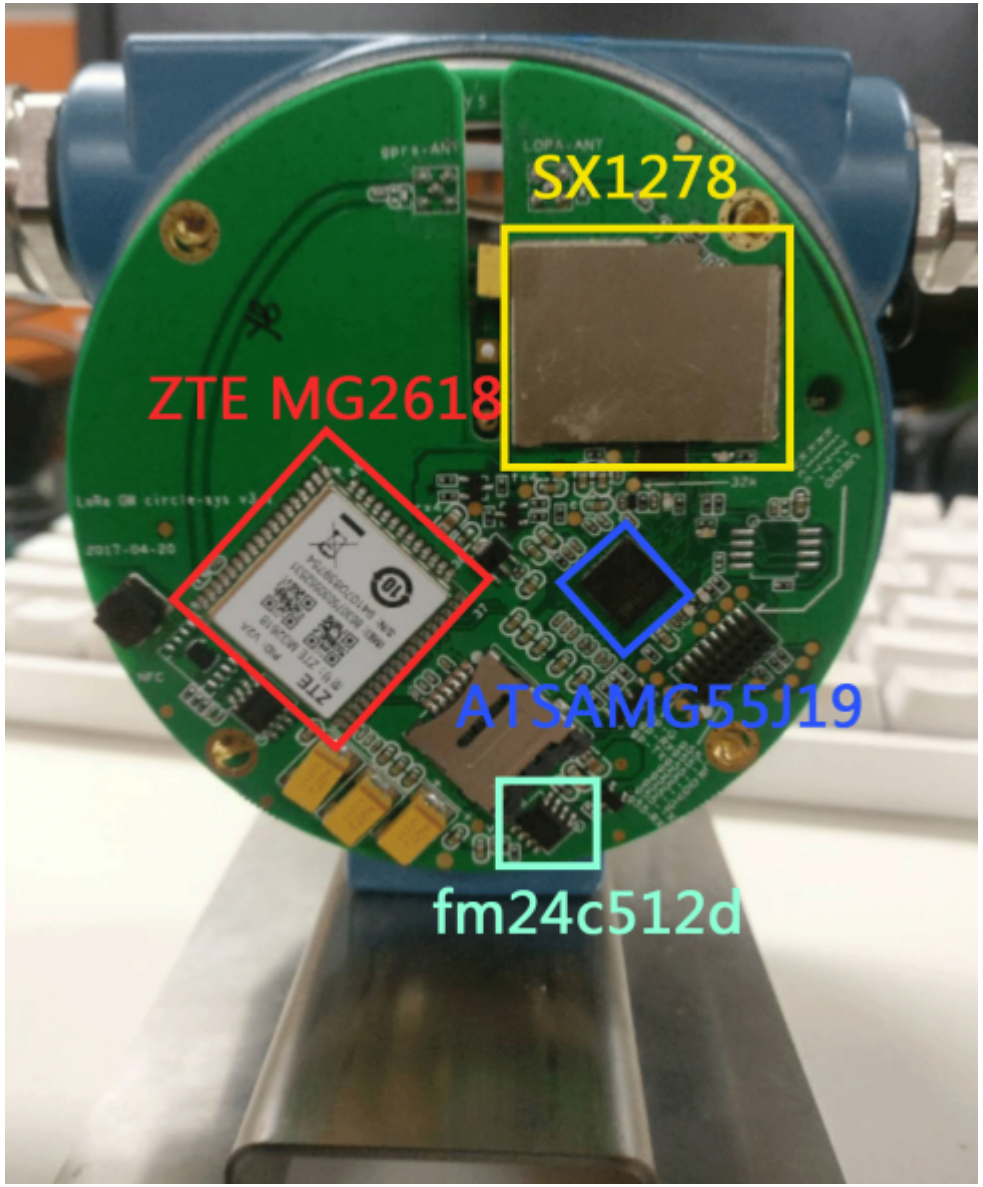
Water meter hardware disassembly

- TMR3001(Magnet sensor)
- M17M5V (NXP MCU)
- SX1278 (LORA)
- 2dBi-3dBi (ANT)
- 2700mAh (Battery)



Gateway hardware disassembly

- ZTE MG2618 (GPRS)
- ATSAMG55J19 (MCU)
- SX1278 (LORA)
- FM24c512D (EEPROM)
- RT9048

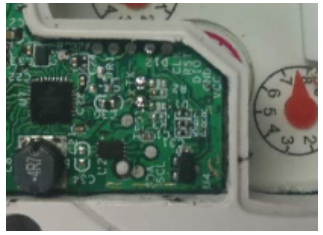


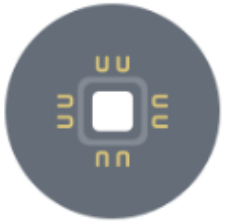
How does a water meter work?



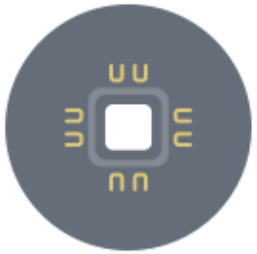


Magnet





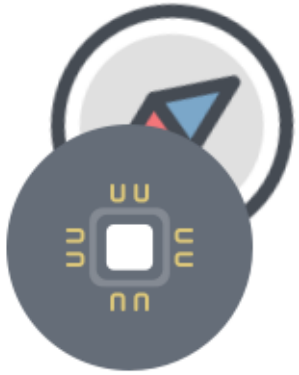
Sensor



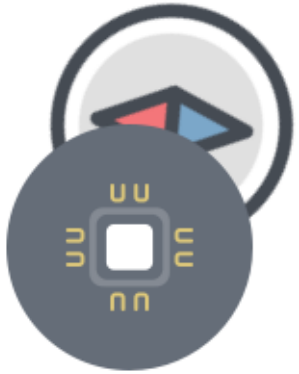
TMR3001
Magnet sensor



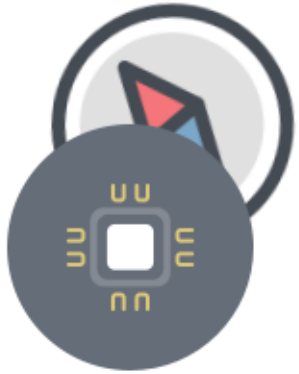
Magnet



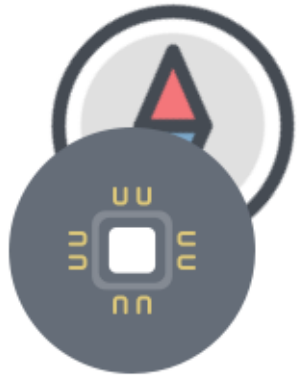
Magnet Voltage:
400 mv



Magnet Voltage:
20mv

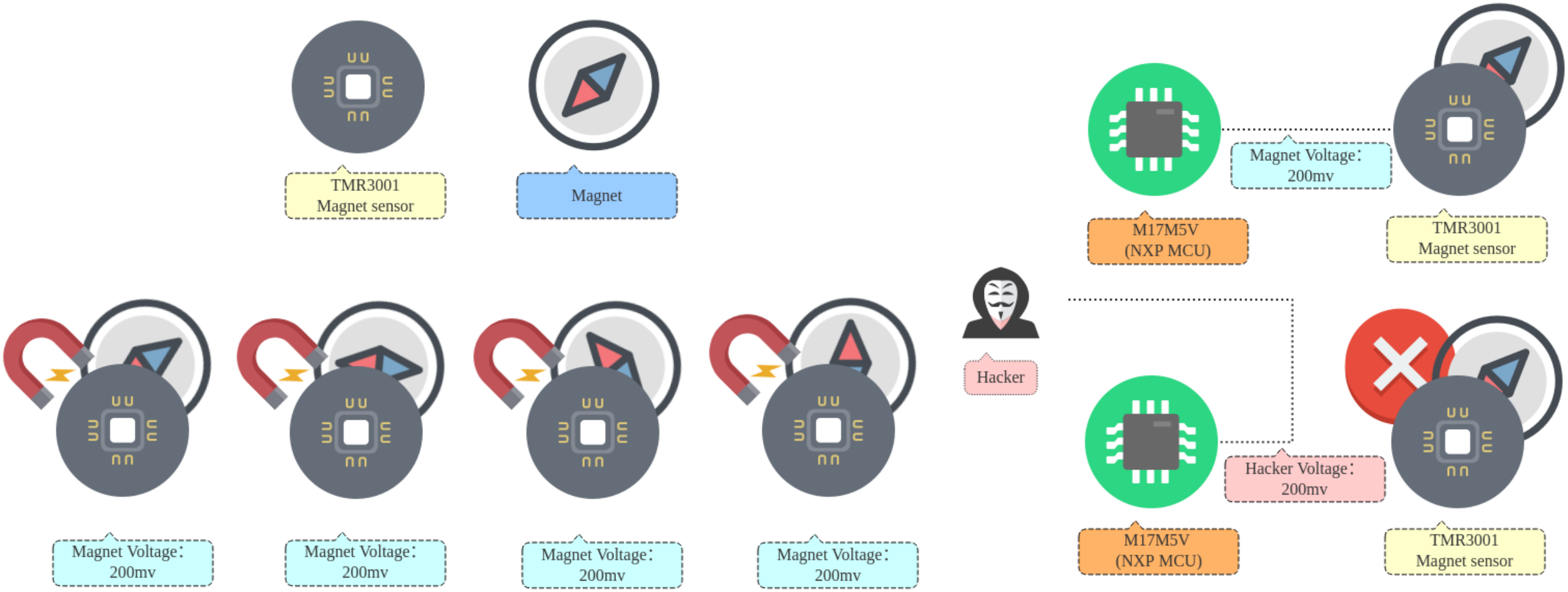


Magnet Voltage:
-200 mv

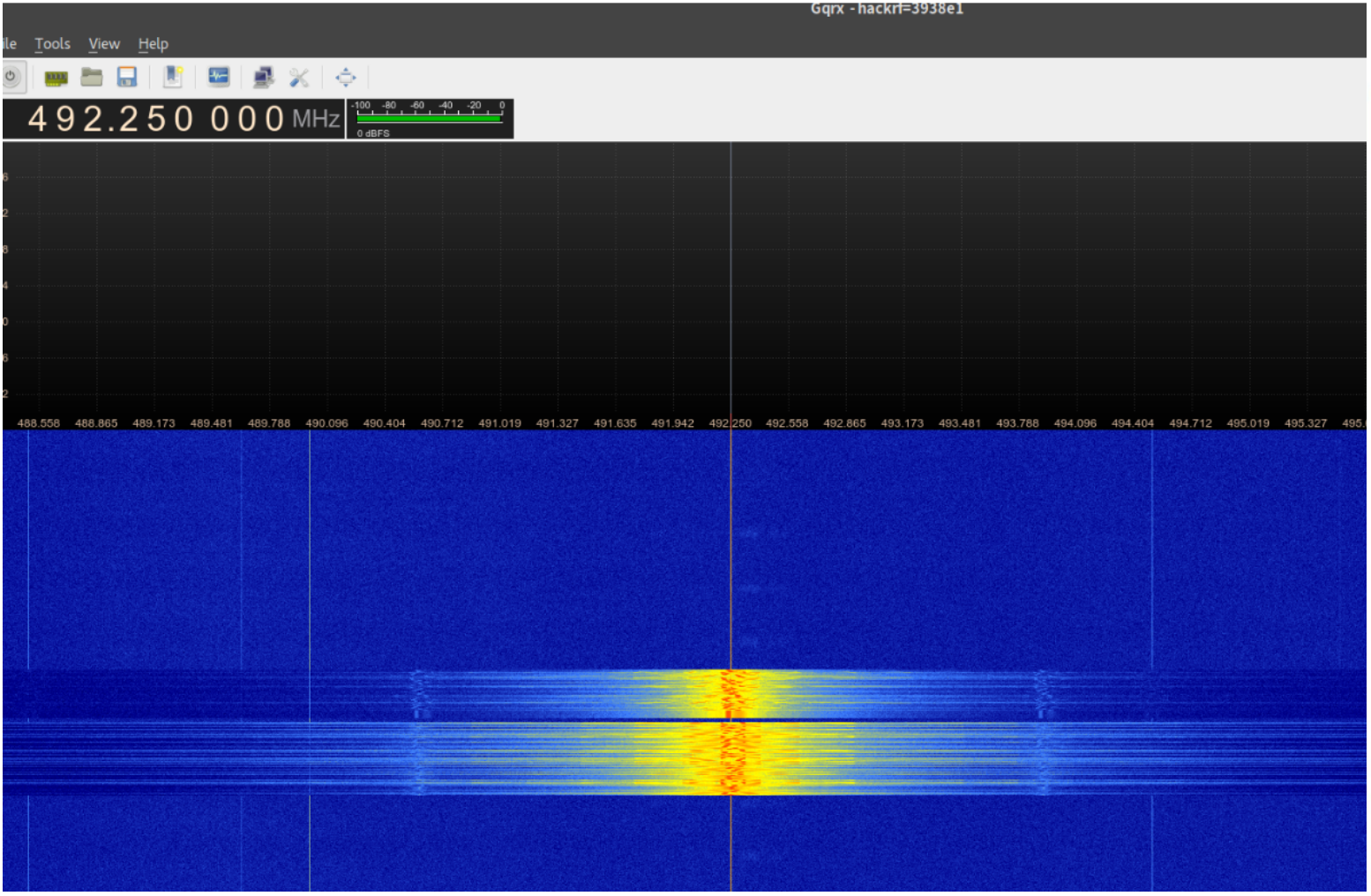


Magnet Voltage:
-300 mv

Attack the sensor

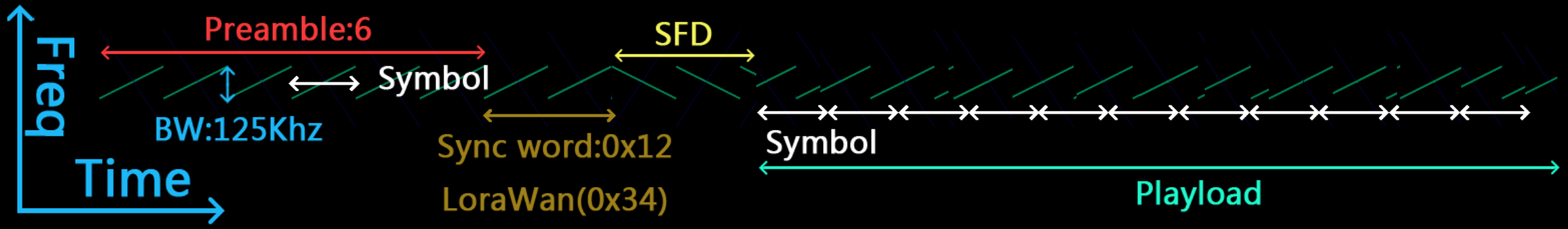


LORA Frequency



Countries	Frequency band review	Max. output power
EU	868 MHz	14 dBm
USA	915 MHz	20 dBm
Korea	900 MHz	14 dBm
Japan	920 MHz	
Malaysia	862 to 875 MHz	20 dBm
Philippines	868 MHz	
Vietnam	920 to 925 MHz	
India	865 to 867 MHz	
Singapore	922 MHz	
Thailand	920 to 925 MHz	
Indonesia	922 MHz	
ANZ	915 to 928 MHz	
Taiwan	920 to 925 MHz	
China	470 to 510 MHz	17 dBm

LORA Format



Bandwidth:125Khz(BW)

Preamble:6

Sync word:0x12

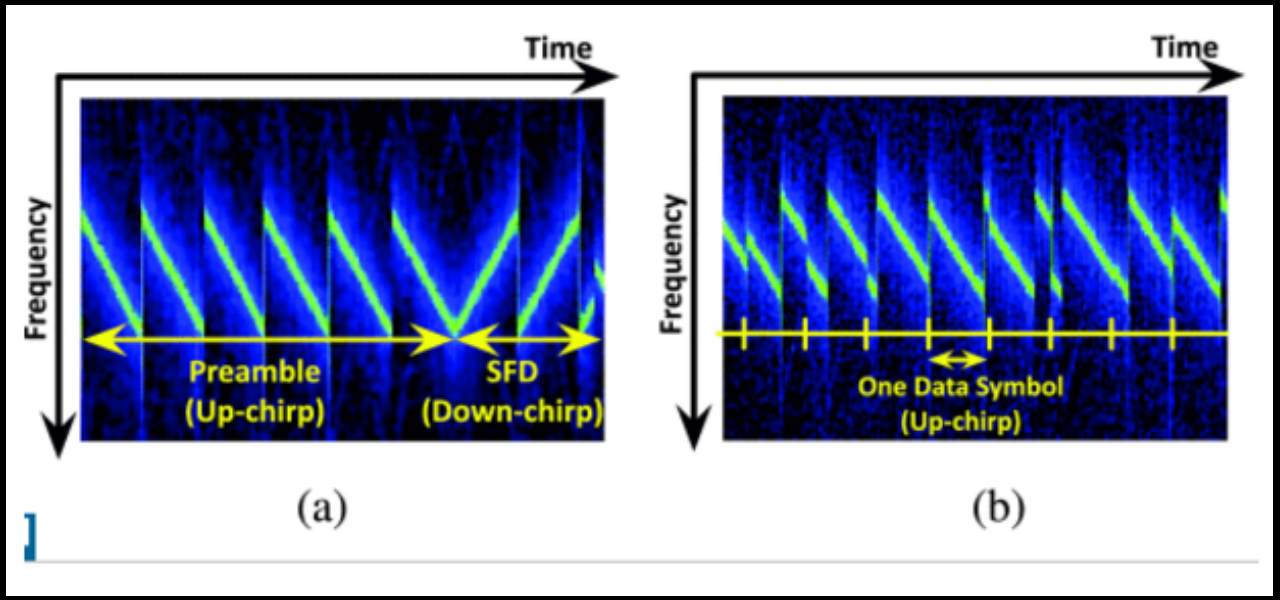
CodingRate:4/5(CR)

Spreading factor:8 (SF)

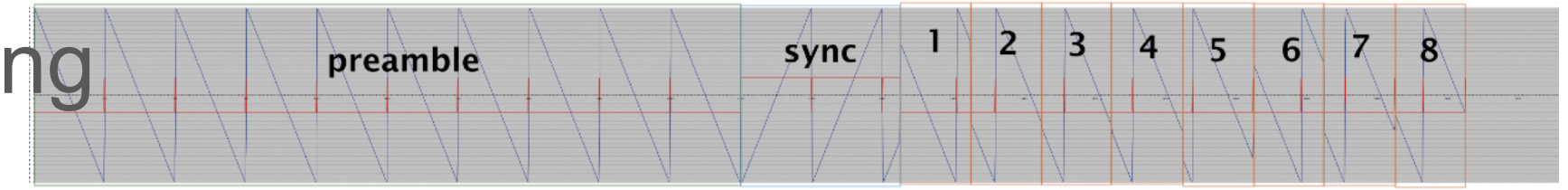
Symbol=BW/(2^SF)

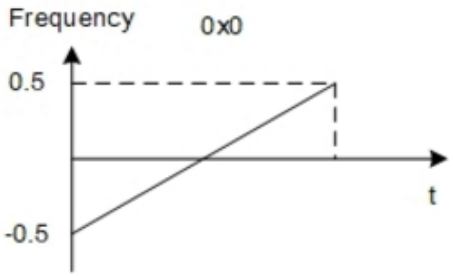
DownChirp

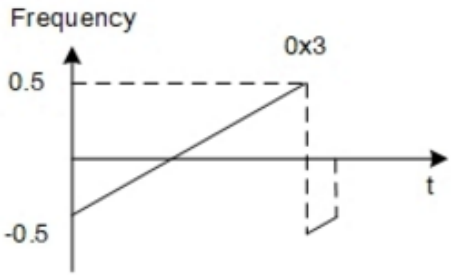
UpChirp

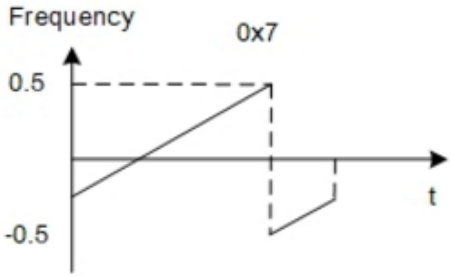


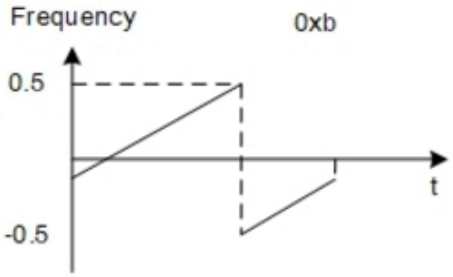
Playload encoding

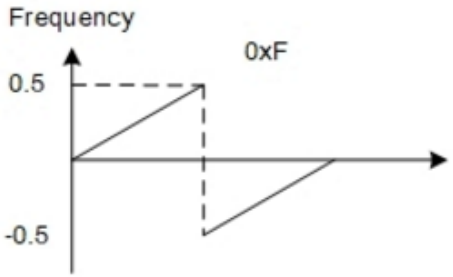








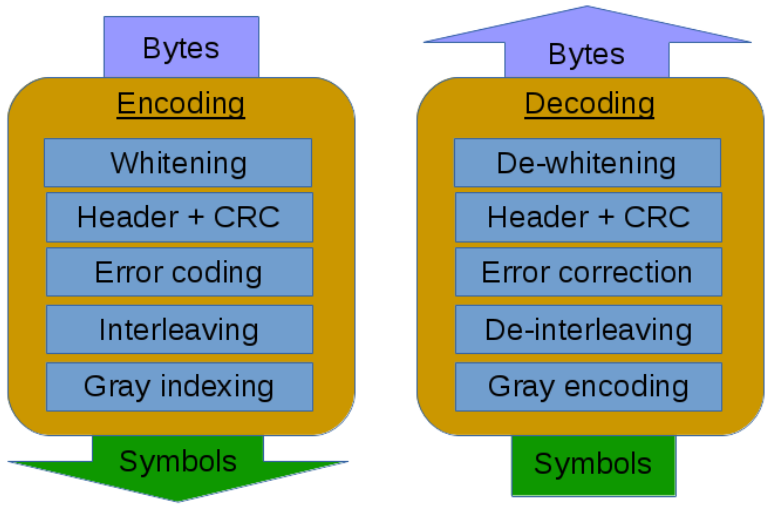




http://blog.csdn.net/ronhu

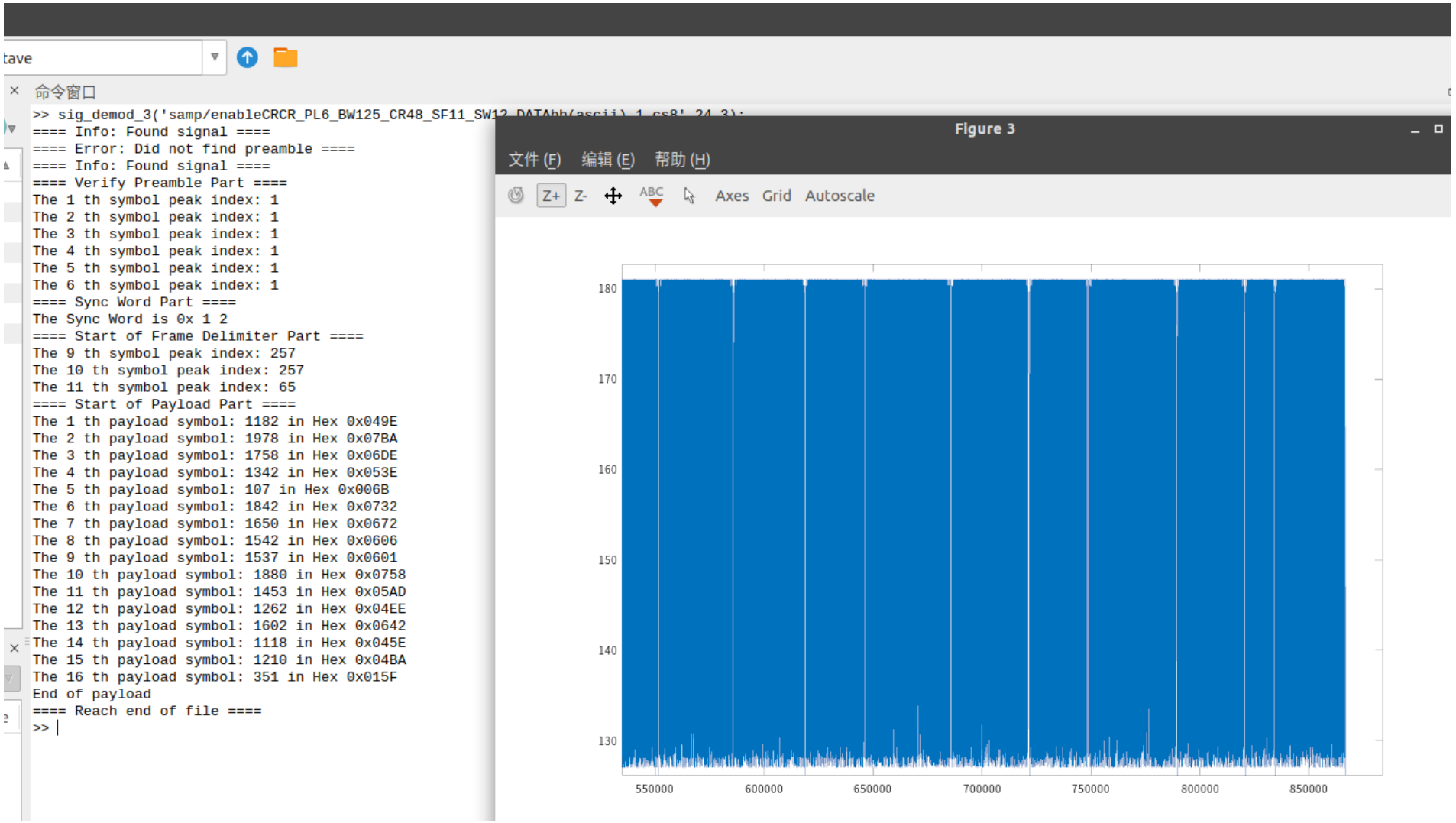
Code rate	Error Correction [bits]	Error detection [bits]
4/5	0	0
4/6	0	1
4/7	1	2
4/8	1	3

Table 2.1: Error correction and detection capabilities of LoRa



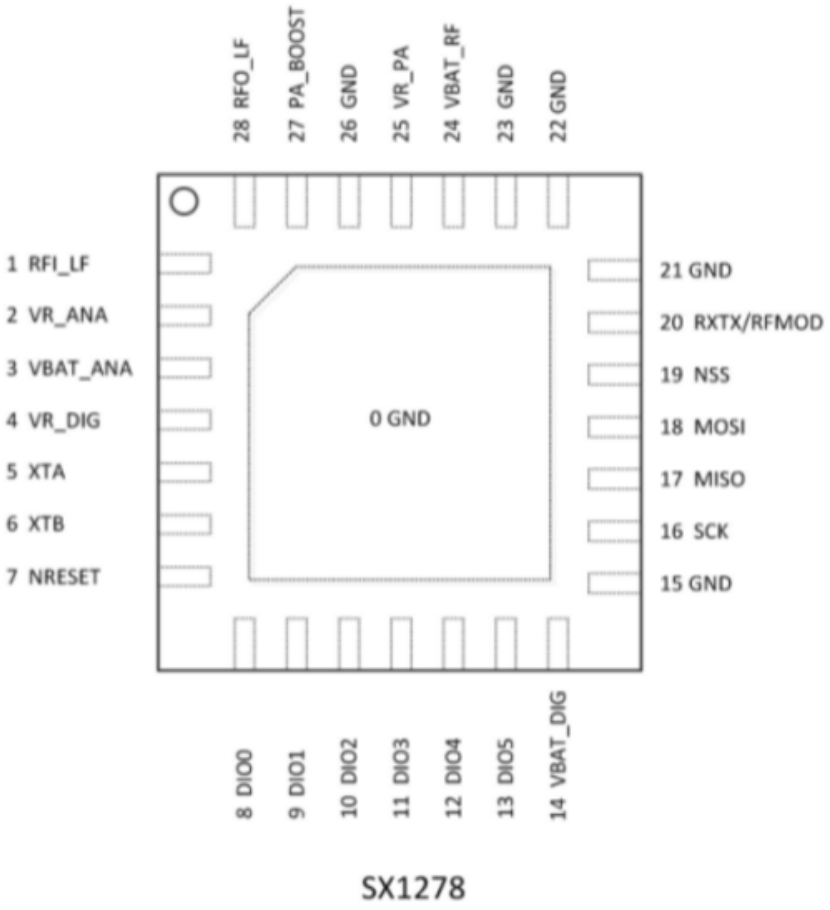
Chirp
Whitening
Error encoder
Interleaver
Gray

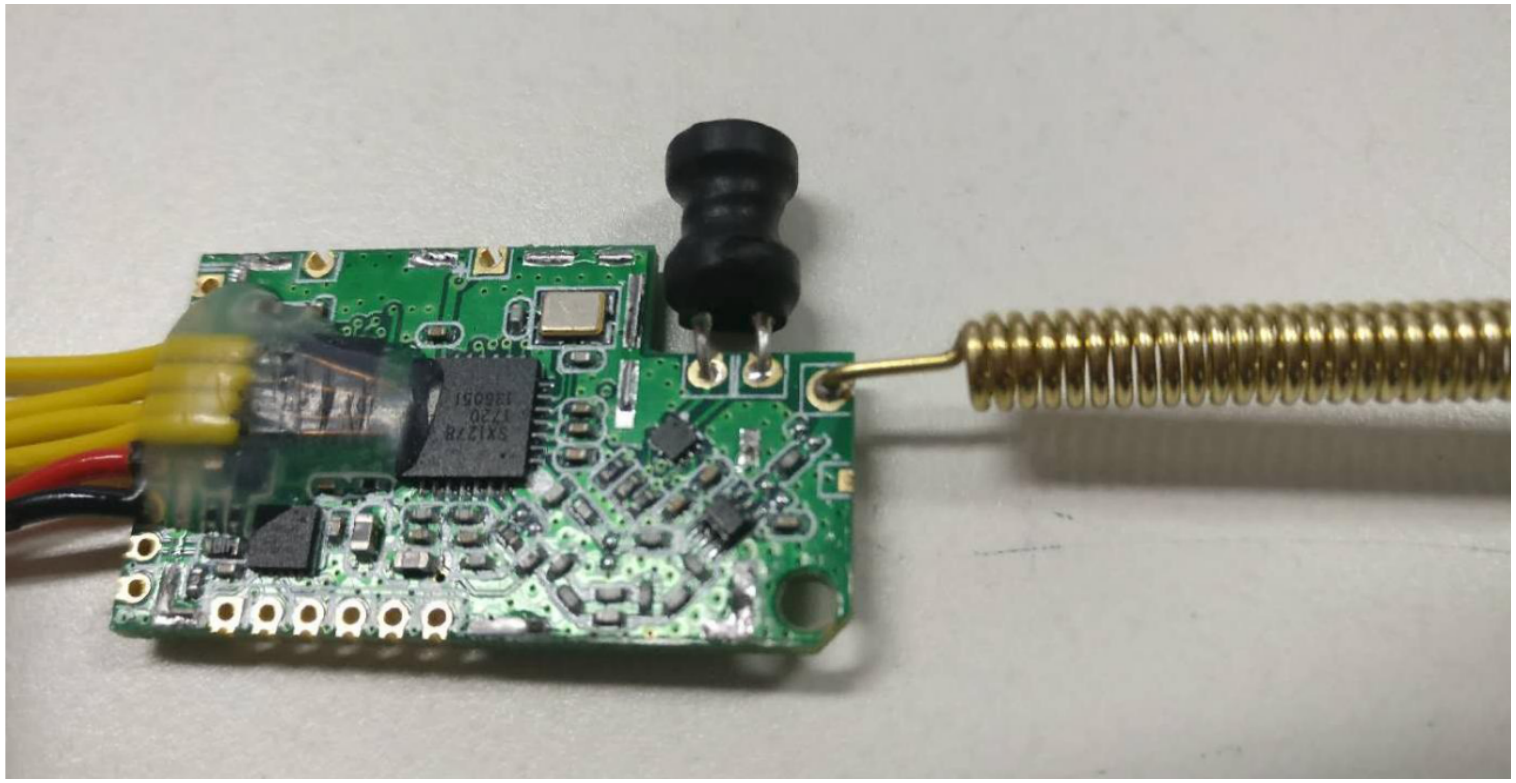
Demodulation lora



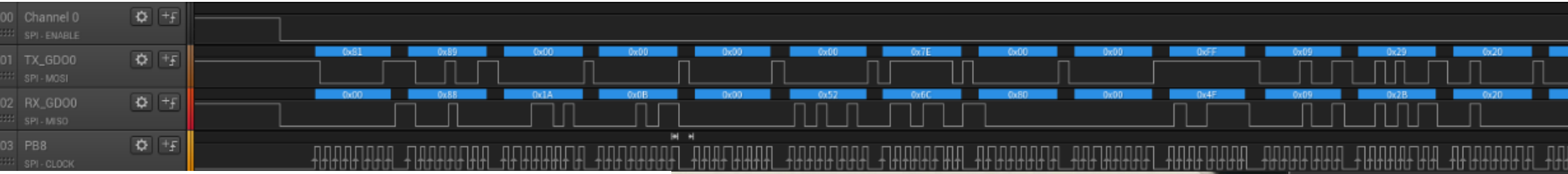
github.com/UnicornTeam/lora-demodlora

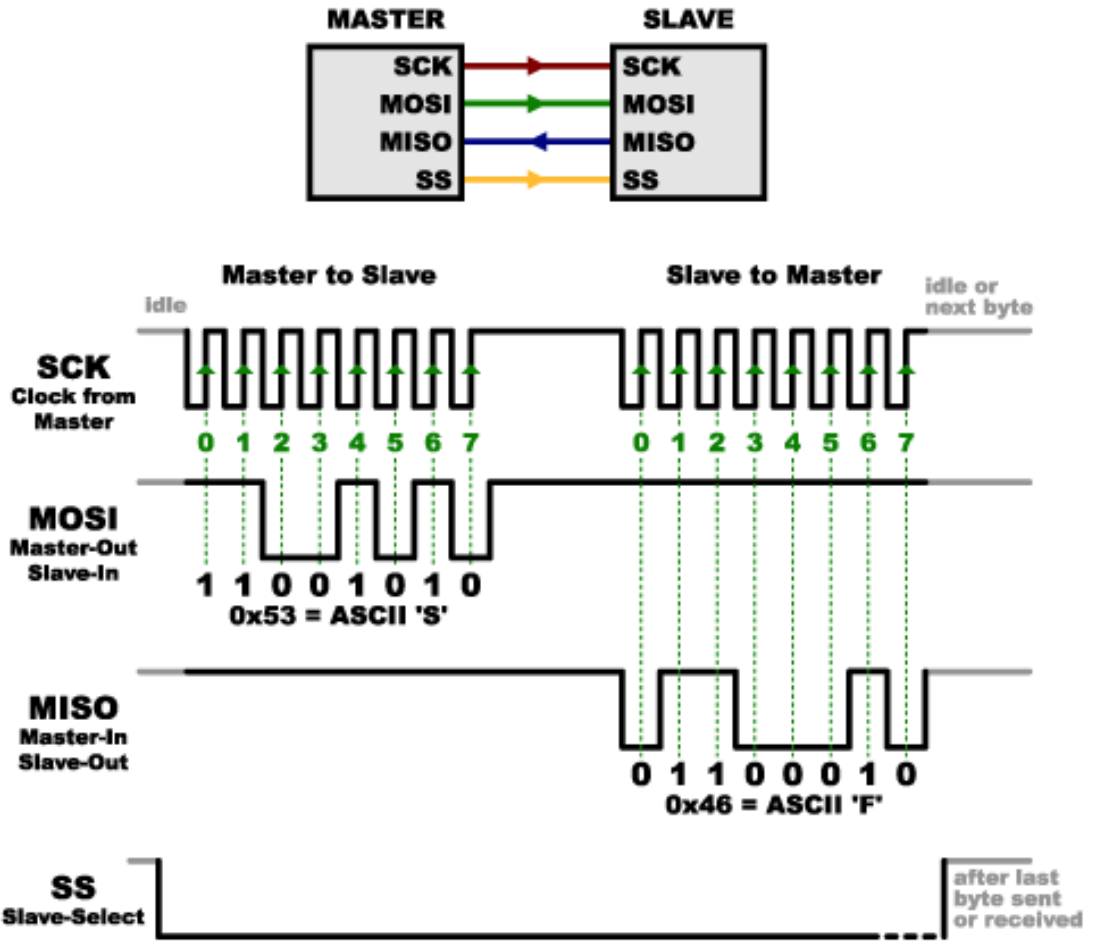
RF chip SPI sniffer

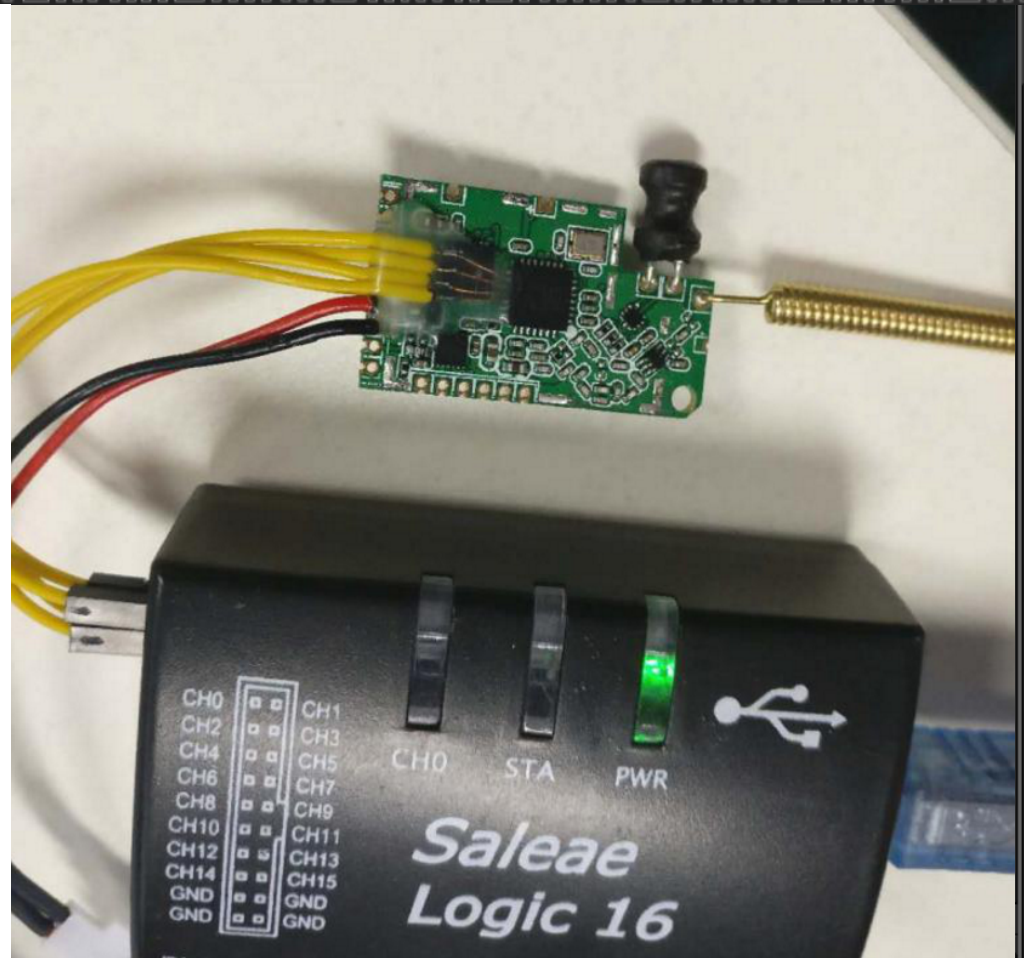




RF chip SPI sniffer









RF chip instructions reverse

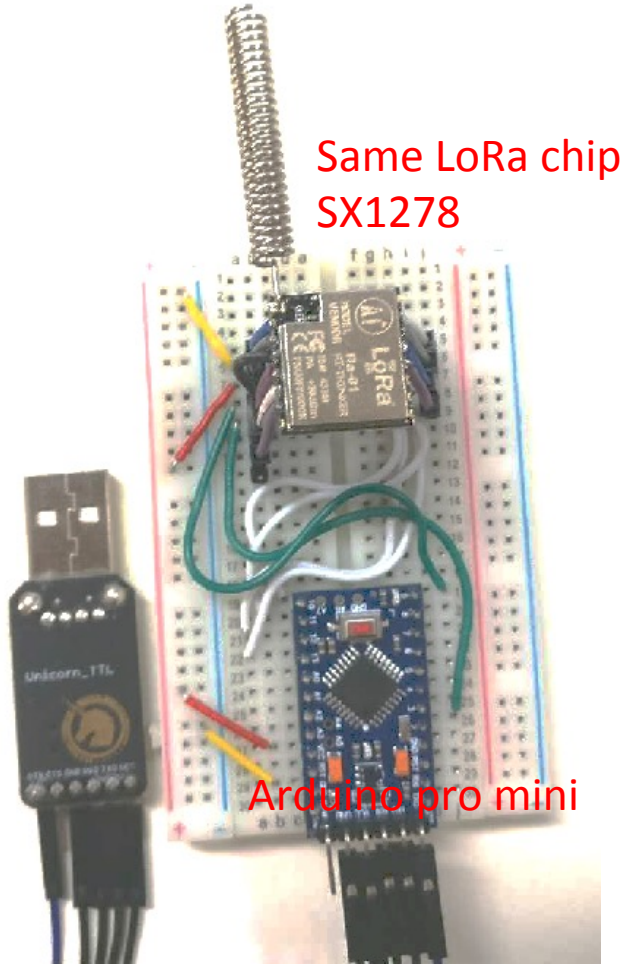
MOSI: 0x86; MISO: 0x68
MOSI: 0x7B; MISO: 0x6C
MOSI: 0x87; MISO: 0x68
MOSI: 0x0F; MISO: 0x80
MOSI: 0x88; MISO: 0x68
MOSI: 0xFF; MISO: 0x00
...
MOSI: 0x81; MISO: 0x68
MOSI: 0x81; MISO: 0x80
MOSI: 0xB1; MISO: 0x68
MOSI: 0xC3; MISO: 0xC3
MOSI: 0xB7; MISO: 0x68
MOSI: 0x0A; MISO: 0x0A
MOSI: 0x1E; MISO: 0x68
MOSI: 0x00; MISO: 0x70
MOSI: 0x9E; MISO: 0x68
MOSI: 0x70; MISO: 0x70
MOSI: 0x1D; MISO: 0x68
MOSI: 0x00; MISO: 0x72
MOSI: 0x9D; MISO: 0x68
MOSI: 0x72; MISO: 0x72

Address	Register Name		Reset (POR)	Default (FSK)	Description	
	FSK/OOK Mode	LoRa™ Mode			FSK Mode	LoRa™ Mode
0x00	RegFifo		0x00		FIFO read/write access	
0x01	RegOpMode		0x01		Operating mode & LoRa™ / FSK selection	
0x02	RegBitrateMsb	Unused	0x1A		Bit Rate setting, Most Significant Bits	
0x03	RegBitrateLsb		0x0B		Bit Rate setting, Least Significant Bits	
0x04	RegFdevMsb		0x00		Frequency Deviation setting, Most Significant Bits	
0x05	RegFdevLsb		0x52		Frequency Deviation setting, Least Significant Bits	
0x06	RegFrfMsb		0x6C		RF Carrier Frequency, Most Significant Bits	
0x07	RegFrfMid		0x80		RF Carrier Frequency, Intermediate Bits	
0x08	RegFrfLsb		0x00		RF Carrier Frequency, Least Significant Bits	
0x09	RegPaConfig		0x4F		PA selection and Output Power control	
0x0A	RegPaRamp		0x09		Control of PA ramp time, low phase noise PLL	
0x0B	RegOcp		0x2B		Over Current Protection control	
0x0C	RegLna		0x20		LNA settings	
0x0D	RegRxConfig	RegFifoAddrPtr	0x08	0x0E	AFC, AGC, ctrl	FIFO SPI pointer
0x0E	RegRssiConfig	RegFifoTxBa-seAddr	0x02		RSSI	Start Tx data
0x0F	RegRssiCollision	RegFifoRxBa-seAddr	0x0A		RSSI Collision detector	Start Rx data
0x10	RegRssiThresh	FifoRxCurren-tAddr	0xFF		RSSI Threshold control	Start address of last packet received
0x11	RegRssiValue	RegIrqFlagsMask	n/a	n/a	RSSI value in dBm	Optional IRQ flag mask
0x12	RegRxBw	RegIrqFlags	0x15		Channel Filter BW Control	IRQ flags
0x13	RegAfcBw	RegRxNbBytes	0x0B		AFC Channel Filter BW	Number of received bytes
0x14	RegOokPeak	RegRxHeaderCnt ValueMsb	0x28		OOK demodulator	Number of valid headers received
0x15	RegOokFix	RegRxHeaderCnt ValueLsb	0x0C		Threshold of the OOK demod	
0x16	RegOokAvg	RegRxPacketCnt ValueMsb	0x12		Average of the OOK demod	Number of valid packets received
0x17	Reserved17	RegRxPacketCnt ValueLsb	0x47		-	
0x18	Reserved18	RegModemStat	0x32		-	Live LoRaTM modem status
0x19	Reserved19	RegPktSnrValue	0x3E		-	Espimation of last packet SNR
0x1A	RegAfcFei	RegPktRssiValue	0x00		AFC and FEI control	RSSI of last packet

RF chip instructions reverse

```
MOSI: 0x42;  MISO: 0x00
MOSI: 0x00;  MISO: 0x12      //Read VERSION
MOSI: 0x86;  MISO: 0x68
MOSI: 0x7B;  MISO: 0x6C
MOSI: 0x87;  MISO: 0x68
MOSI: 0x0F;  MISO: 0x80
MOSI: 0x88;  MISO: 0x68
MOSI: 0xFF;  MISO: 0x00      // Set frequency 492.25 Mhz
MOSI: 0xA6;  MISO: 0x68
MOSI: 0x04;  MISO: 0x04      // Set the AGC LNA gain
MOSI: 0x81;  MISO: 0x68
MOSI: 0x81;  MISO: 0x80      // Enter IDLE
MOSI: 0xB1;  MISO: 0x68      // writeRegister(REG_DETECTION_OPTIMIZE, 0xc3);
MOSI: 0xC3;  MISO: 0xC3      // Set SF
MOSI: 0xB7;  MISO: 0x68      // writeRegister(REG_DETECTION_THRESHOLD, 0x0a);
MOSI: 0x0A;  MISO: 0x0A      // Set SF
MOSI: 0x1E;  MISO: 0x68      // readRegister(REG_MODEM_CONFIG_2) & 0x0f)
MOSI: 0x00;  MISO: 0x70      // Set SF
MOSI: 0x9E;  MISO: 0x68      // writeReg(readRegister(REG_MODEM_CONFIG_2) & 0x0f) | ((sf << 4) & 0xf0));
MOSI: 0x70;  MISO: 0x70      // Set SF 7
MOSI: 0x1D;  MISO: 0x68      // writeReg (readRegister(REG_MODEM_CONFIG_1) & 0x0f)
MOSI: 0x00;  MISO: 0x72      // Set BW CR Headr
MOSI: 0x9D;  MISO: 0x68      // writeReg (readRegister(REG_MODEM_CONFIG_1) & 0x0f) | (bw << 4));
MOSI: 0x72;  MISO: 0x72      // Set BW 125Khz CR4/5 Explicit Header mode
```

Lora Sniffer hardware



Sniffer_Watermeter_ESP32_simulation

```
  SPI.transfer(0x00);
  SPI.transfer(0x00);
  SPI.transfer(0x00);
  SPI.transfer(0x00);
  SPI.transfer(0x7E);
  SPI.transfer(0x00);
  SPI.transfer(0x00);
  SPI.transfer(0xFF);
  SPI.transfer(0x09);
  SPI.transfer(0x29);
  SPI.transfer(0x20);
  //LoRa.writeRegister(0x01, 0x88);
  //LoRa.writeRegister(0x01, 0x88);
}
void setup() {
  Serial.begin(9600);
  while (!Serial); //if just the the basic f
  delay(1000);

  Serial.println("LoRa Receiver");
  //Simulation_SPI();
  SPI.begin(SCK, MISO, MOSI, SS);
  LoRa.setPins(SS, RST, DI00);
  if (!LoRa.begin(BAND, PABOOST)) {
    Serial.println("Starting LoRa failed!");
    while (1);
  }
  LoRa.implicitHeaderMode();
  //LoRa.explicitHeaderMode();
  //LoRa.explicitHeaderMode();
  LoRa.setSpreadingFactor(11);
  LoRa.setCodingRate4(5);
  LoRa.setSignalBandwidth(125E3);
  LoRa.setPreambleLength(7);
  LoRa.setSyncWord(0x12);
  //LoRa.disableCrc();
  LoRa.crc();
  Serial.println("begin");
  LoRa.receive();
```

/dev/ttyUSB3

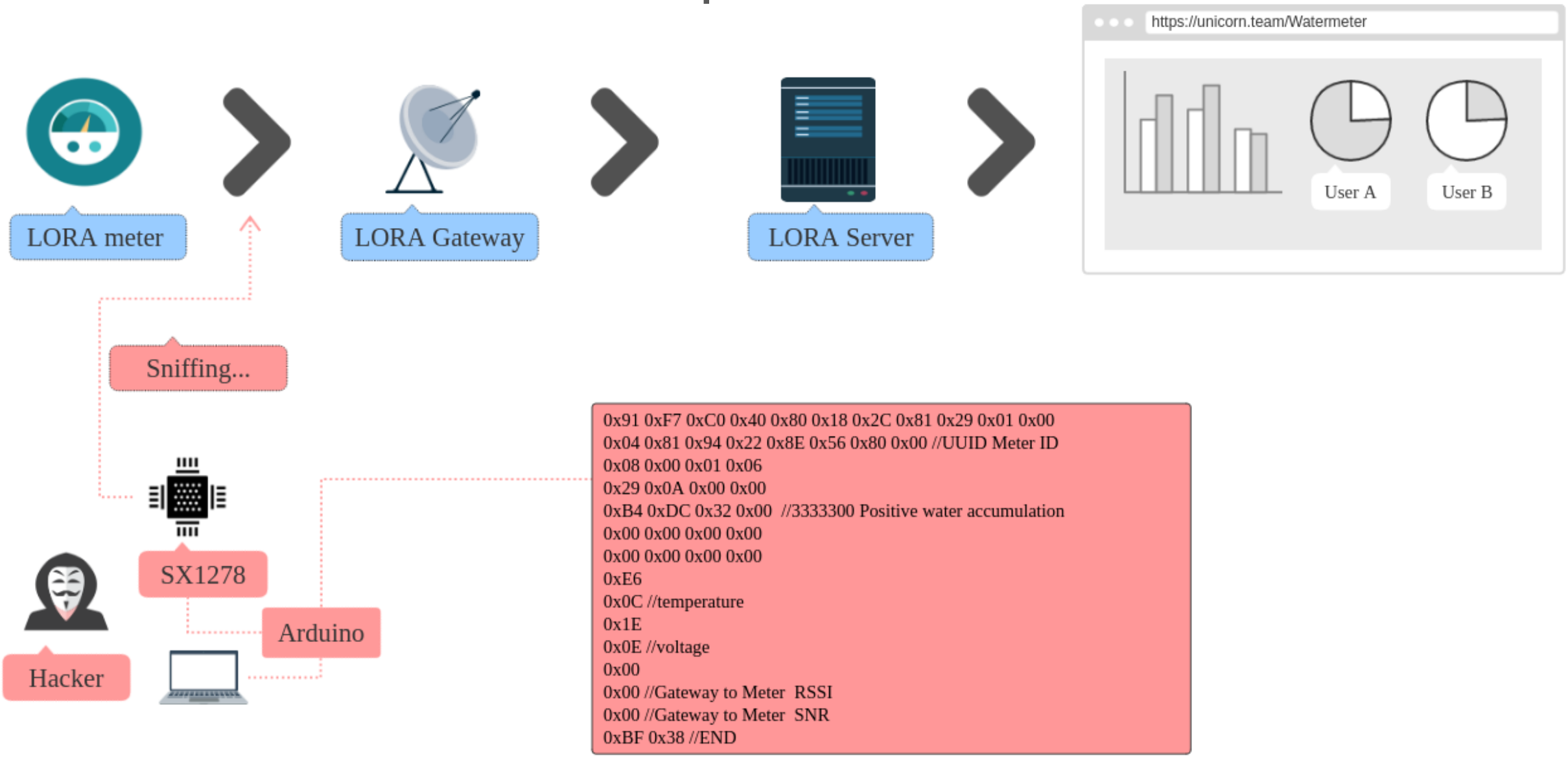
发送

```
LoRa Receiver
begin
Received packet '0x18 0x2C 0x81 0x29 0x01 0x00 0x04 0x81 0x94 0x22 0x8
Received packet '0x18 0x2C 0x81 0x29 0x01 0x00 0x04 0x81 0x94 0x22 0x8
Received packet '0x18 0x2C 0x81 0x29 0x01 0x00 0x04 0x81 0x94 0x22 0x8
```

自动滚屏 | 没有结束符 | 9600 波特率 | Clear output

Then we config our sniffer module using the parameters snooped from the SPI

Reverse communication protocol



Privacy risks





User A



User B



User C

User A	User B	User C
8th 20:20	8th 21:20	17号 22:20
Dosage:2.1 T	Dosage:12.01 T	Dosage:102.12 T
•••	•••	•••
•••	•••	•••

0x04 0x81 ..... 0x80 0x02 //Meter ID
....
0x01 0x20 0x00 0x00 //Water Dosage
....
0x0C //TEMP
..
0x0E //POWER
...

0x04 0x81 ..... 0x80 0x01 //Meter ID
....
0xB4 0xDC 0x32 0x00 //Water Dosage
....
0x0C //TEMP
..
0x0E //POWER
...





LORA Gateway



SX1278

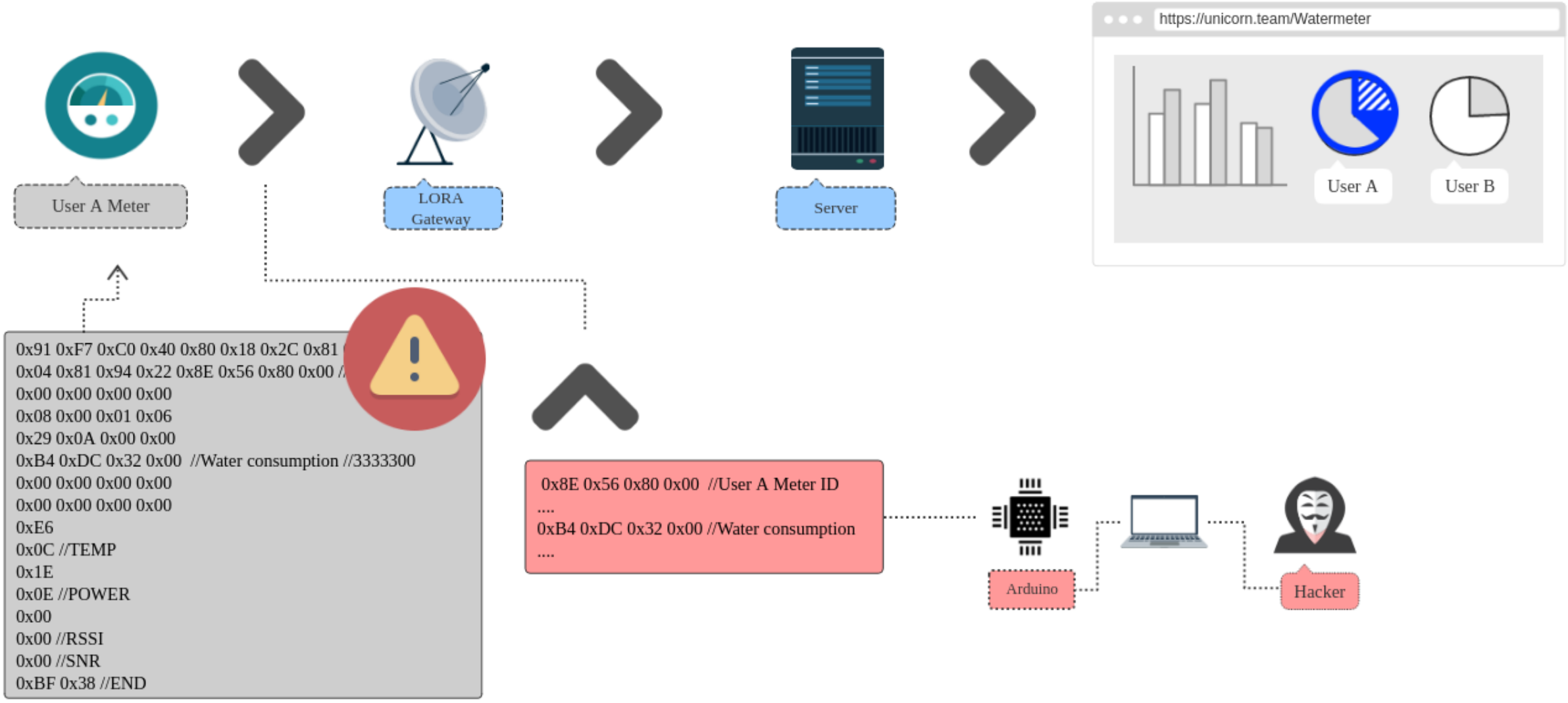
Arduino





Hacker

Fake upload data



Forge the gateway to issue instructions.



UserA LORA Meter

Meter ID: 0x8E 0x56 0x80 0x00



Gas Valve

Valve ID: 0x8E 0x56 0x80 0x01

Use sx1278 to enter sniff mode
....
Sniffer LORA Network ID info
....
//Sniffing the ID and then forging the gateway to transmit malicious instructions
....
Close .... ID 0x00 Valve
....
Close .... ID 0x01 Valve





Forged
LORA Gateway



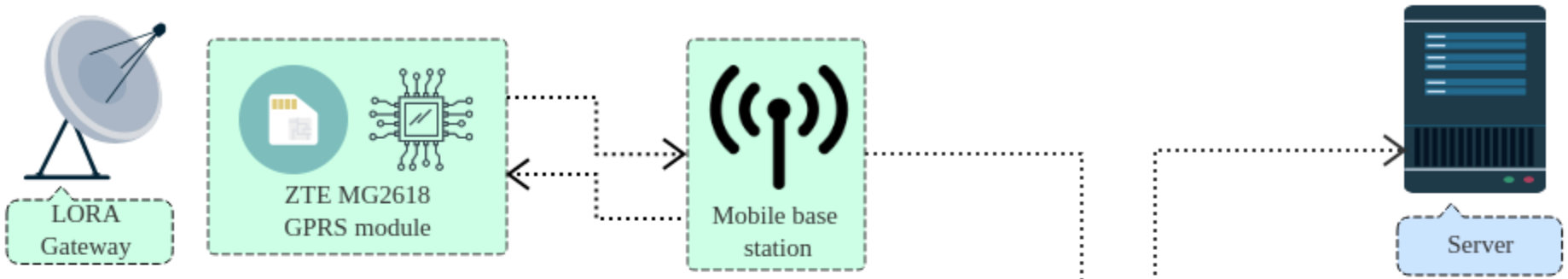
SX1278

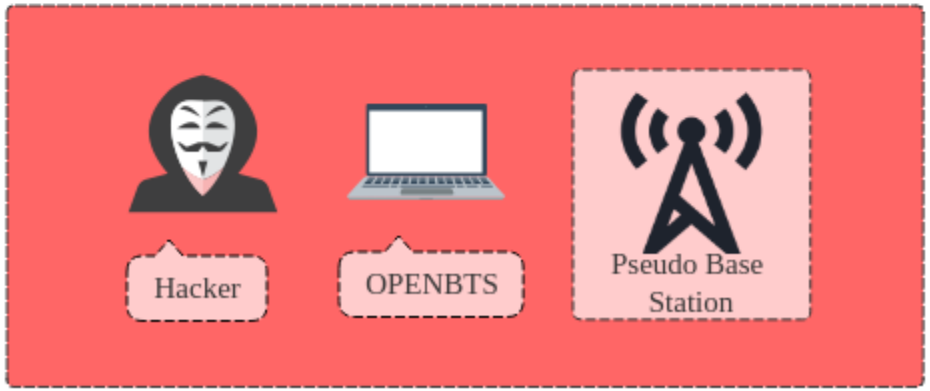




Hacker

GPRS communication link

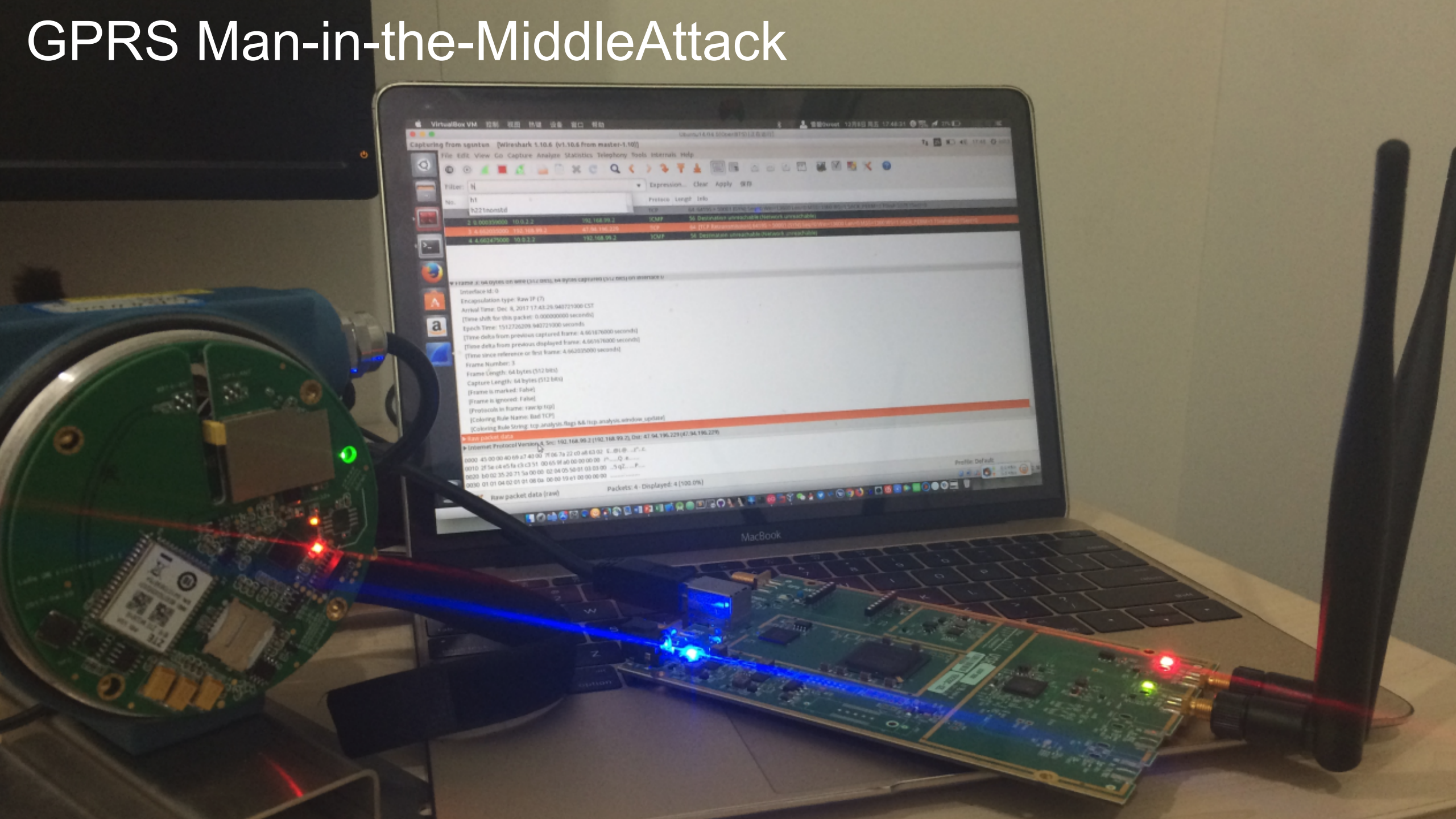


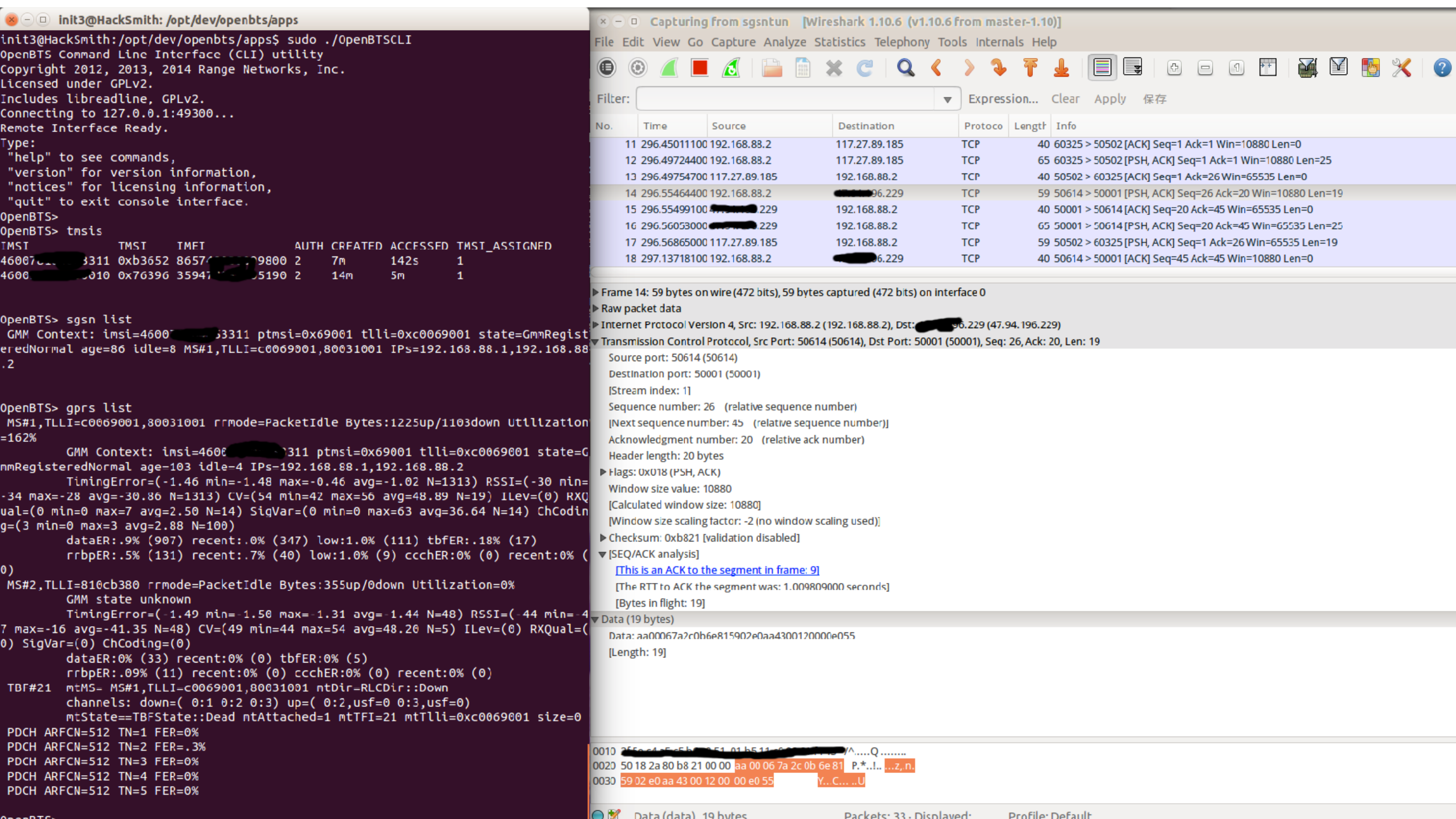


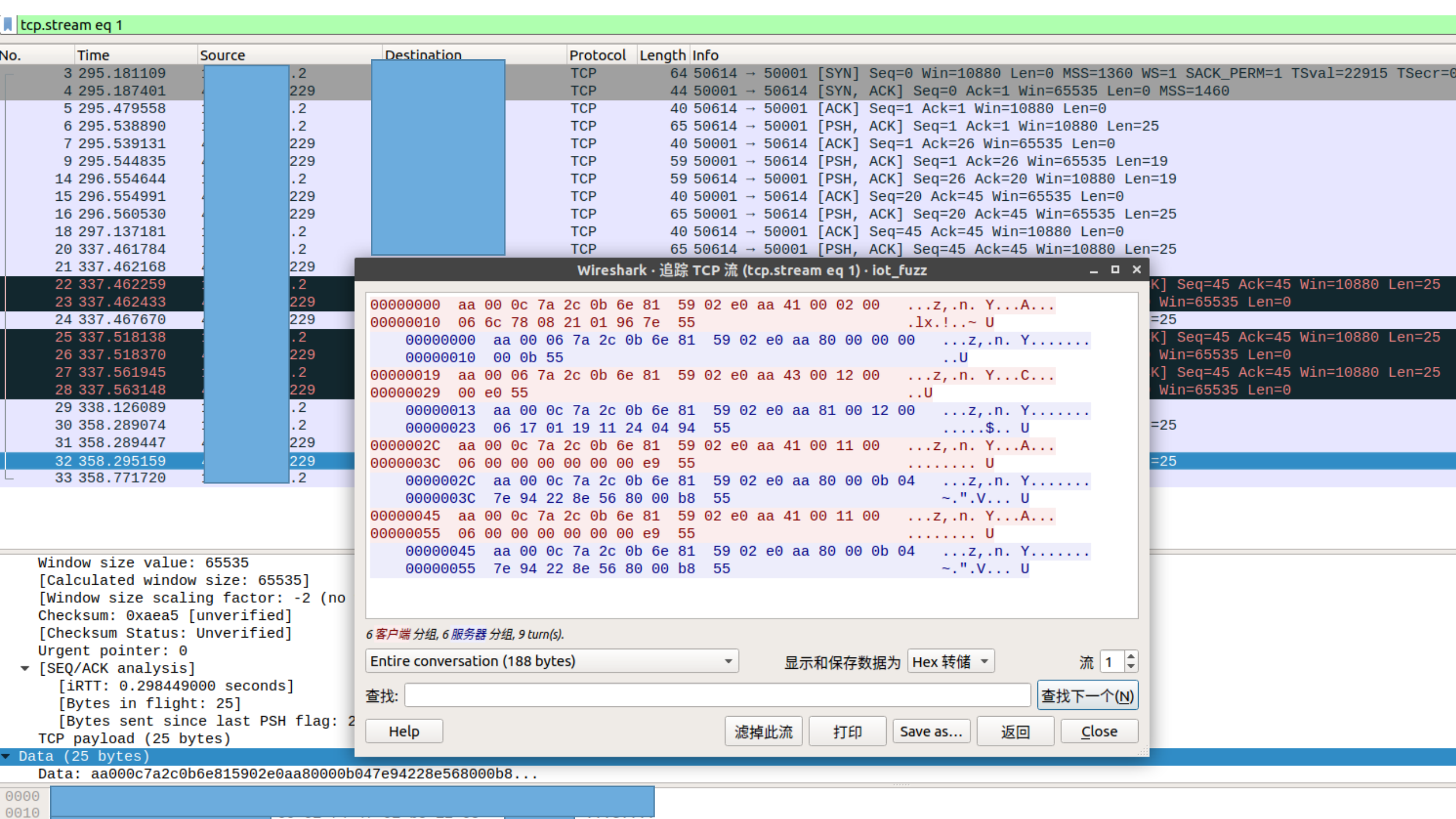
AA 00 28 7A 2C 0B 6E 81 59 02 E0 AA 42 00 1D 00 22 04 81 94 22 8E 56 80 00 00 32 DC B4 00 00 00 00 00 00 00 00 00 0E 0C E6 1E 17 01 19 11 25 65 0B 00 00 84 55

7A 2C 0B 6E 81 59 02 E0 //Gateway ID
AA //Header
42 //Cont
??00
1D //application
00 22 //Len
04 81 94 22 8E 56 80 00 //UUID Meter ID
00 32 DC B4 //3333300 Positive water accumulation
00 00 00 00/ /Negative cumulative water volume
00 00
00 00 //Meter State
00 0E //Voltage
0C //temperature
??: E6 1E
17 01 19 11 25 //Meter reading time
65 //Meter to Gateway RSSI
0b //Meter to Gateway SNR
00 //Gateway to Meter RSSI
00 //Gateway to Meter SNR
84 //CRC

GPRS Man-in-the-MiddleAttack




init3@HackSmith: /opt/dev/openbts/apps
init3@HackSmith:/opt/dev/openbts/apps$ sudo ./OpenBTSCLI
OpenBTS Command Line Interface (CLI) utility
Copyright 2012, 2013, 2014 Range Networks, Inc.
Licensed under GPLv2.
Includes libreadline, GPLv2.
Connecting to 127.0.0.1:49300...
Remote Interface Ready.
Type:
"help" to see commands,
"version" for version information,
"notices" for licensing information,
"quit" to exit console interface.
OpenBTS>
OpenBTS> tmsis
IMSI TMSI TMFI AUTH CREATED ACCESSED TMSI_ASSIGNED
OpenBTS> sgsn list
GMM Context: imsi=4600 3311 ptmsi=0x69001 tlli=0xc0069001 state=GmmRegistered
eredNormal age=86 idle=8 MS#1,TLLI=c0069001,80031001 IPs=192.168.88.1,192.168.88
.2
OpenBTS> gprs list
MS#1,TLLI=c0069001,80031001 rrmode=PacketIdle Bytes:1225up/1103down Utilization
=162%
GMM Context: imsi=4600 3311 ptmsi=0x69001 tlli=0xc0069001 state=G
nmRegisteredNormal age=103 idle=4 IPs=192.168.88.1,192.168.88.2
TimingError=(-1.46 min=-1.48 max=-0.46 avg=-1.02 N=1313) RSSI=(-30 min=
-34 max=-28 avg=-30.86 N=1313) CV=(54 min=42 max=56 avg=48.89 N=19) ILev=(0) RXQ
ual=(0 min=0 max=7 avg=2.50 N=14) SigVar=(0 min=0 max=63 avg=36.64 N=14) ChCodin
g=(3 min=0 max=3 avg=2.88 N=100)
dataER:.9% (907) recent:.0% (347) low:1.0% (111) tbfER:.18% (17)
rrbpER:.5% (131) recent:.7% (40) low:1.0% (9) ccchER:0% (0) recent:0% (
0)
MS#2,TLLI=810cb380 rrmode=PacketIdle Bytes:355up/0down Utilization=0%
GMM state unknown
TimingError=(-1.49 min=-1.50 max=-1.31 avg=-1.44 N=48) RSSI=(-44 min=-4
7 max=-16 avg=-41.35 N=48) CV=(49 min=44 max=54 avg=48.20 N=5) ILev=(0) RXQual=(
0) SigVar=(0) ChCoding=(0)
dataER:0% (33) recent:0% (0) tbfER:0% (5)
rrbpER:.09% (11) recent:0% (0) ccchER:0% (0) recent:0% (0)
TBF#21 mtMS- MS#1,TLLI=c0069001,80031001 mtDir=RLCDir::Down
channels: down=(0:1 0:2 0:3) up=(0:2,usf=0 0:3,usf=0)
mtState==TBFState::Dead ntAttached=1 mtTFI=21 mtTlli=0xc0069001 size=0
PDCH ARFCN=512 TN=1 FER=0%
PDCH ARFCN=512 TN=2 FER=.3%
PDCH ARFCN=512 TN=3 FER=0%
PDCH ARFCN=512 TN=4 FER=0%
PDCH ARFCN=512 TN=5 FER=0%
Capturing from sgsntun [Wireshark 1.10.6 (v1.10.6 from master-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: Expression... Clear Apply 保存
No. Time Source Destination Protoco Length Info
11 296.45011100 192.168.88.2 117.27.89.185 TCP 40 60325 > 50502 [ACK] Seq=1 Ack=1 Win=10880 Len=0
12 296.49724400 192.168.88.2 117.27.89.185 TCP 65 60325 > 50502 [PSH, ACK] Seq=1 Ack=1 Win=10880 Len=25
13 296.49754700 117.27.89.185 192.168.88.2 TCP 40 50502 > 60325 [ACK] Seq=1 Ack=26 Win=65535 Len=0
14 296.55464400 192.168.88.2 96.229 TCP 59 50614 > 50001 [PSH, ACK] Seq=26 Ack=20 Win=10880 Len=19
15 296.55499100 229 192.168.88.2 TCP 40 50001 > 50614 [ACK] Seq=20 Ack=45 Win=65535 Len=0
16 296.56053000 229 192.168.88.2 TCP 65 50001 > 50614 [PSH, ACK] Seq=20 Ack=45 Win=65535 Len=25
17 296.56865000 117.27.89.185 192.168.88.2 TCP 59 50502 > 60325 [PSH, ACK] Seq=1 Ack=26 Win=65535 Len=19
18 297.13718100 192.168.88.2 6.229 TCP 40 50614 > 50001 [ACK] Seq=45 Ack=45 Win=10880 Len=0
Frame 14: 59 bytes on wire (472 bits), 59 bytes captured (472 bits) on interface 0
Raw packet data
Internet Protocol Version 4, Src: 192.168.88.2 (192.168.88.2), Dst: 6.229 (47.94.196.229)
Transmission Control Protocol, Src Port: 50614 (50614), Dst Port: 50001 (50001), Seq: 26, Ack: 20, Len: 19
Source port: 50614 (50614)
Destination port: 50001 (50001)
[Stream index: 1]
Sequence number: 26 (relative sequence number)
[Next sequence number: 45 (relative sequence number)]
Acknowledgment number: 20 (relative ack number)
Header length: 20 bytes
Flags: 0x018 (PSH, ACK)
Window size value: 10880
[Calculated window size: 10880]
[Window size scaling factor: -2 (no window scaling used)]
Checksum: 0xb821 [validation disabled]
[SEQ/ACK analysis]
[This is an ACK to the segment in frame: 9]
[The RTT to ACK the segment was: 1.009809000 seconds]
[Bytes in flight: 19]
Data (19 bytes)
Data: aa00067a2c0b6e815902e0aa4300120000e055
[Length: 19]
0010
0020 50 18 2a 80 b8 21 00 00 aa 00 06 7a 2c 0b 6e 81 P.*..!.. ...z,.n.
0030 59 02 e0 aa 43 00 12 00 00 e0 55 Y...C... ..U
Data (data), 19 bytes
Packets: 33 · Displayed:
Profile: Default


tcp.stream eq 1
No. Time Source Destination Protocol Length Info
3 295.181109 .2 TCP 64 50614 → 50001 [SYN] Seq=0 Win=10880 Len=0 MSS=1360 WS=1 SACK_PERM=1 TSval=22915 TSecr=0
4 295.187401 229 TCP 44 50001 → 50614 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
5 295.479558 .2 TCP 40 50614 → 50001 [ACK] Seq=1 Ack=1 Win=10880 Len=0
6 295.538890 .2 TCP 65 50614 → 50001 [PSH, ACK] Seq=1 Ack=1 Win=10880 Len=25
7 295.539131 229 TCP 40 50001 → 50614 [ACK] Seq=1 Ack=26 Win=65535 Len=0
9 295.544835 229 TCP 59 50001 → 50614 [PSH, ACK] Seq=1 Ack=26 Win=65535 Len=19
14 296.554644 .2 TCP 59 50614 → 50001 [PSH, ACK] Seq=26 Ack=20 Win=10880 Len=19
15 296.554991 229 TCP 40 50001 → 50614 [ACK] Seq=20 Ack=45 Win=65535 Len=0
16 296.560530 229 TCP 65 50001 → 50614 [PSH, ACK] Seq=20 Ack=45 Win=65535 Len=25
18 297.137181 .2 TCP 40 50614 → 50001 [ACK] Seq=45 Ack=45 Win=10880 Len=0
20 337.461784 .2 TCP 65 50614 → 50001 [PSH, ACK] Seq=45 Ack=45 Win=10880 Len=25
21 337.462168 229
22 337.462259 .2 K] Seq=45 Ack=45 Win=10880 Len=25
23 337.462433 229 Win=65535 Len=0
24 337.467670 229 =25
25 337.518138 .2 K] Seq=45 Ack=45 Win=10880 Len=25
26 337.518370 229 Win=65535 Len=0
27 337.561945 .2 K] Seq=45 Ack=45 Win=10880 Len=25
28 337.563148 229 Win=65535 Len=0
29 338.126089 .2
30 358.289074 .2 =25
31 358.289447 229
32 358.295159 229 =25
33 358.771720 .2
Window size value: 65535
[Calculated window size: 65535]
[Window size scaling factor: -2 (no
Checksum: 0xaea5 [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0
[SEQ/ACK analysis]
[iRTT: 0.298449000 seconds]
[Bytes in flight: 25]
[Bytes sent since last PSH flag: 2
TCP payload (25 bytes)
Data (25 bytes)
Data: aa000c7a2c0b6e815902e0aa80000b047e94228e568000b8...
0000
0010
Wireshark · 追踪 TCP 流 (tcp.stream eq 1) · iot_fuzz
00000000 aa 00 0c 7a 2c 0b 6e 81 59 02 e0 aa 41 00 02 00 ...z,.n. Y...A...
00000010 06 6c 78 08 21 01 96 7e 55 .lx.!..~ U
00000000 aa 00 06 7a 2c 0b 6e 81 59 02 e0 aa 80 00 00 00 ...z,.n. Y.......
00000010 00 0b 55 ..U
00000019 aa 00 06 7a 2c 0b 6e 81 59 02 e0 aa 43 00 12 00 ...z,.n. Y...C...
00000029 00 e0 55 ..U
00000013 aa 00 0c 7a 2c 0b 6e 81 59 02 e0 aa 81 00 12 00 ...z,.n. Y.......
00000023 06 17 01 19 11 24 04 94 55 .....$.. U
0000002C aa 00 0c 7a 2c 0b 6e 81 59 02 e0 aa 41 00 11 00 ...z,.n. Y...A...
0000003C 06 00 00 00 00 00 00 e9 55 .... U
0000002C aa 00 0c 7a 2c 0b 6e 81 59 02 e0 aa 80 00 0b 04 ...z,.n. Y.......
0000003C 7e 94 22 8e 56 80 00 b8 55 ~."V... U
00000045 aa 00 0c 7a 2c 0b 6e 81 59 02 e0 aa 41 00 11 00 ...z,.n. Y...A...
00000055 06 00 00 00 00 00 00 e9 55 .... U
00000045 aa 00 0c 7a 2c 0b 6e 81 59 02 e0 aa 80 00 0b 04 ...z,.n. Y.......
00000055 7e 94 22 8e 56 80 00 b8 55 ~."V... U
6 客户端 分组, 6 服务器 分组, 9 turn(s).
Entire conversation (188 bytes)
显示和保存数据为 Hex 转储
流 1
查找:
查找下一个(N)
Help
滤掉此流
打印
Save as...
返回
Close

Reverse upload protocol

recv:AA 00 0C 7A 2C 0B 6E 81 59 02 E0 AA 41 00 02 00 06 6C 78 08 21 01 96 7E 55 //Gateway registration
Server:aa 00 06 7a 2c 0b 6e 81 59 02 e0 aa 80 00 00 00 00 0b 55 //Service returns the packet

recv:AA 00 06 7A 2C 0B 6E 81 59 02 E0 AA 43 00 12 00 00 E0 55 //Gateway request time
Server:aa 00 0c 7a 2c 0b 6e 81 59 02 e0 aa 81 00 12 00 06 17 01 19 11 24 04 94 55 //Server return time 17 01 19 11 24 04

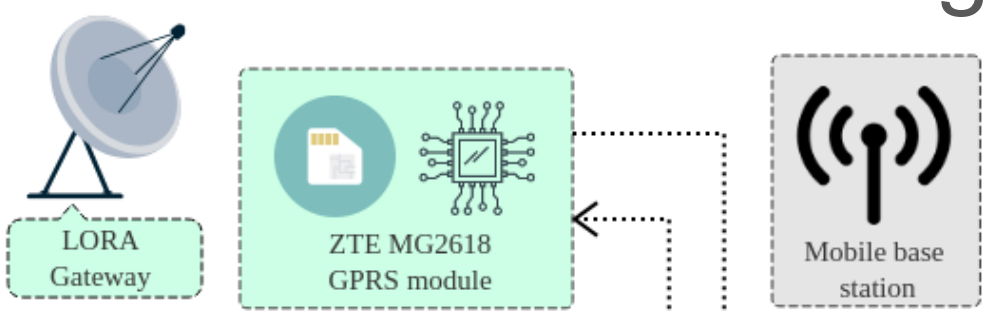
AA // Header
00 0C // Len
7A 2C 0B 6E 81 59 0 E0 // Gateway ID
AA // Header
41 // Cont Code
00 //
02 // Application
00 06 // Data length
6C 78 08 21 01 96 // Data (mcu version:6C 78 radio mcu:08 21 NID:01 CH:96(150))
7E // Check code =
(0c+7a+2c+0b+6e+81+59+02+e0+aa+41+02+06+6c+78+08+21+01+96)
55 // END

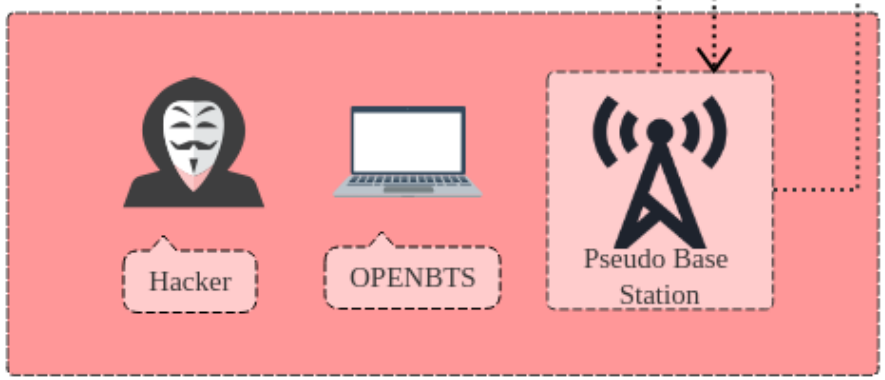
Reverse Meter upload protocol

AA 00 28 7A 2C 0B 6E 81 59 02 E0 AA 42 00 1D 00 22 04 81 94 22 8E 56 80 00 00 32 DC B4 00 00 00 00 00 00 00 00 00 0E 0C E6 1E 17 01 19 11 25 65 0B 00 00 84 55

7A 2C 0B 6E 81 59 02 E0	//Gateway ID
AA	//Header
42	//Cont
??00	
1D	//application
00 22	//Len
04 81 94 22 8E 56 80 00	//UUID Meter ID
00 32 DC B4	//3333300 Positive water accumulation
00 00 00 00	//00 00
00 00	//Meter State
00 0E	//Voltage
0C	//temperature
E6 1E	
17 01 19 11 25	//Meter reading time
65	//Meter to Gateway RSSI
0b	//Meter to Gateway SNR
00	//Gateway to Meter RSSI
00	//Gateway to Meter SNR
84	//CRC

GPRS module data sniffing



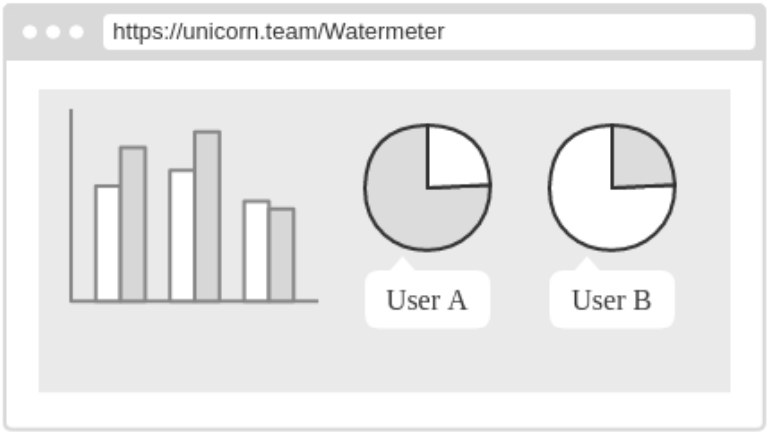


AA 00 28 7A 2C 0B 6E 81 59 02 E0 AA 42 00 1D 00 22 04 81 94 22 8E 56 80 00 00 32 DC B4 00 00 00 00 00 00 00 00 00 0E 0C E6 1E 17 01 19 11 25 65 0B 00 00 84 55

7A 2C 0B 6E 81 59 02 E0 //Gateway ID
AA //Header
42 //Cont
??00
1D //application
00 22 //Len
04 81 94 22 8E 56 80 00 //UUID Meter ID
00 32 DC B4 //3333300 Positive water accumulation
00 00 00 00/ /Negative cumulative water volume
00 00
00 00 //Meter State
00 0E //Voltage
0C //temperature
??: E6 1E
17 01 19 11 25 //Meter reading time
65 //Meter to Gateway RSSI
0b //Meter to Gateway SNR
00 //Gateway to Meter RSSI
00 //Gateway to Meter SNR
84 //CRC

Communications link



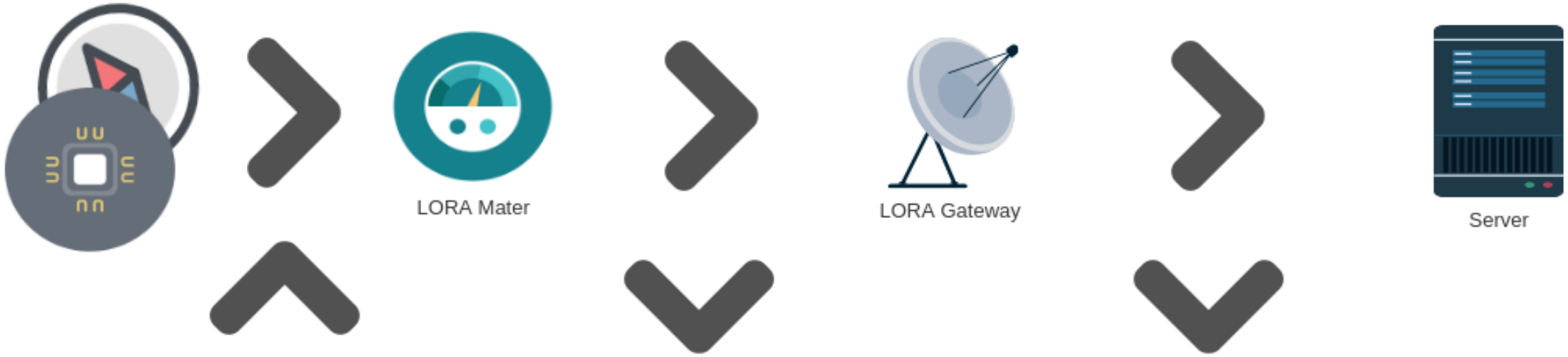


0x91 0xF7 0xC0 0x40 0x80 0x18 0x2C 0x81 0x29 0x01 0x00
0x04 0x81 0x94 0x22 0x8E 0x56 0x80 0x00 //UUID Meter ID
0x08 0x00 0x01 0x06
0x29 0x0A 0x00 0x00
0xB4 0xDC 0x32 0x00 //3333300 Positive water accumulation
0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00
0xE6
0x0C //temperature
0x1E
0x0E //voltage
0x00
0x00 //Gateway to Meter RSSI
0x00 //Gateway to Meter SNR
0xBF 0x38 //END

AA 00 28 7A 2C 0B 6E 81 59 02 E0 AA 42 00 1D 00 22 04 81 94 22 8E 56 80 00 00 32
DC B4 00 00 00 00 00 00 00 00 00 0E 0C E6 1E 17 01 19 11 25 65 0B 00 00 84 55

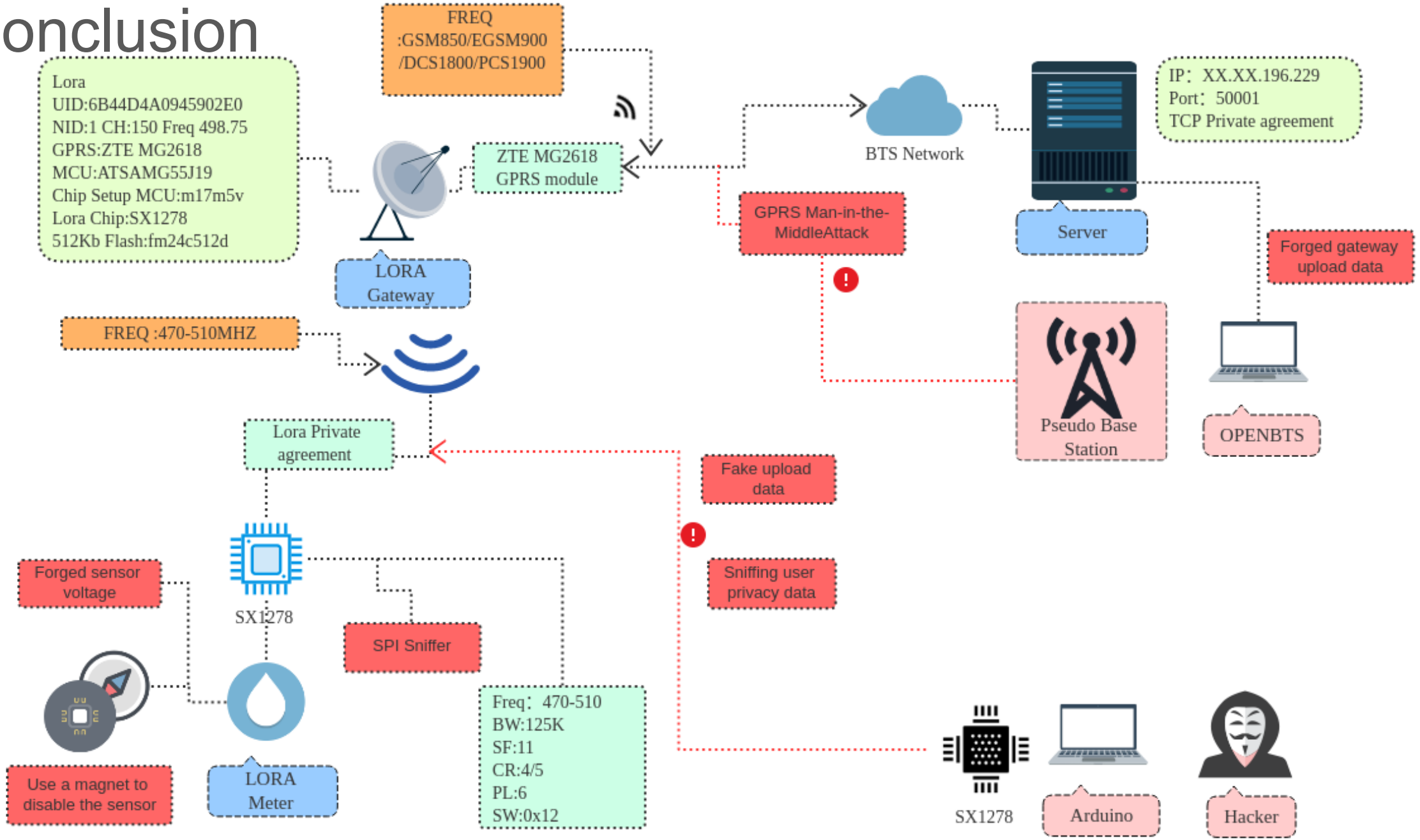
7A 2C 0B 6E 81 59 02 E0 //Gateway ID
AA //Header
42 //Cont
??00
1D //application
00 22 //Len
04 81 94 22 8E 56 80 00 //UUID Meter ID
00 32 DC B4 //3333300 Positive water accumulation
00 00 00 00/ /Negative cumulative water volume
00 00
00 00 //Meter State
00 0E //Voltage
0C //temperature
??: E6 1E
17 01 19 11 25 //Meter reading time
65 //Meter to Gateway RSSI
0b //Meter to Gateway SNR
00 //Gateway to Meter RSSI
00 //Gateway to Meter SNR
84 //CRC

test environment

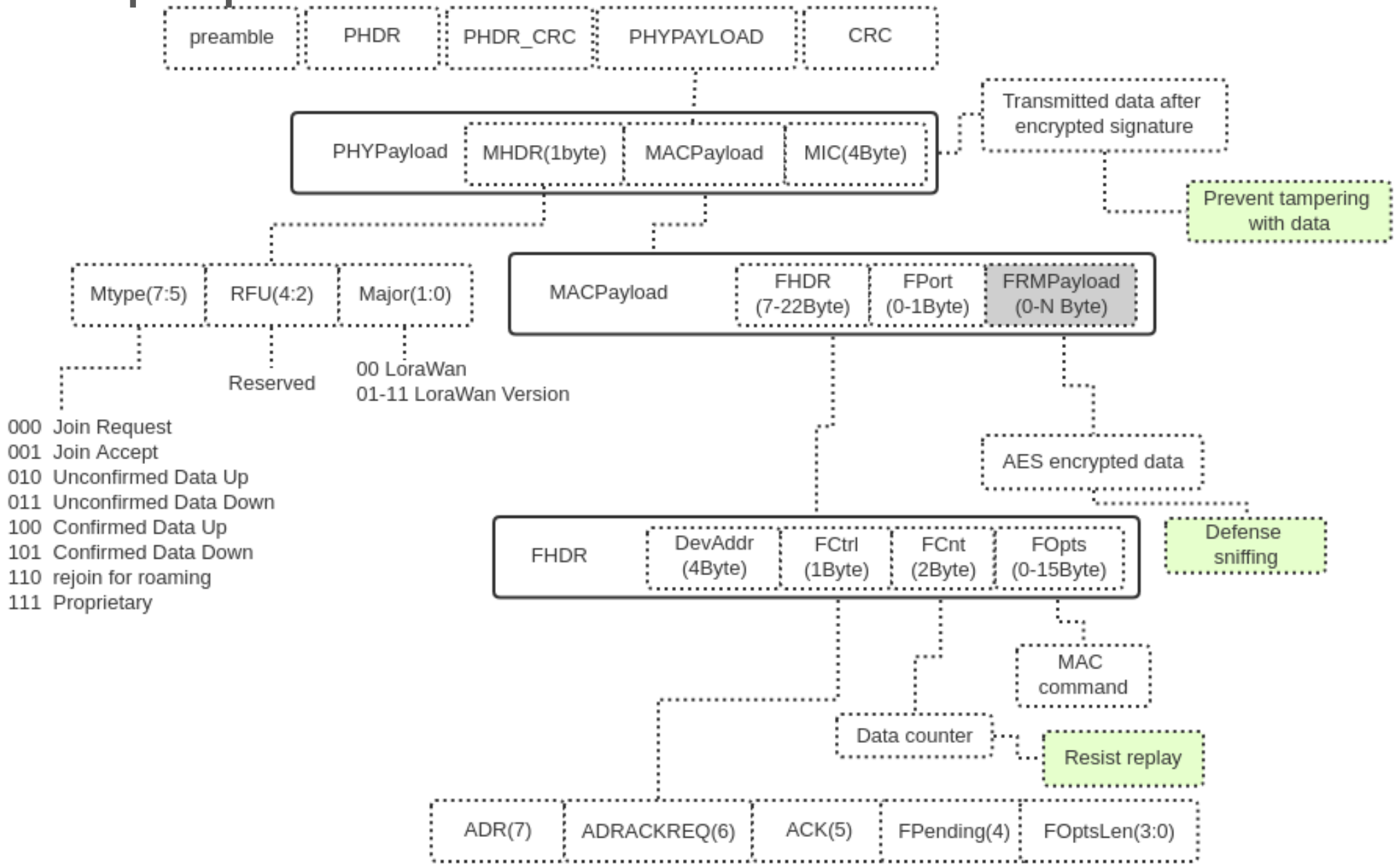




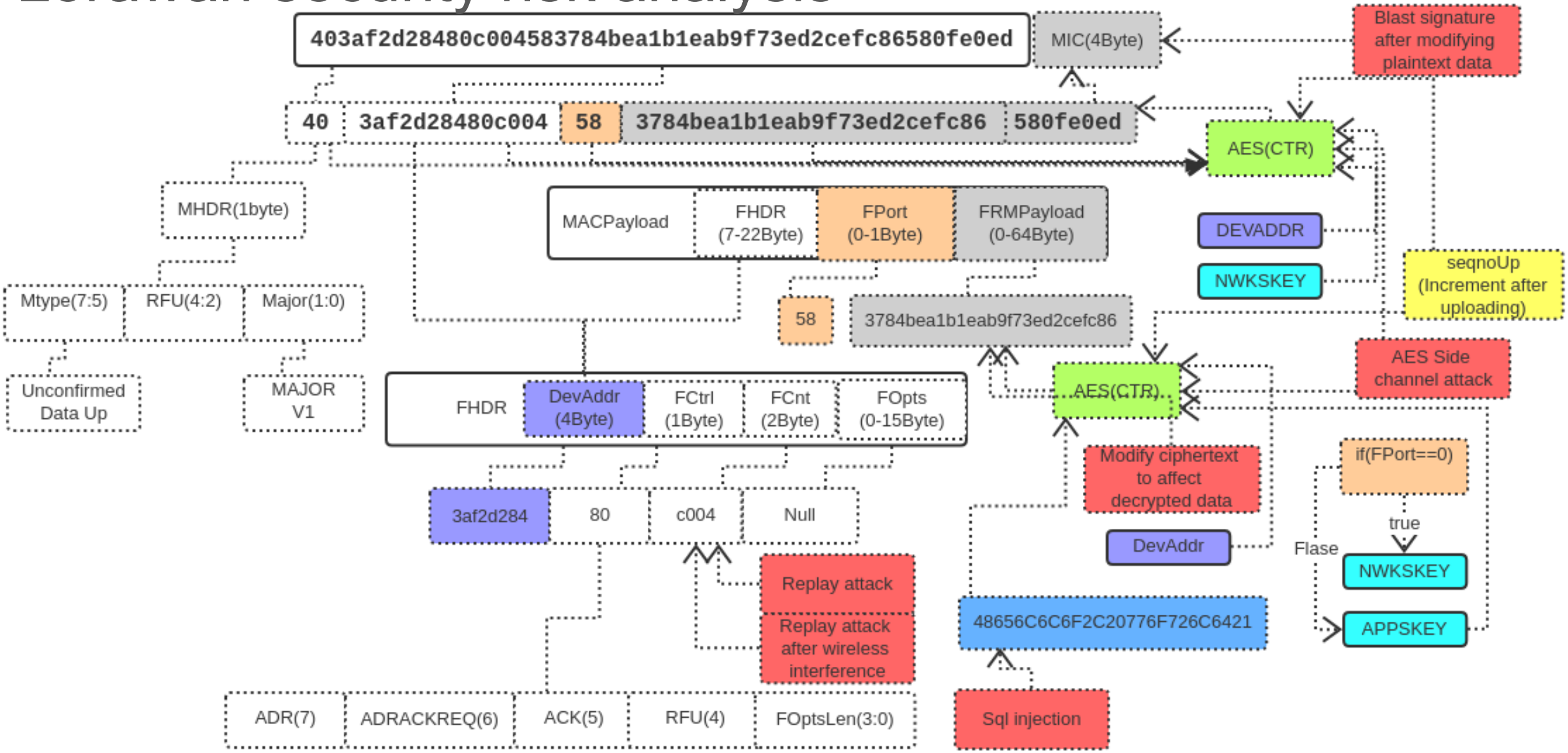
Conclusion


FREQ
:GSM850/EGSM900
/DCS1800/PCS1900
Lora
UID:6B44D4A0945902E0
NID:1 CH:150 Freq 498.75
GPRS:ZTE MG2618
MCU:ATSAMG55J19
Chip Setup MCU:m17m5v
Lora Chip:SX1278
512Kb Flash:fm24c512d
ZTE MG2618
GPRS module
LORA
Gateway
BTS Network
IP：XX.XX.196.229
Port：50001
TCP Private agreement
Server
GPRS Man-in-the-
MiddleAttack
Forged gateway
upload data
FREQ :470-510MHZ
Pseudo Base
Station
OPENBTS
Lora Private
agreement
Fake upload
data
Sniffing user
privacy data
Forged sensor
voltage
SX1278
SPI Sniffer
Use a magnet to
disable the sensor
LORA
Meter
Freq：470-510
BW:125K
SF:11
CR:4/5
PL:6
SW:0x12
SX1278
Arduino
Hacker

Defense proposal



Lorawan security risk analysis



References

https://github.com/BastilleResearch/gr-lora

https://blog.csdn.net/ronhu

https://myriadrf.org/blog/lora-modem-limesdr

https://lora-alliance.org/sites/default/files/2018-04/lorawantm_specification_-v1.1.pdf

https://www.semtech.com/uploads/documents/DS_SX1276-7-8-9_W_APP_V5.pdf

Thank You!