Obtaining The Firmware

ISEN
ALL IS DIGITAL!
yncréa

- Depending on the device, the hacker can have different options:
  - Downloading it directly from the vendor’s site
  - Proxying or sniffing traffic during device update
  - Dumping it from the device

Obtaining The Firmware

- Downloading it from the vendor's site
  - The physical analysis of the device can provide its reference
    - Vendors usually freely provide firmware access on their support sites
    - Forums can give you access to older versions no longer officially available (but still found on connected devices)
  - Network services analysis or UART access can provide you with the firmware version used

Obtaining The Firmware

- Proxying or sniffing traffic during device update
  - Proxying relies on MiTM attacks meaning a more global attack of the target which can be performed through:
    - Access to its network
    - Compromised DNS
    - Compromised Internet gateway
    - Compromised update machine
  - Sniffing traffic can be performed under specific conditions if you have access to the wired or the WiFi network of the victim

Obtaining The Firmware

- Dumping it directly from the device:
  - Accessing the EEPROM through SPI or I2C
    - SPI & I2C are communication protocols with direct access to memory banks
  - Accessing the EEPROM through JTAG
    - JTAG was originally design to test & identify circuit issues
    - JTAG coupled with openCD can allow firmware dumping
  - Specific tools to dump the memory modules might be available through UART connections or connecting through the UART can grant you root access to the device itself

Analyzing The Firmware

- A firmware is usually composed of three parts:
  1. The bootloader: in charge of basic resource allocations and various device components initialization
  2. The kernel: the core component of the IoT device ; can be seen as the intermediary level between the software and the hardware
  3. The filesystem: contains individual “files” needed by the device to work

Analyzing The Firmware

- A standard Linux device bootup process:
    - The bootloader initializes various hardware components
    - The bootloader loads the kernel starting initial processes and services ; the bootloader usually “dies” once the kernel is loaded
    - The root filesystem is mounted
    - The ”init” program is executed
        - The init program then starts the different services and applications (usually through different shell scripts)

Analyzing The Firmware

- Standard actions when analyzing the firmware consist in:
  - Identifying & extracting the filesystem
  - Mounting the filesystem
  - Analyzing the filesystem content
  - Emulating the device for dynamic (close to real life) analysis

Identifying The Filesystem

ISEN ALL IS DIGITAL! | yncréa

- Standard file system types found in IoT devices:
  - Squashfs
  - Cramfs
  - JFFS2
  - YAFFS2
  - Ext2
- Different compression algorithms can also be used to reduce the storage print:
  - LZMA
  - Gzip
  - Zip
  - Zlib
  - ARJ

Extracting The Filesystem

ISEN ALL IS DIGITAL! | yncréa

- Once identified, the filesystem must be extracted to be analyzed:
  - Specific tools can be used to automate the full process
  - Should the tools fail, manual analysis must be performed
- Once extracted (and eventually mounted), the filesystem can be analyzed

Analyzing The Filesystem

- In this part, the hacker is usually going to target different POIs:
  - Hardcoded credentials
  - Configuration files
  - Application source code (typically for web apps)
  - Vulnerable programs & services
  - “Homemade” programs
  - Private keys

Emulating The Device

ISEN
ALL IS DIGITAL!
yncréa

- To ease firmware analysis and find possible weak points, the hacker can also emulate the device
  - Previous analysis should have identified the CPU type
  - Emulation programs can be used to execute programs found in the filesystem "natively"
  - Possible POIs:
    - Understand how the device works (applications, services, authentication,…)
    - Identify programs with possible vulnerabilities - Debuggers can be hooked to programs in the virtual device
    - Test network attacks on the virtual device
    - Locate and decode encryption algorithms or password generation programs

Modifying The Firmware

- Hackers might want to modify an existing firmware to:
  - Change the configuration
  - Add services including backdoors
- Challenges:
  - Compiling new binaries
  - Adding the new “features” to the firmware image
  - Pushing the new firmware on the device

Modifying The Firmware

ISEN ALL IS DIGITAL! yncréa

- Compiling new binaries:
  - Similar to what you did in the MCU class
  - Install a cross-compilation environment for the target CPU
  - Compile the malware source for the target architecture
  - Test it in the emulated environment

Modifying The Firmware

- Adding the new “features” to the firmware image
  - It depends on the type of devices (MCUs, embedded standard OS, proprietary solutions)
  - From the firmware analysis, you should have a good understanding of the bootup process
  - Create the necessary scripts to activate your malware at startup
  - To create a new firmware image, you must do in reverse what you did in the extraction part
    - Tools such as the “firmware-mod-kit” exist to ease the process on Linux based systems

Pushing A New Firmware On The Device

- The process depends on the device type and on the global IoT environment
  - Do you have physical access to the device?
  - Do you have access to the device through the network?
  - Did you retrieve admin rights on the device?
  - Can you perform MiTM attack to intercept normal update requests?
- Network updates can also be conditioned:
  - Are update files signed?
  - Is the update process encrypted?

ISEN | école d'ingénieurs
ALL IS DIGITAL!

The communication layer

The Communication Layer Attack Vector

- IoT devices need to communicate
  - Standard "low-level" communication protocols can be used:
    - WiFi, Bluetooth, Bluetooth LE, NFC, LoRaWan,…
    - Proprietary defined protocols: remote controls, smart home devices
  - Standard "high-level" communication protocols can be used:
    - TCP/IP, Zigbee
  - IoT devices can use wired or wireless communication solutions

Standard Attack Vectors

ISEN ALL IS DIGITAL! | yncréa

- Depending on the device and on what can be accessed by the hacker, two main attack vectors can be identified:
  - The network layers based on the OSI or on the TCP/IP model
  - The physical layers:
    - Wired medium (not covered in this class)
    - Wireless medium

The Network Attack Vector

- The global approach is similar to attacks performed on standard (non IoT) devices
- These techniques are not covered in this class
- Usually based on different techniques:
  - Scanning
  - Identification
  - Exploitation
  - Escalation
  - Spoofing
  - MiTM
  - Proxying

The Network Attack Vector

- Standard protocols used by IoT devices to interact with the “outside” world are:
  - HTTP
  - MQTT
  - CoAP

The Radio Vector

- IoT wireless communication protocols are based on radio telecommunications
- Different protocols can be used:
  - Standard IoT protocols (WiFi, BLE, NFC, LoRa)
  - Vendor defined radio protocols
- Frequencies used in IoT depends on the protocol and can range between a few hundreds of kHz to some GHz
- Frequency used can also be country dependent
  - E.g.: Zigbee operates on 2.4 GHz in most countries, 902 to 928 MHz in America and Australia, and 868 to 868.6 MHz in Europe

The Radio Vector

- For an IoT hacker, different layers can be studied:
  - The carrier waves
  - The modulation schemes
    - Analog modulation schemes
    - Digital modulation schemes
  - The specific encoding
  - The protocols used between the senders and the receivers

The Carrier Waves

- Device analysis can provide you with the frequencies used to transmit information
  - Via FCC information
  - Via tags directly found on the device or in the user manual
- Frequencies can also be “retrieved” using SDR devices and software tools

FCC information example

ISEN
ALL IS DIGITAL!
yncréa

- FCC retrieved information: Garage door example
- FCC ID: M48TOP-NA
- Information:

1.2 *E.U.T. classification*

R&TTE category : Transmitter 433,92 MHz in AM/ASK

TEST CONDITIONS		***Occupied frequency range (at 20 dB point)***		
		f_L [MHz]	**F_C [MHz]**	**f_H [MHz]**
T_{amb} : + 24 °C	V_{nom} : 3.0 Vdc	433,9050	433.9255	433.9415
Bandwidth =		***40 kHz***		

The Carrier Waves

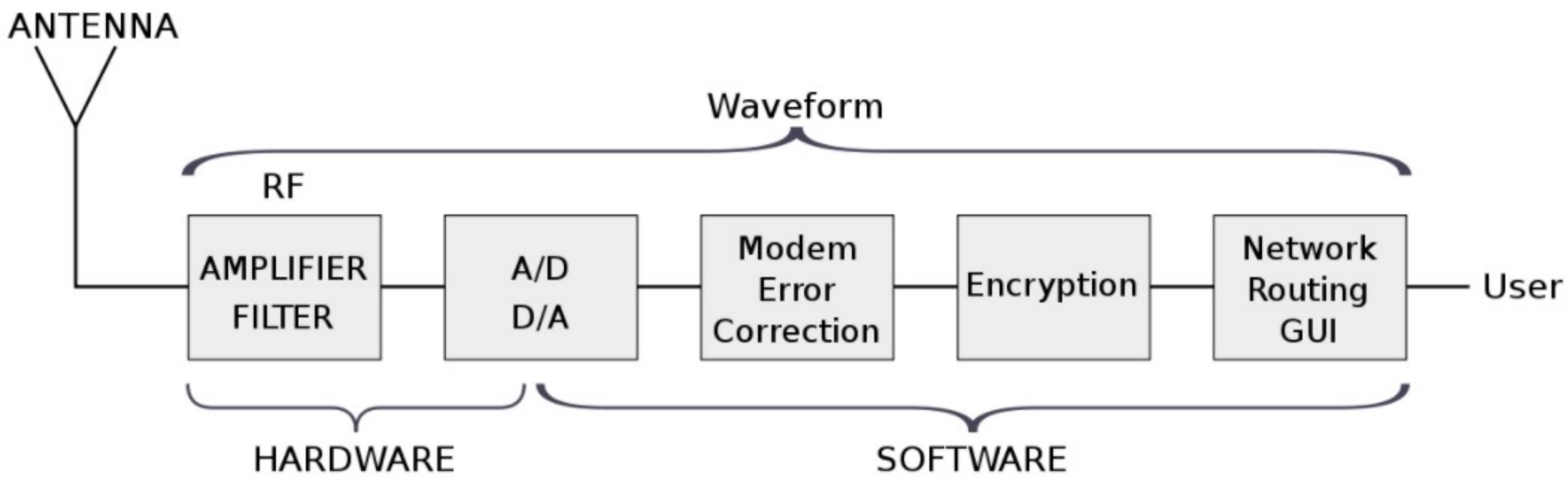
- Tags found on the device or in the user manual

The Carrier Waves

- Using SDR
    - What is SDR?
    - How can SDR help us?
    - Examples

What Is SDR?

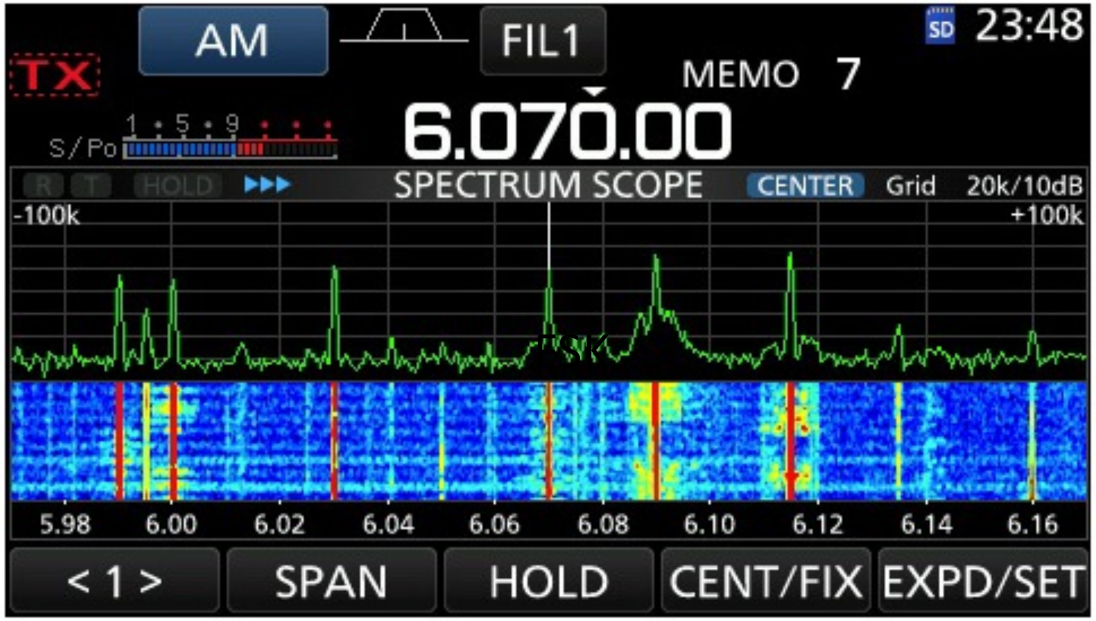
- Software-defined Radio



How Can SDR Help Us?

- SDR devices can:
  - Sweep the standard bandwidths for signals of interest
  - Be used to identify the frequencies involved
    - FFT analysis to identify major frequencies
    - Constellation diagrams to detect patterns in frequencies & phases

SDR Example



Frequency spectrum sweep

Source: https://www.universal-radio.com/

Modulation Schemes

- Most IoT devices use digital modulation schemes
- Modulation information:
  - Can be retrieved through official documents

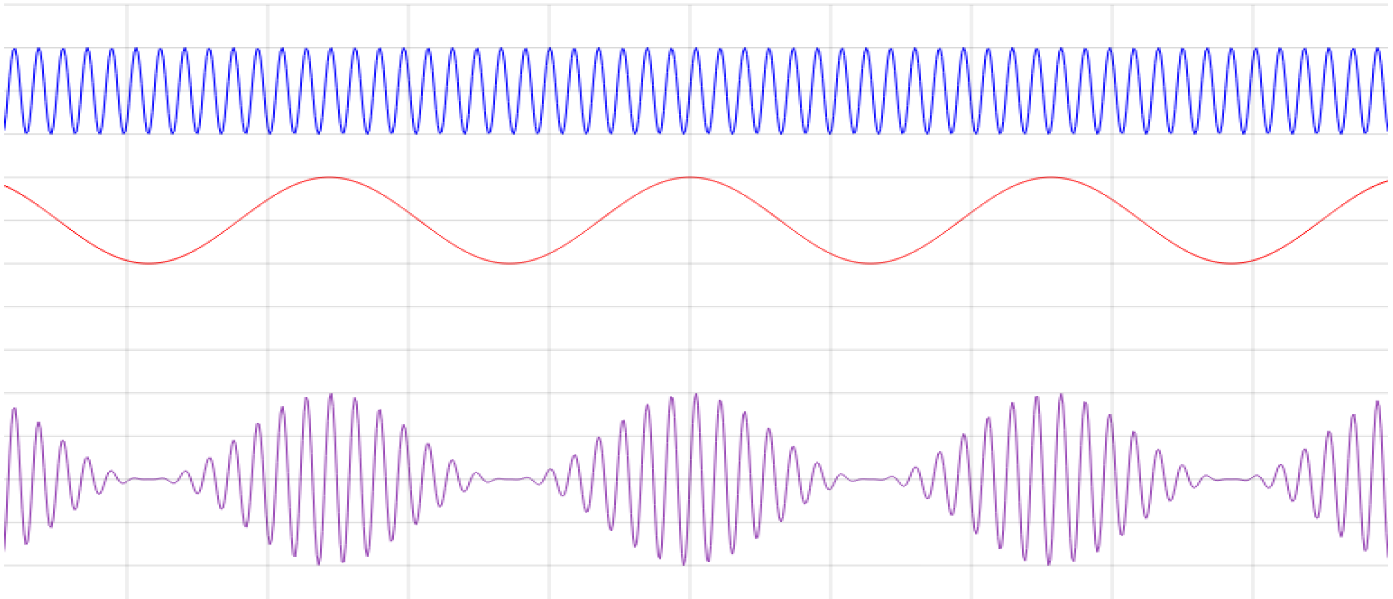
    1.2 E.U.T. classification

    R&TTE category : Transmitter 433,92 MHz in AM/ASK

  - Can be “ guessed” and retrieved through signal analysis

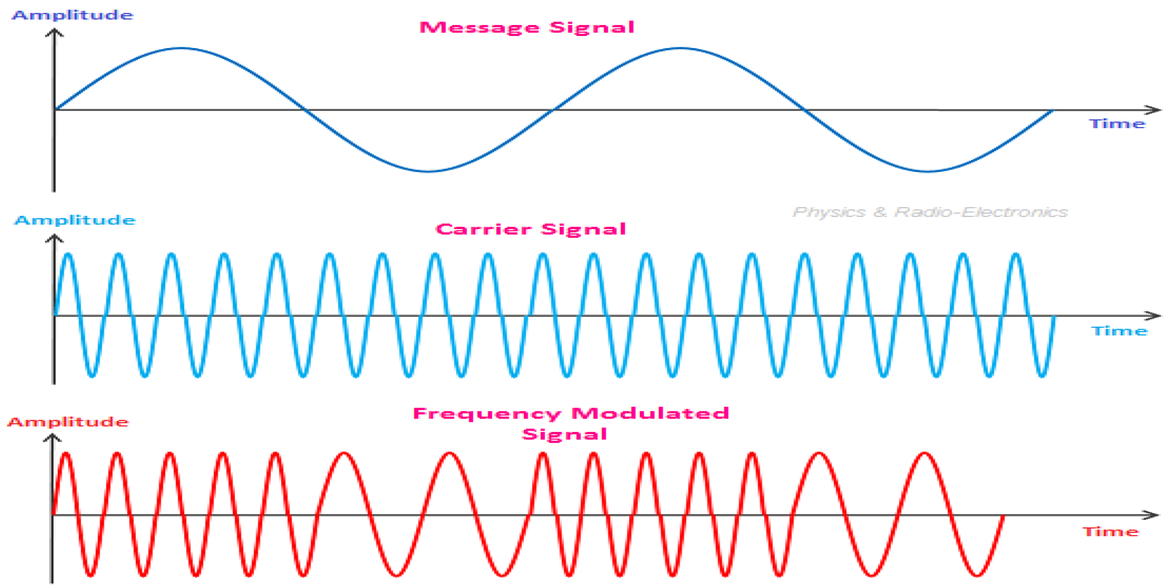
Analog Modulation Schemes

- Standard Analog modulation schemes:
  - Amplitude modulation



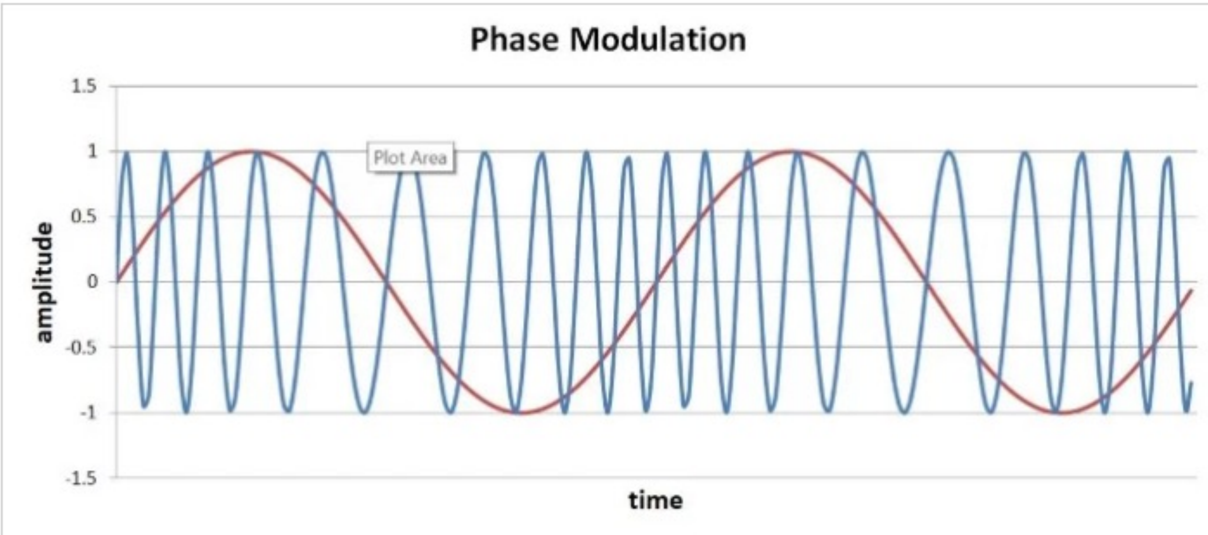
Analog Modulation Schemes

- Standard Analog modulation schemes:
  - Frequency modulation



Analog Modulation Schemes

- Standard Analog modulation schemes:
  - Phase modulation

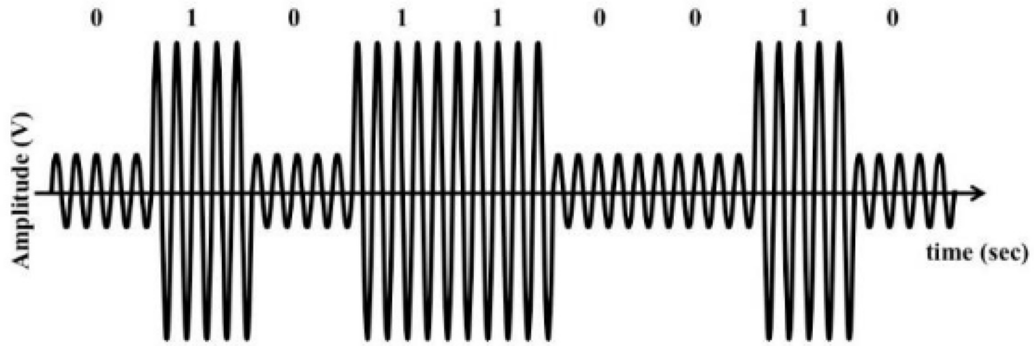


Digital Modulation Schemes

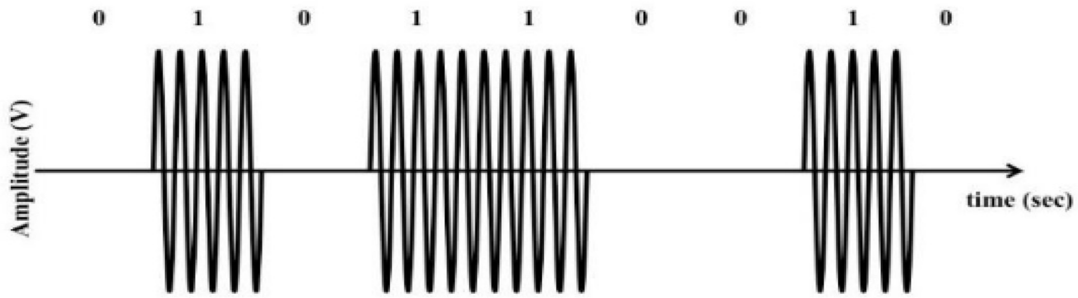
- Most standard modulation types
  - ASK
  - FSK
  - PSK
- Other schemes
  - QAM: Quadrature amplitude modulation (PSK&ASK)
  - Multiple bit transmission modulation schemes
    - QPSK: Quadrature Phase Shift Keying
    - 8-PSK
    - 16-PSK

Digital Modulation Schemes

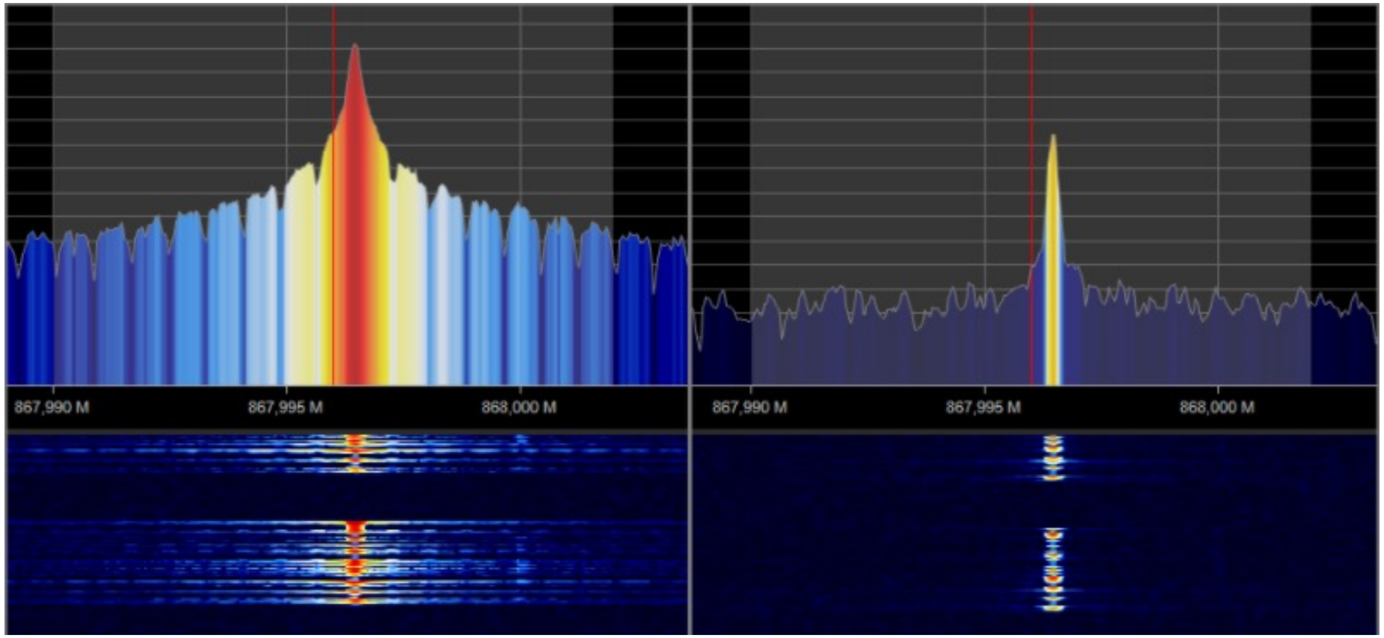
ASK



OOK (On-Off Keying)



SDR Analysis

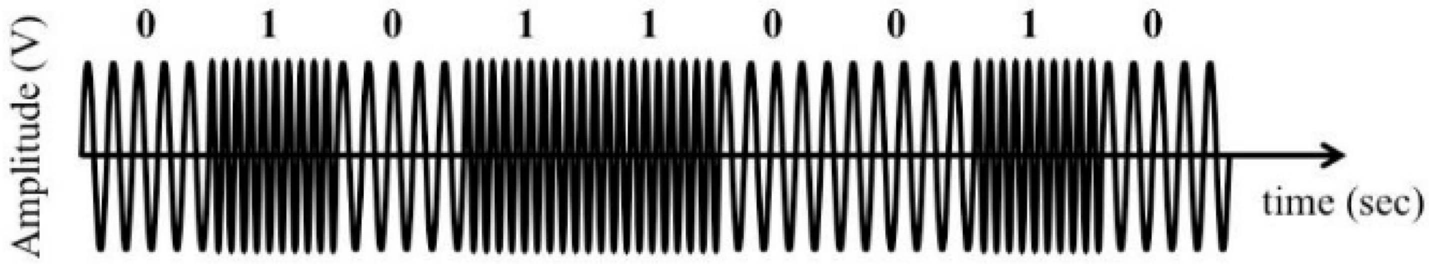


ASK OOK

Source: https://hackaday.com/2020/01/28/rf-modulation-crash-course-for-hackers/

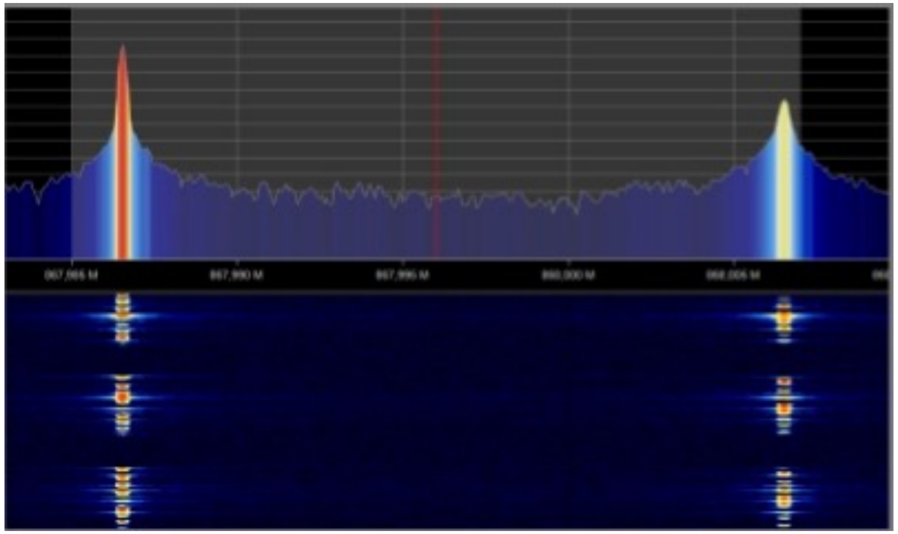
Digital Modulation Schemes

FSK (Frequency Shift Keying)



SDR Analysis

ISEN
ALL IS DIGITAL!
yncréa

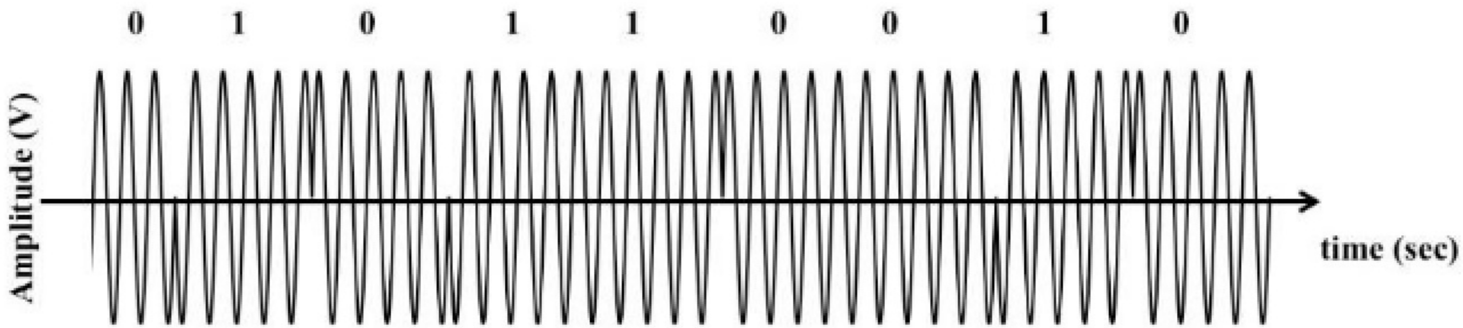


FSK

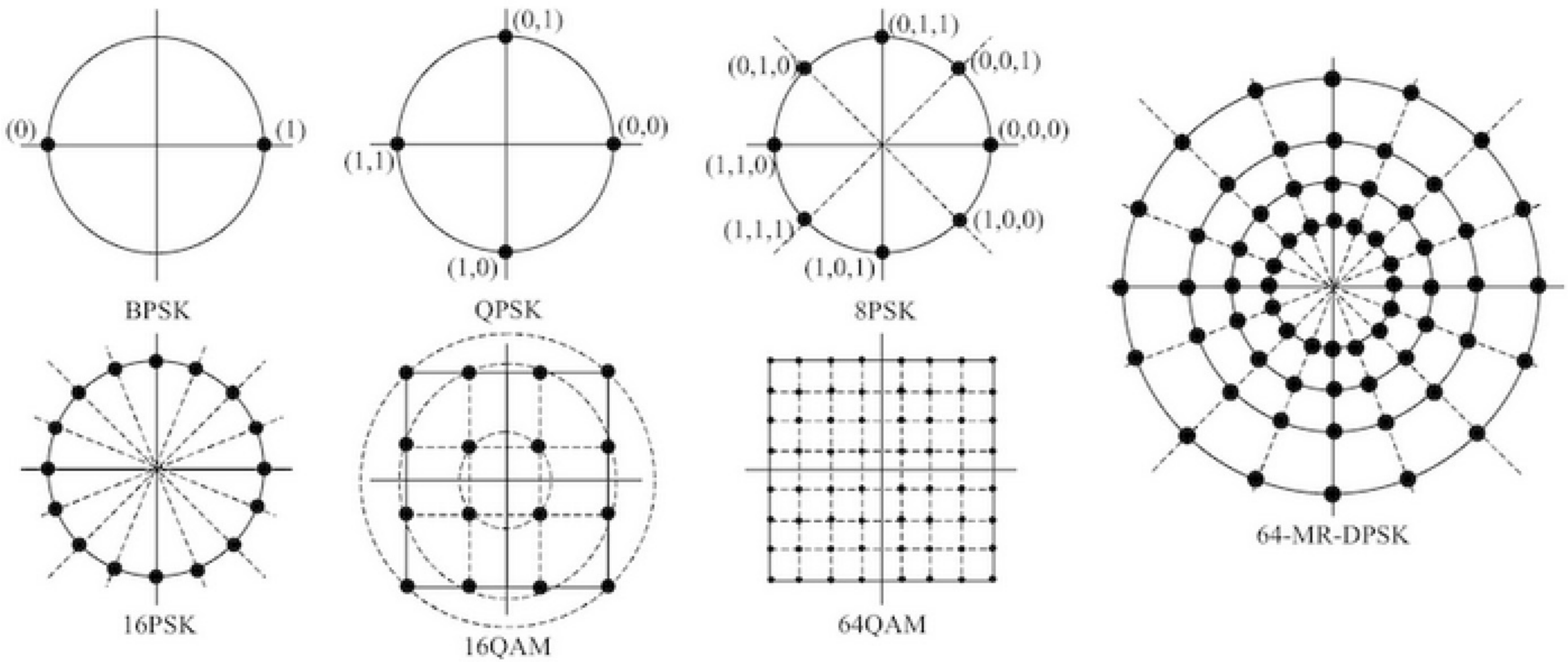
Source: https://hackaday.com/2020/01/28/rf-modulation-crash-course-for-hackers/

Digital Modulation Schemes

PSK – BPSK (Binary Phase Shift Keying)

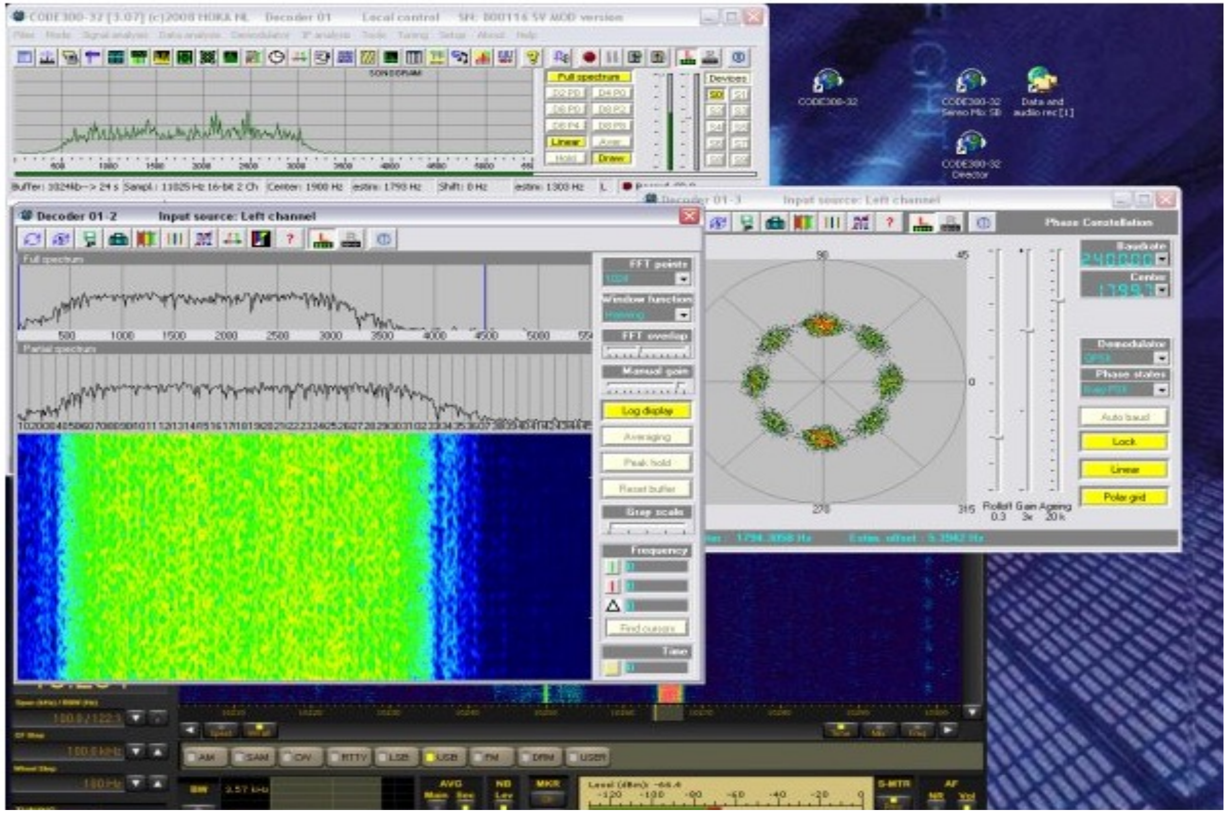


Other modulation schemes



SDR Analysis

ISEN
ALL IS DIGITAL!
yncréa



PSK

Source: http://www.hoka.it/

Specific Encoding

- Specific encoding can be added to the original signal to reduce transmission errors and synchronize the sender and the receiver:
  - Starting / ending / synchronization sequences
  - Standard line coding schemes (e.g., Manchester encoding)
  - Checksums
- If these schemes are not given, they will have to be guessed and derived from the raw signal

Protocols

ISEN
ALL IS DIGITAL!
yncréa

- Just like with any transmission, a protocol must be defined between the different devices
- Protocols can be based on standard protocols or specifically designed by the vendors
- Should they not be described, they will have to be guessed and derived
- Encryption can be used to enforce different security levels