APSYS .Lab

Spark the future. Craft tomorrow.

LAAS CNRS

Cross-protocols attacks: weaponizing a smartphone by diverting its Bluetooth controller

Toulouse Hacking Convention - 2022

Romain CAYRE
rcayre@laas.fr

AN **AIRBUS** COMPANY

WHOAMI

Romain CAYRE

- PHD student at **LAAS-CNRS** and **Apsys.Lab**
- My research thematic is focused on **IoT security** and **wireless security**, both from an **offensive** and **defensive perspective**
- Former student of **INSA Toulouse** and **TLS-SEC**
- **Supervisors:**
  - **V. Nicomette**, **G. Auriol**, **M. Kaâniche** (LAAS-CNRS)
  - **G. Marconato** (Apsys.Lab / Airbus)

AGENDA: CROSS-PROTOCOLS PIVOTING ATTACKS

- **Background** and **research question**
- **Bluetooth Low Energy** overview
- **Reverse engineering** and **patching** the **Samsung Galaxy S20 Bluetooth controller**
- Implementing **non-native protocols** support

BACKGROUND AND RESEARCH QUESTIONS

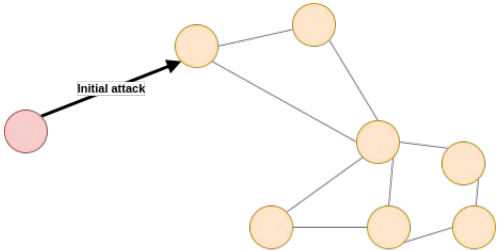
IOT ENVIRONMENTS SECURITY

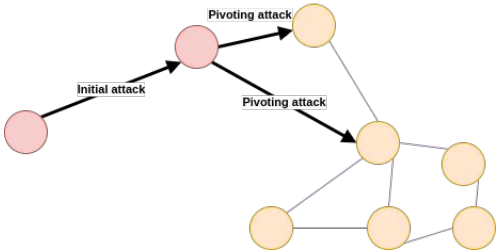
Rapid expansion of **connected objects** in our **daily life: game changer from a security perspective**

- **Massive use** of **heterogeneous** wireless communication protocols, sharing a lot of **similarities** (modulation schemes, frequency bands…) and **co-existing in the same environments**
  → **heterogeneous environments**
- **Peer-to-peer communications**, without central point nor gateways
  → **decentralized environments**
- Presence of **mobile devices** (e.g. smartwatches, smartphones…) with **wireless connectivity**
  → **dynamical environments**

MAIN RESEARCH QUESTION

Can we make a device designed to use **protocol A** communicate with a **different protocol B**?





Offensive scenarios:

- 🎯 Cross-protocol pivoting attacks
- 🎯 Covert channel attacks

CROSS-PROTOCOL PIVOTING ATTACKS: HOW REAL IS THE THREAT ?

Previous work: WazaBee: cross-protocol pivoting attack aiming at diverting a Bluetooth Low Energy transceiver to communicate with Zigbee nodes by exploiting similarities in the modulation schemes

→ mainly evaluated on development boards from Nordic SemiConductors and TI

Samsung Galaxy S20

Extension: Can we perform this kind of attacks from off-the-shelf devices ?

- **Common and mobile Bluetooth controller analysis**: BCM4375 chip from Broadcom, mainly used by Samsung Galaxy S10/S20 smartphones

  → *increasing the attack surface*

- **New protocols support**: Mosart & Enhanced ShockBurst proprietary protocols, commonly used by wireless keyboards and mices

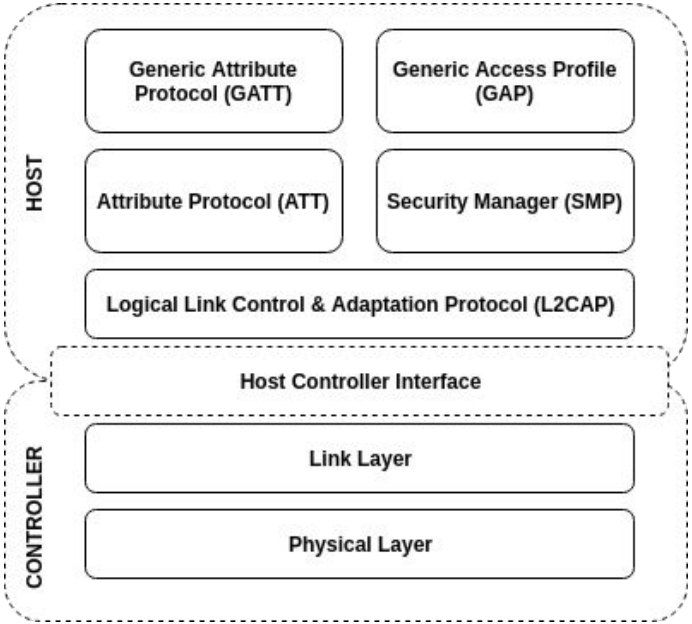
  → *critical devices from a security perspective*

BLUETOOTH LOW ENERGY OVERVIEW

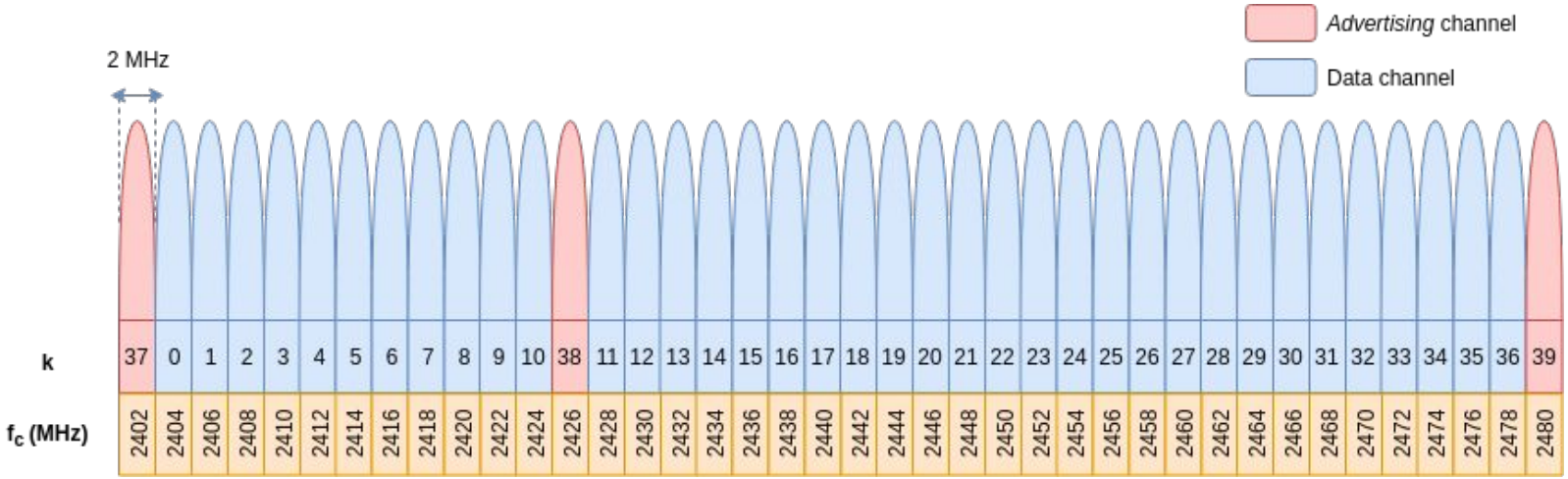
BLUETOOTH LOW ENERGY

- Bluetooth lightweight variant, introduced in Bluetooth Core Specification 4.0
- Designed for low energy consumption
- Low complexity
- Massively deployed (smartphones, laptops, smart devices, …)

BLUETOOTH LOW ENERGY TYPICAL STACK

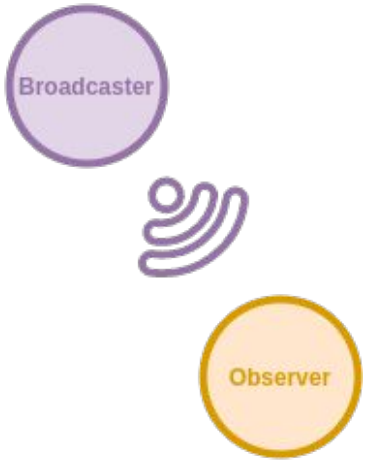


ADVERTISING MODE VS CONNECTED MODE

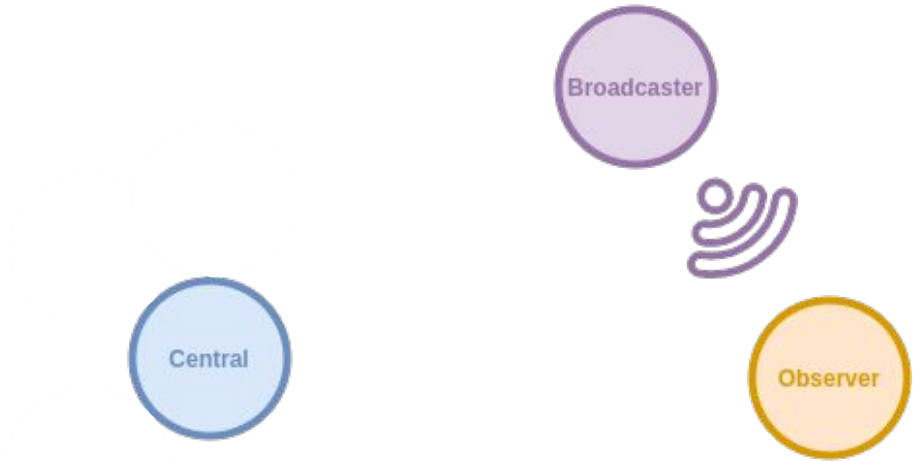


GENERIC ACCESS PROFILE - ROLES

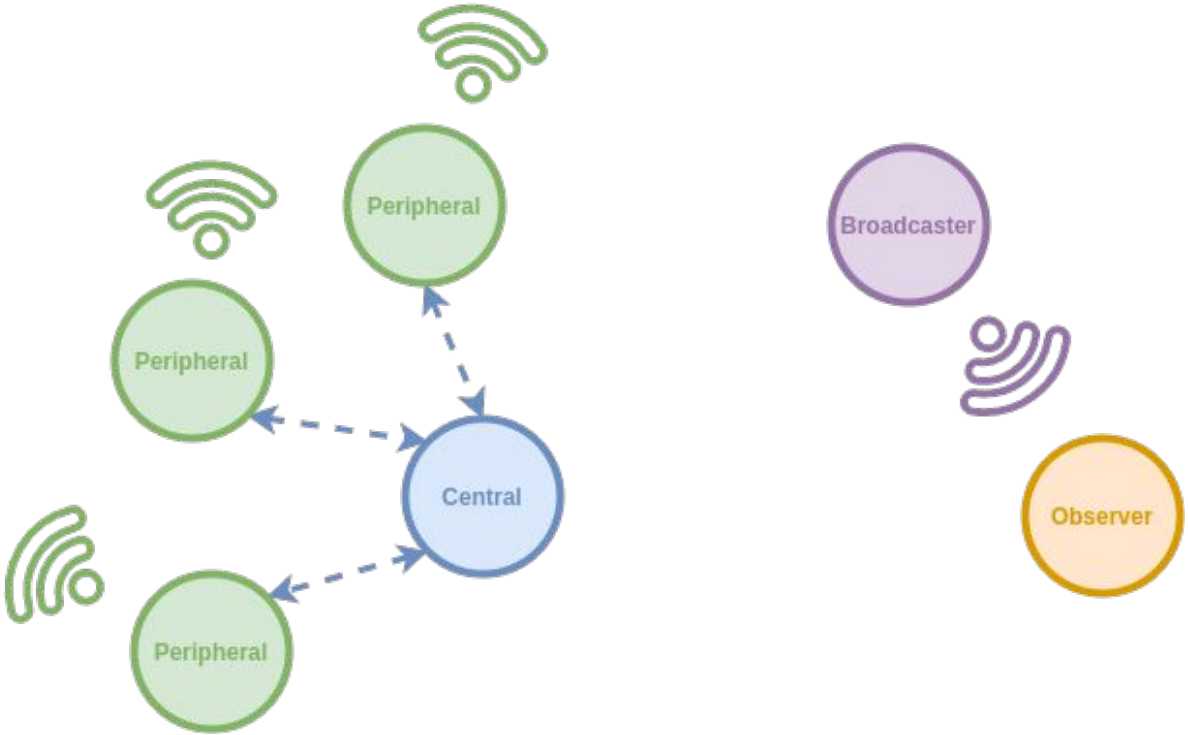
GENERIC ACCESS PROFILE - ROLES



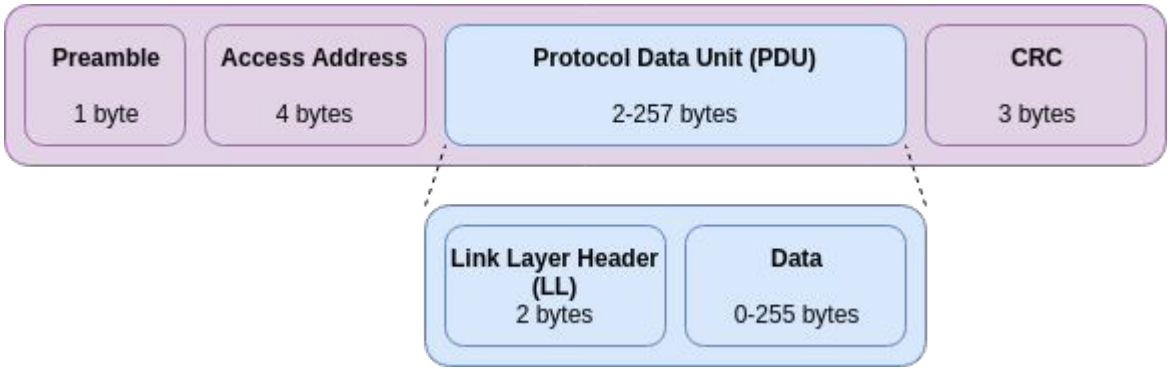
GENERIC ACCESS PROFILE - ROLES



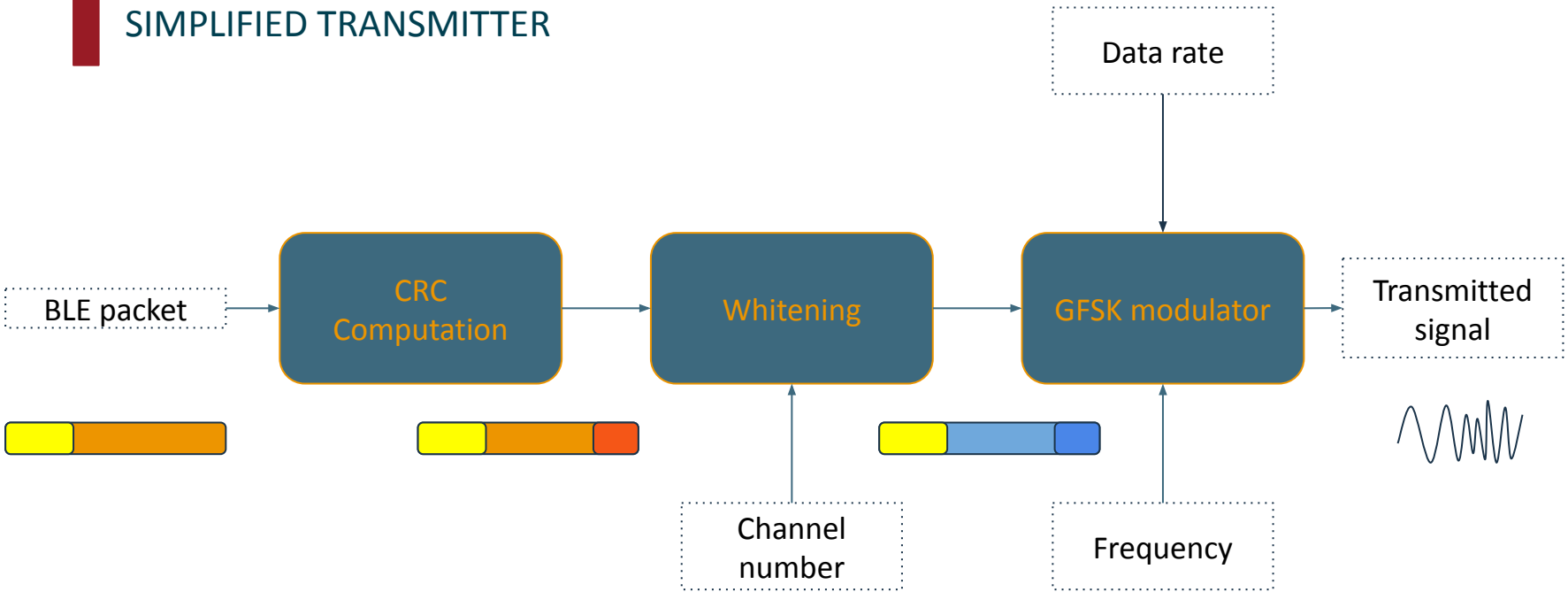
GENERIC ACCESS PROFILE - ROLES



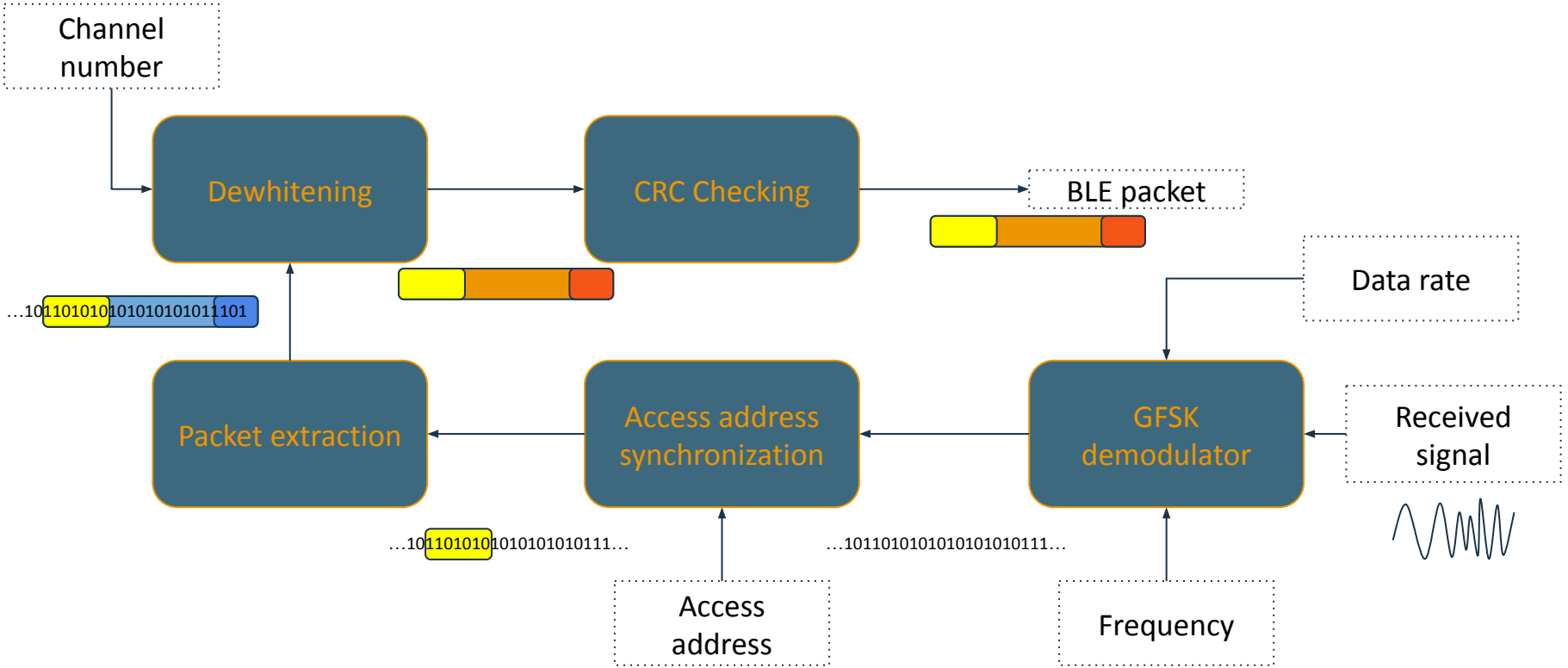
PACKET FORMAT



SIMPLIFIED TRANSMITTER



SIMPLIFIED RECEIVER



REVERSE ENGINEERING AND PATCHING

FIRMWARE REVERSE ENGINEERING

- **Use of InternalBlue framework (SeemooLab)**
  firmware dumping, dynamic analysis, patching

- **BCM4375 (Samsung Galaxy S20) and CYW20735 (IoT development kit) firmwares analysis**
  - Both firmwares share a consequent amount of code
  - CYW20735 symbols are known

- **Static analysis (IDA Pro) and dynamic analysis (InternalBlue)**

- **We need to understand :**
  - how to configure the RF hardware (frequency, preamble, whitening, data rate,...)
  - how to use reception and transmission callbacks (controlling demodulator output and modulator input)
  - how to interact with the Host (HCI commands and events)

InternalBlue

InternalBlue framework

CYW920735Q60EVB-01 IoT development kit

DIVERTING SCANNING AND ADVERTISING TASKS

- **Bluetooth Low Energy roles are implemented as *Tasks*:** Scanner, Advertiser, Central, Peripheral

The tasks linked to advertising and scanning are good candidates to implement our custom reception / transmission primitives:

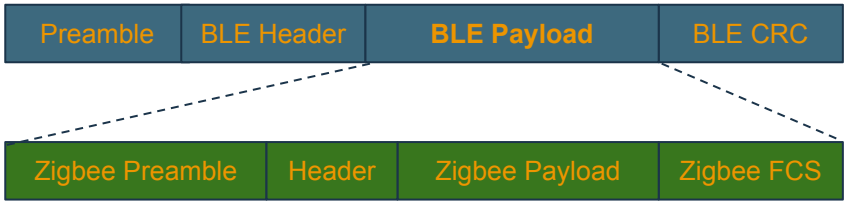
they do not require the establishment of a Bluetooth Low Energy connection as a prerequisite to send and transmit packets

SCANNING TASK

- **scanTaskRxDone function:** provides a direct access to the demodulator output buffer
- **The function has been hooked to execute our own code:** it allows to decode the received packet and send it to the Host

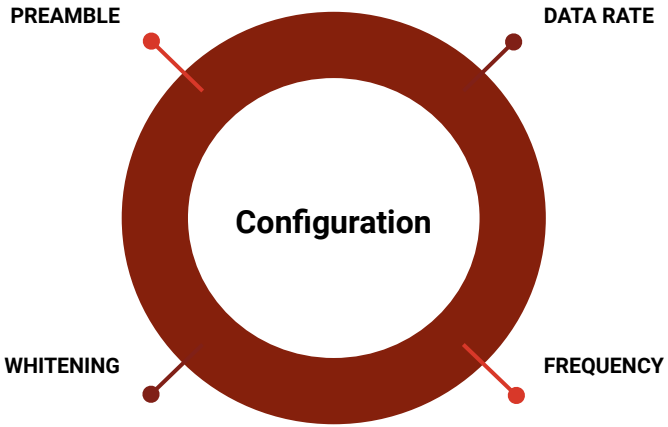
ADVERTISING TASK

- **advTaskProgHw function:** indirect access to the modulator input, we can only configure the advertising packet payload
  - **"Packet in Packet" variant:** the full packet is encapsulated into the advertisement payload

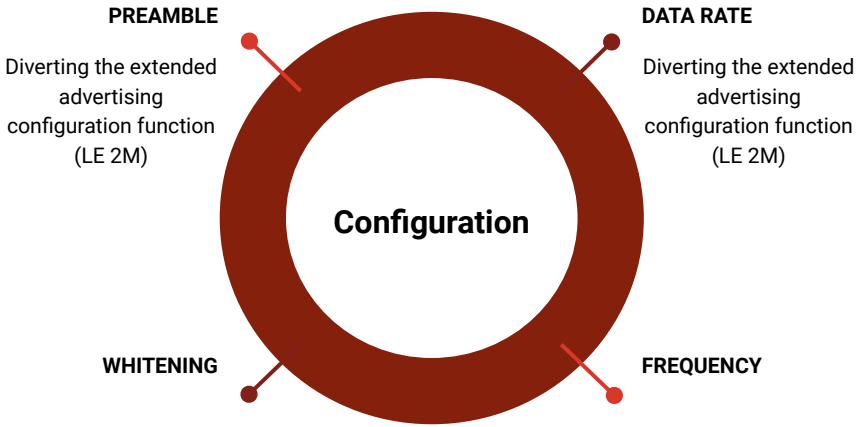


- **The function has been hooked to execute our own code:** it allows to format the packet to transmit

RF HARDWARE CONFIGURATION



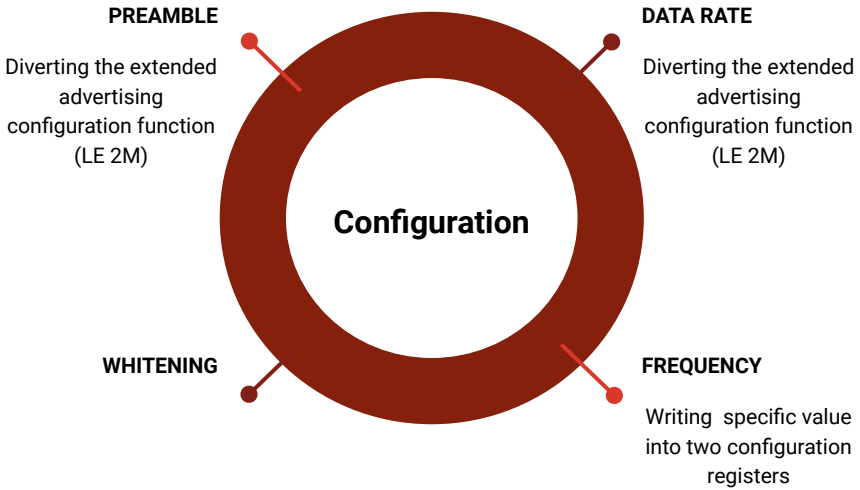
RF HARDWARE CONFIGURATION



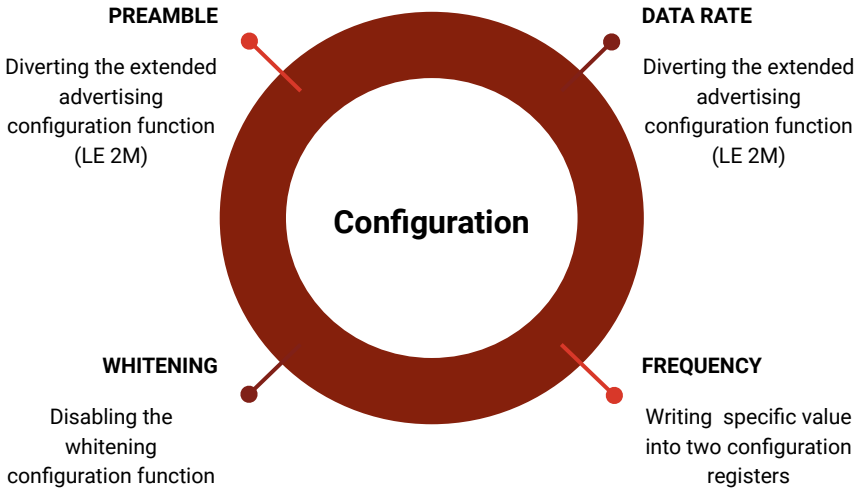
Extended advertising

- **New feature** introduced in Bluetooth Core Specification 5.0
- Allows to use **Data Channels** as **secondary Advertising Channels**
- Uses a **random access address** and not the predefined advertising Access Address
- Can use **1** or **2 Mbit/s datarate**

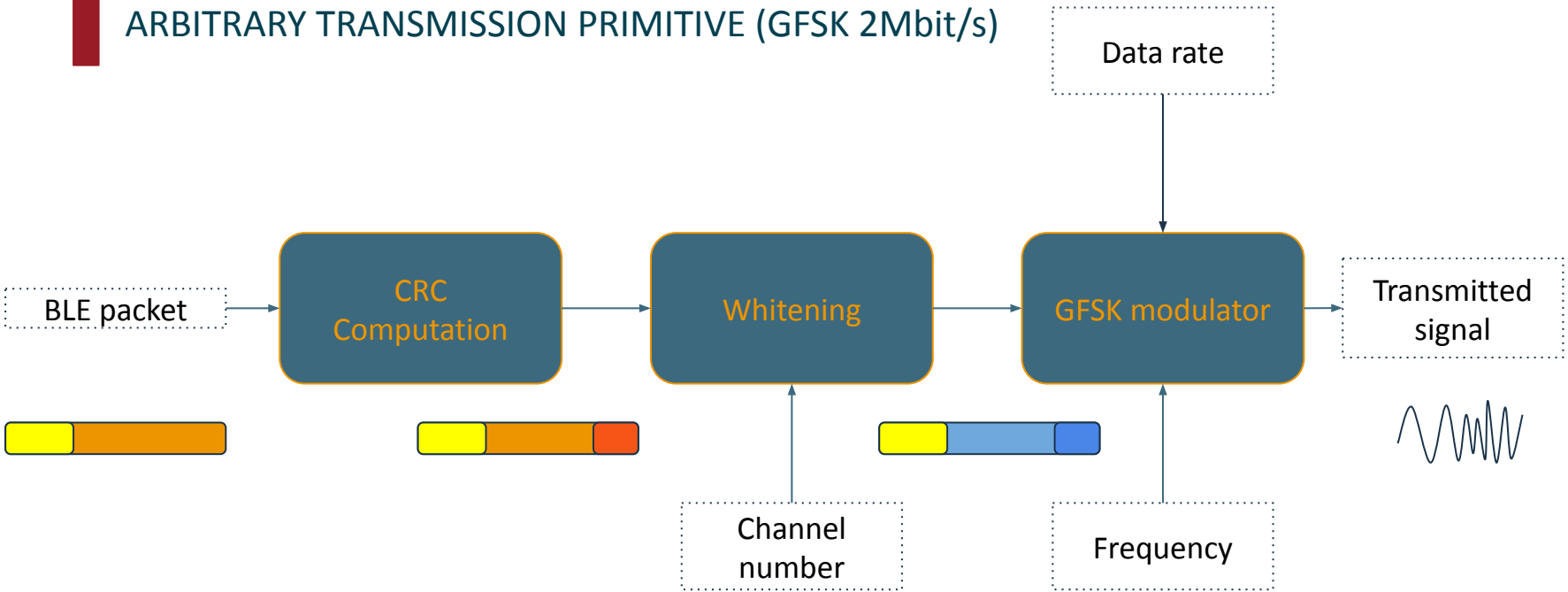
RF HARDWARE CONFIGURATION



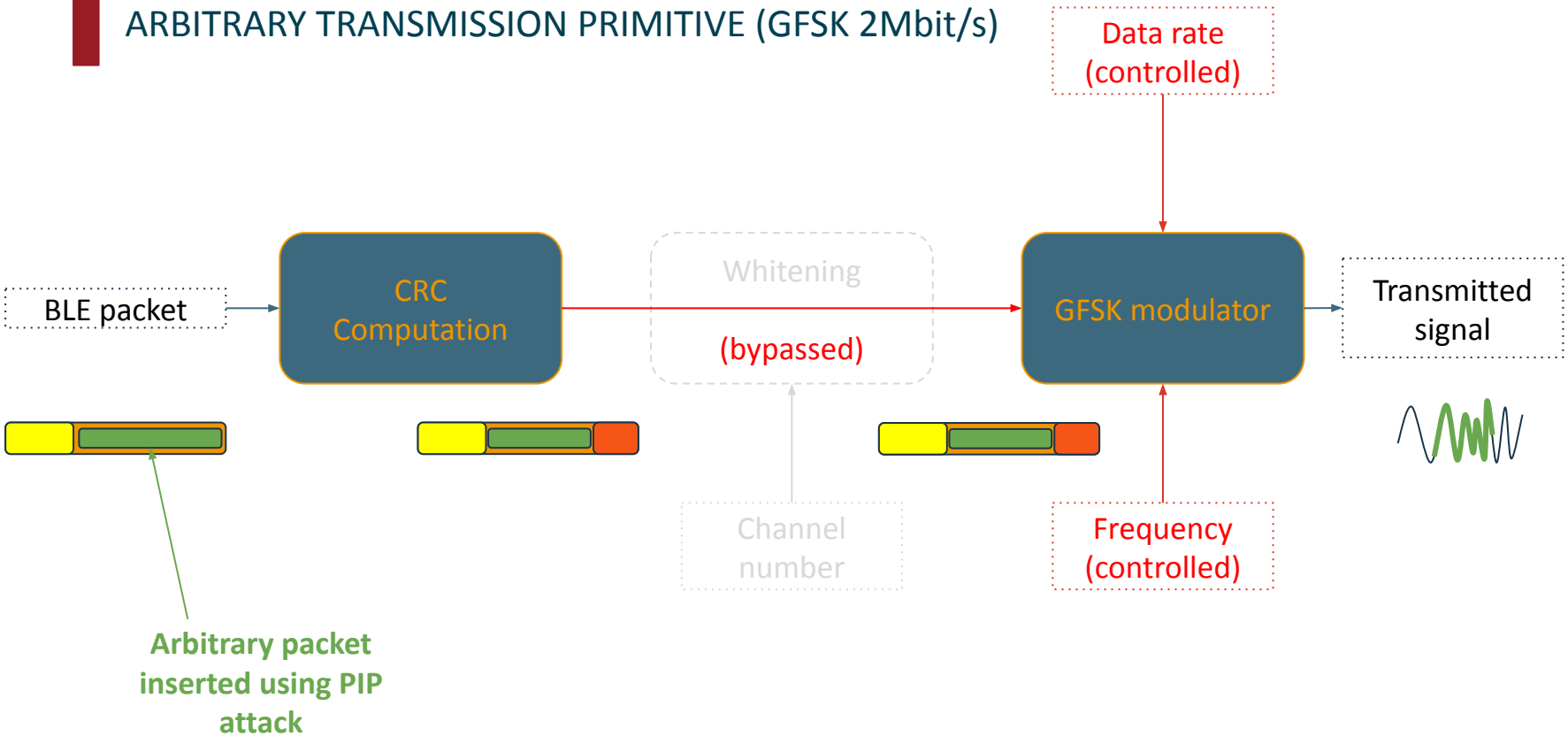
RF HARDWARE CONFIGURATION



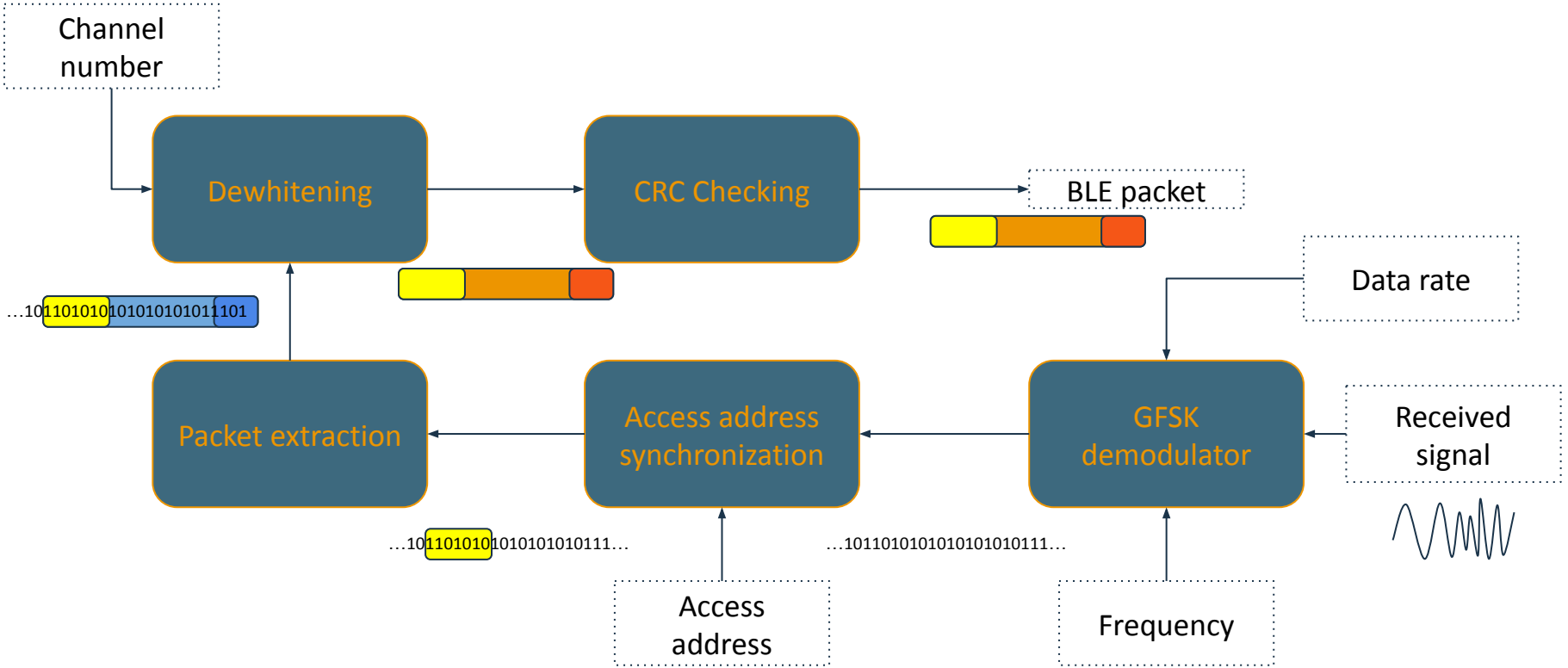
ARBITRARY TRANSMISSION PRIMITIVE (GFSK 2Mbit/s)



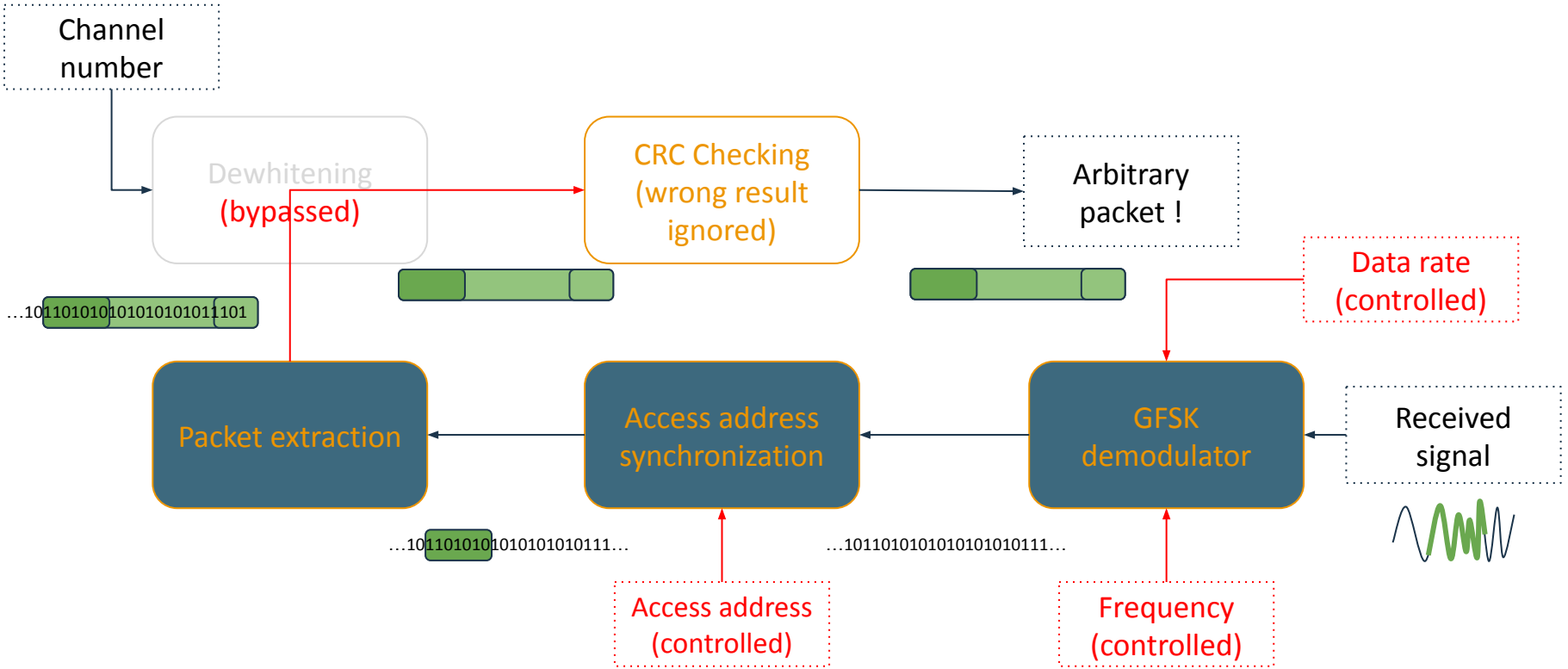
ARBITRARY TRANSMISSION PRIMITIVE (GFSK 2Mbit/s)



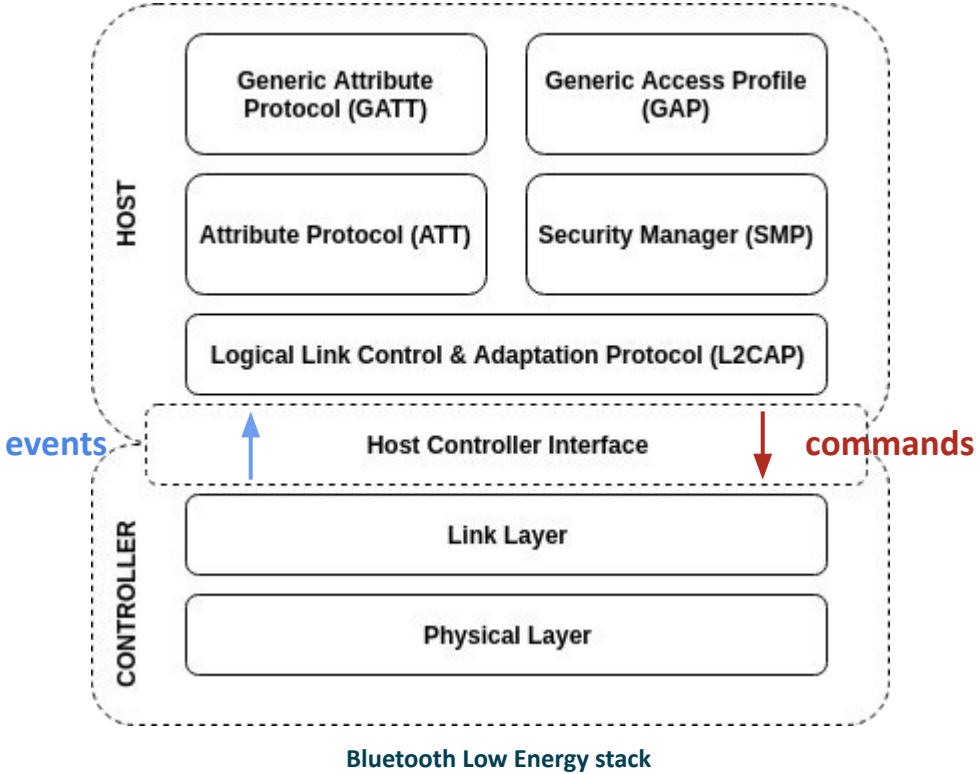
ARBITRARY RECEPTION PRIMITIVE (GFSK 2Mbit/s)



ARBITRARY RECEPTION PRIMITIVE (GFSK 2Mbit/s)



HOST-CONTROLLER COMMUNICATION



Bluetooth Low Energy stack

- **Host to Controller communication (Commands)**
  - HCI commands are handled using an **array of function pointers**
  - the command identifier is used to **calculate an index**, allowing to select the corresponding function
  - We found two **unused** command identifiers and added our **own function's addresses** at the right place

- **Controller to Host communication (Events)**

  We identified two functions allowing to:
  - **allocate a buffer** describing an event message
  - **send** it to the Host

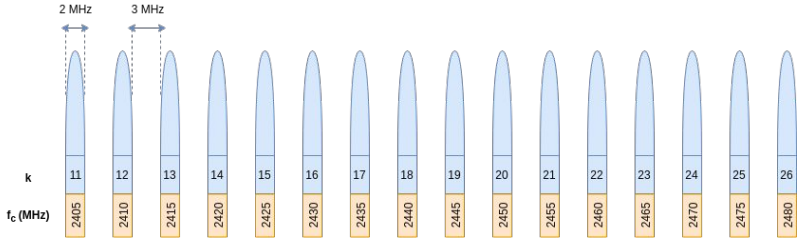
IMPLEMENTING NON-NATIVE PROTOCOLS SUPPORT

ZIGBEE PROTOCOL

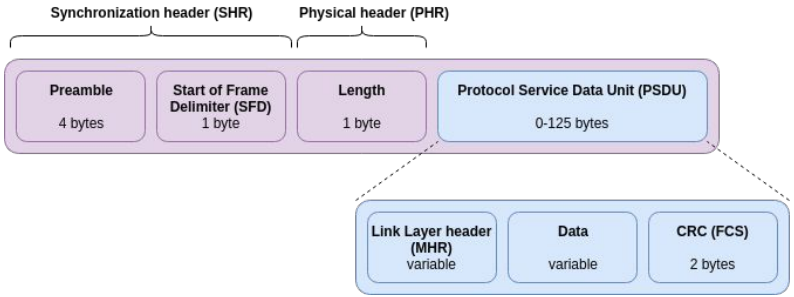
- **Implementing WazaBee attack:**
  - WazaBee establishes an equivalence between 31 bits modulated using a GFSK at 2Mbit/s (BLE) and 32 bits modulated using O-QPSK (Zigbee)
  - **We added two functions allowing to perform the conversion**

- **Selecting the channel:**
  - 16 Zigbee channels, numbered from 11 to 26
  - Central frequency calculation

- **Synchronising the receiver:**
  - A Zigbee preamble is composed of four null bytes
  - We use the GFSK sequence corresponding to “0” symbol as preamble



Zigbee channels (2.4-2.5 GHz)



Zigbee packet format

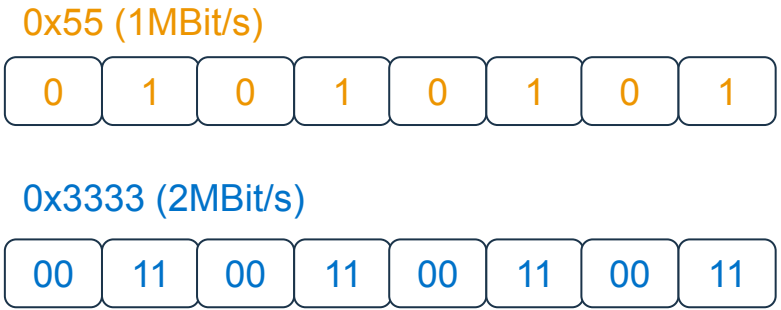
MOSART PROTOCOL

- **Mosart: proprietary protocol commonly used by wireless mices and keyboards**
  - based on a GFSK modulation at 1Mbit/s
  - no encryption, no pairing
  - the protocol can't be customized

Mosart packet format

- **Limitation: we can't select an arbitrary preamble using LE 1M (1Mbit/s)**
  - **solution**: using LE 2M and duplicating every bit (**0x5555** → **0x33333333**) **(preamble selection)**
  - Adding a set of **conversion functions**

- **Experiments:**
  - keylogger
  - keystrokes injection



GFSK 1Mbit/s - GFSK 2Mbit/s equivalence

ENHANCED SHOCKBURST PROTOCOL (ESB)

- **ESB: proprietary protocol used by wireless keyboards, wireless mices, drones, ...**
  - based on a GFSK modulation at 2Mbit/s
  - The protocol can be customized: **Logitech Unifying**
  - Logitech Unifying makes use of **channel hopping**

- **Manual channel scanning:** it could be automated and implemented directly in the firmware

- **Two modes are implemented:**
  - scanning: 0x000000AA (preamble)
    - we receive large blocks of raw demodulated data and look for valid packets into it
    - we extract the address and send it to the Host
  - sniffing / injection: 4 bytes of address (preamble)
    - we extract the payload and send it to the Host

ESB Packet format

Logitech M185 mouse (Logitech Unifying)

- **Experiments:**
  - **sniffing mouse packets**
  - **injecting mouse packets**

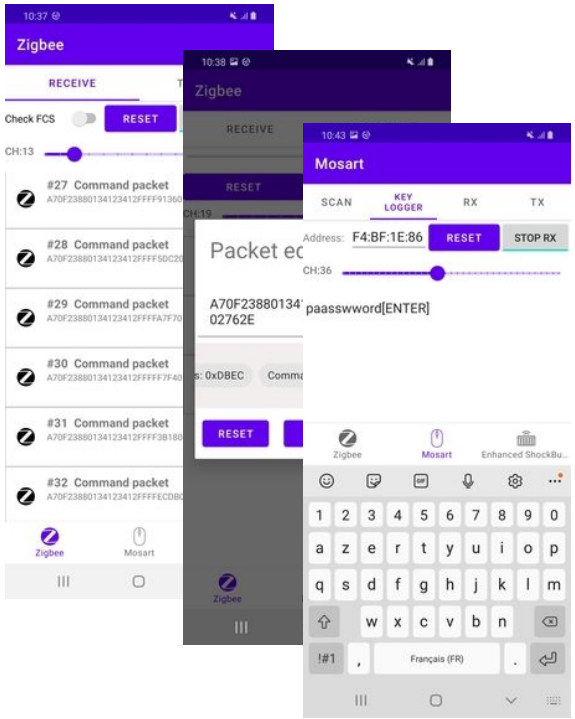
CONCLUSION AND FUTURE WORK

- **Critical attack:** we demonstrated the practical feasibility of implementing a mobile and cross-protocol attack platform on a smartphone
  - Offensive applications :
    - active and passive attacks
    - covert-channel attacks
    - cross-protocol pivoting attacks
  - It increases the attack surface of Zigbee, ESB and Mosart networks

- **It can probably be extended to other Broadcom / Cypress chips** if they supports extended advertising and to **other protocols** (ANT+, …)

- It could also be used to **communicate legitimately with these protocols** (eliminates the need for gateways)

Thanks for your attention !



The code is released as open-source software under MIT license :)

- **Android application:**

  ```
  $ git clone https://github.com/RCayre/radiosploit
  ```

- **Controller patches:**

  ```
  $ git clone https://github.com/RCayre/radiosploit_patches
  ```