

802.11

Couche transport				
Couche réseau				
Couche liaison	802.11 MAC			
Couche physique	802.11	802.11b	802.11a	802.11g

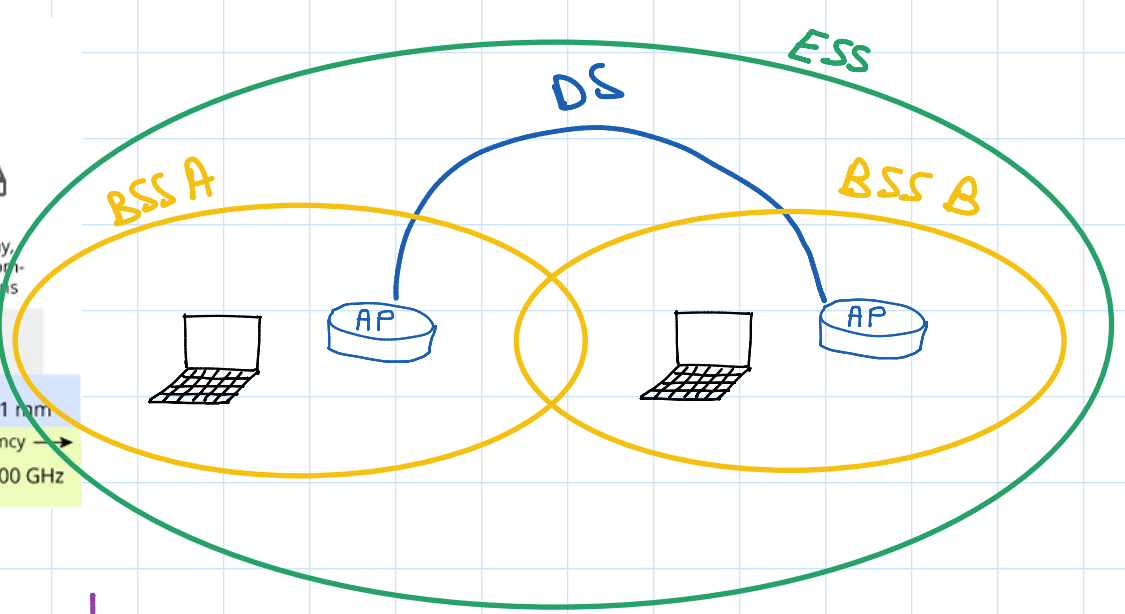
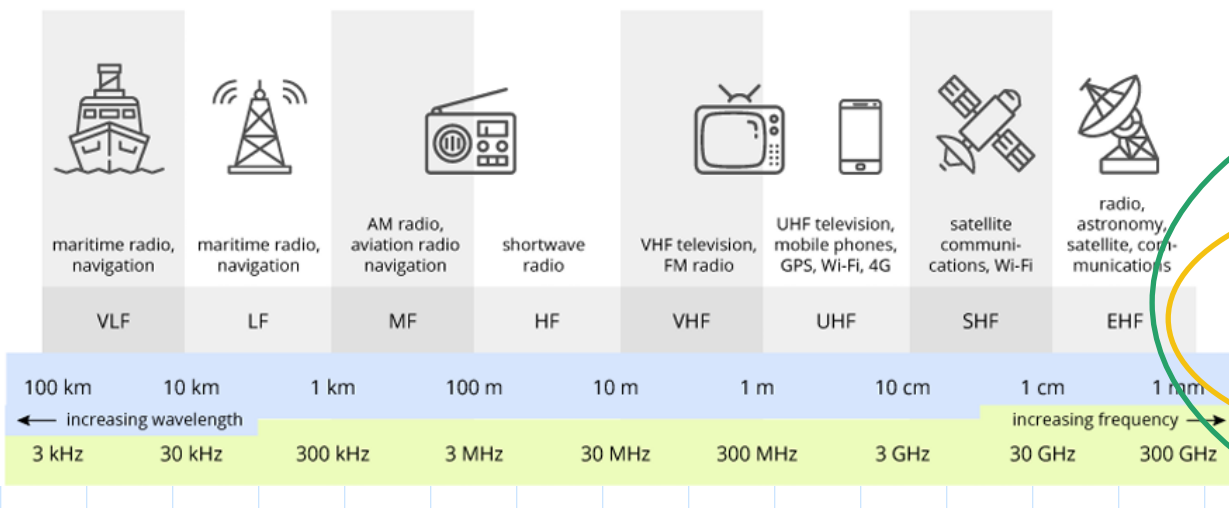
Peuvent étendre leur réseau :
① WDS (Wireless Distribution System)
② structure maillées

				ETSI
IEEE	802.20	WWAN	WAN	GSM, 3G, Edge
IEEE	802.16	Wi-Max	MAN	Hiperman
IEEE	802.11	Wi-Fi	LAN	Hiperlan
IEEE	802.15	Bluetooth, ZigBee, Wireless-usb	PAN	Hiperpan

- AP: Access Point
- STA: Stations
- BSS: Basic Service Set

- 2 modes :
- ① infrastructure : réseau géré par l'AP
 - ② Adhoc : chacun maîtrise son réseau

- * Grandeur d'une onde EM :
- Fréquence (inverse période)
 - Longueur d'onde
 - Amplitude
 - Puissance
- * Transmission Radio :
- Portée du signal
 - Débit du signal
 - Modulation (DSSS et OFDM)



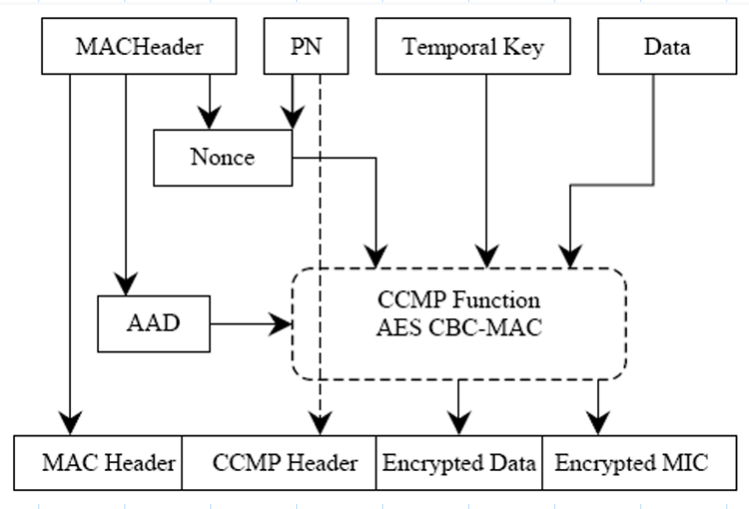
Affichage débits et fréquences supportés par une carte WIFI : `$ iw phy`
Afficher les AP environnants : `$ iwlist wlan0 scanning` ~ `$ iwlist wlan0 keys/enc/auth/wpa`
802.11 repose sur du {
- trafic de contrôle (bonne arrivée des trames entre pairs) → **medium** (PS-Poll, RTS, CTS, ACK, CF-End, CF-Ack)
- trafic de gestion (établissement et maintien de la connexion) → **WIFI** (association, sondage, ATIM, Beacon, auth, de-auth, de-association)
- trafic de données (transmission données utilisateur) → **MAC**, payload (IP, TCP, HTTP...)

- Procédure d'association (mode infra) :
- choisir ESS via l'ESSID
 - choisir le BSS dans l'ESS
 - demander à s'authentifier (open vs shared)
 - demander à s'associer
 - la communication peut se faire via l'AP.

- 3 schémas de sécurité 802.11 :
- * WEP (Wired Equivalent Privacy)
 - * WPA (Wireless Protected Access)
 - * 802.11i / WPA2 *
- concerne le trafic de données

- confidentialité : AES
- intégrité : CCMP/AES
- authentification
 - WPA-PSK (pre shared key)
 - WPA-EAP
- anti-rejeu, distribution de clés

WPA2 : Protocole AES-CCMP



```
Connexion à un réseau ouvert
# iwconfig wlan0 essid "TLS-SEC-open" channel 3 key off
# dhclient wlan0

Connexion à un réseau WEP
# iwconfig wlan0 essid "TLS-SEC-WEP" channel 4 key 1234ABCD
# dhclient wlan0

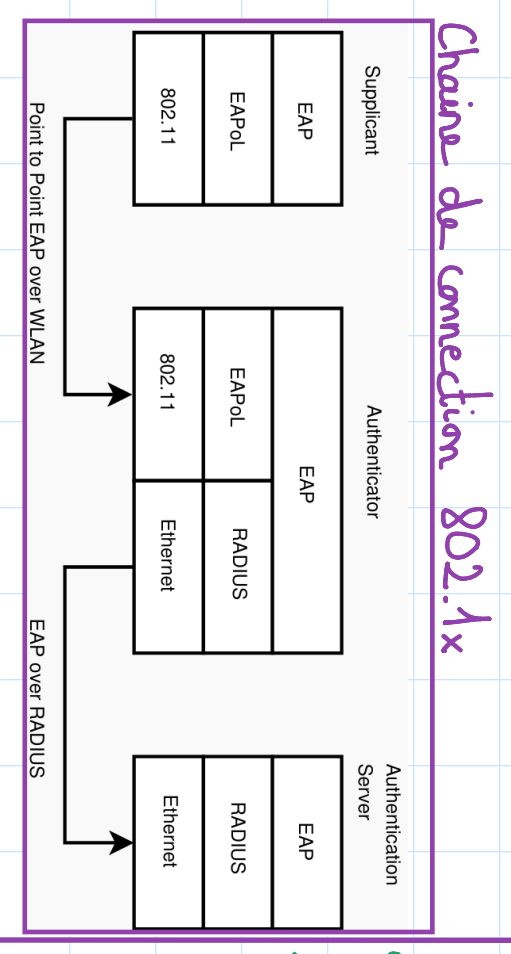
Connexion à un réseau WPA PSK avec wpa_supplicant

network={
    ssid="TLS-SEC-WPA"
    scan_ssid=1
    proto=WPA
    key_mgmt=WPA-PSK
    psk="monmotdepassestressecurise"
}

# wpa_supplicant -D wext -i wlan0 -c /etc/wpa_supplicant.conf &
# dhclient wlan0

Ou bien :

# wpa_supplicant -i "$IFACE" -c <(wpa_passphrase "$SSID" "$PASSPHRASE")
```



WPA2 : 2 authentifications :

- WPA - Personnel : WPA - PSK (secret pré-partagé)
- WPA - Entreprise : WPA-802.1x ou WPA-EAP

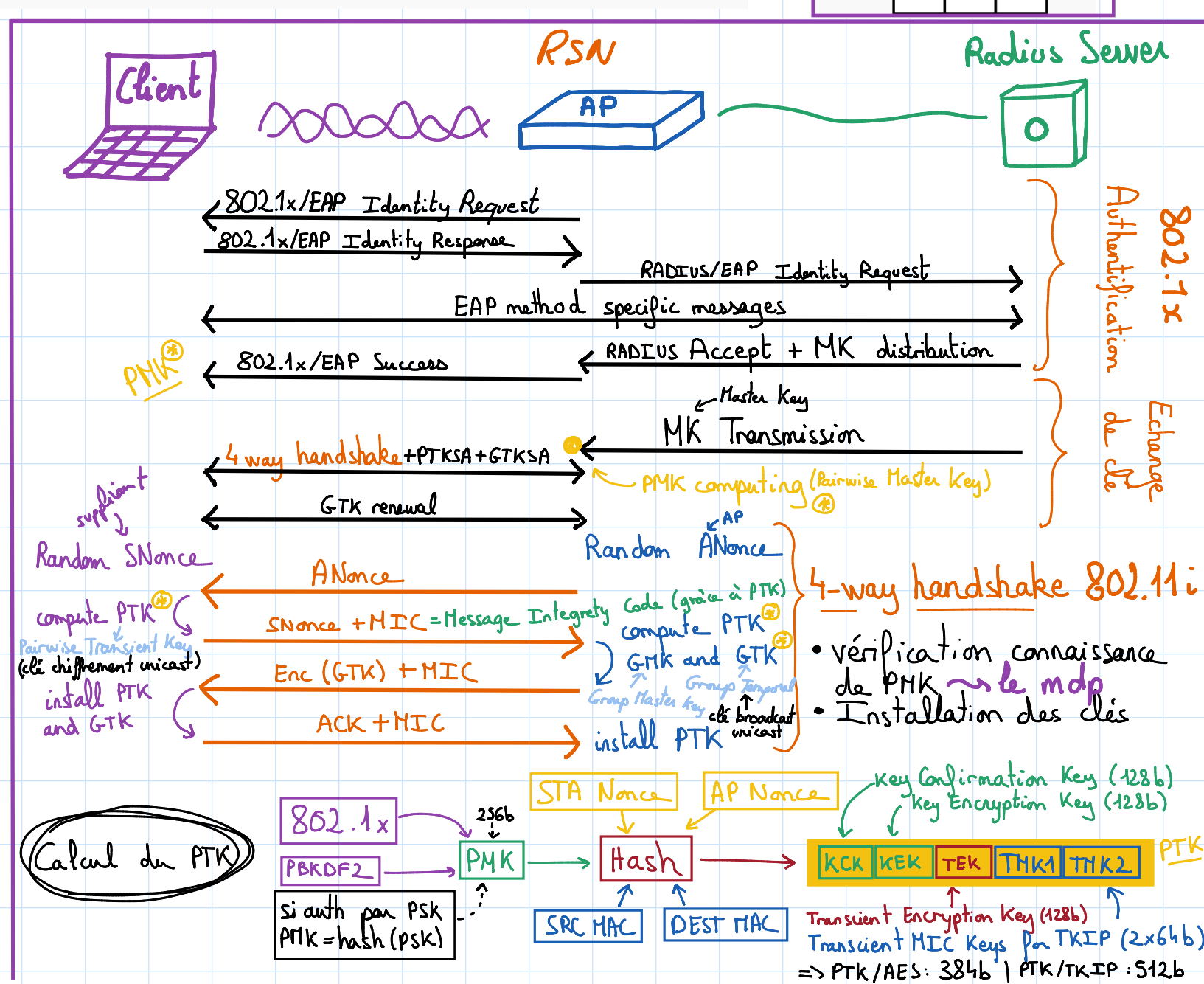
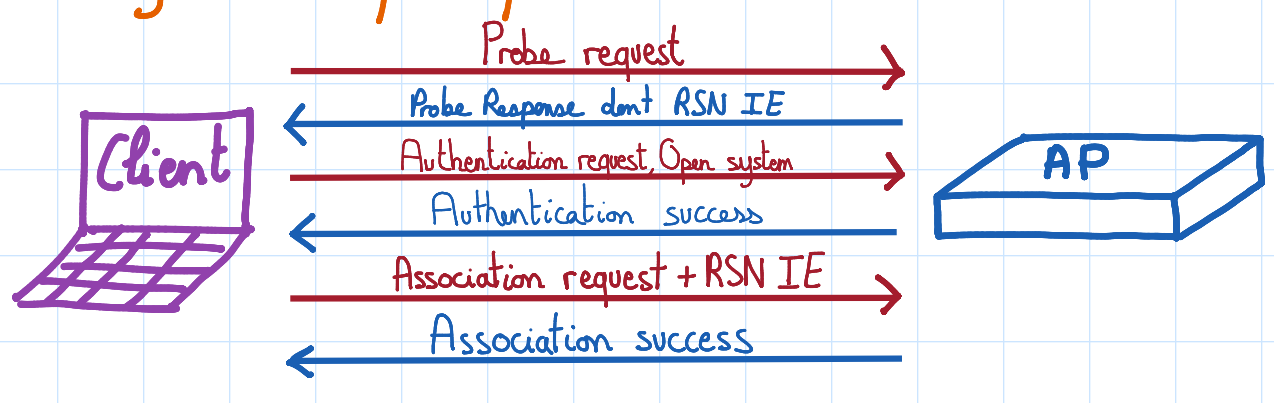
802.1x : ~ même filaire

- authentification forte
- gère distribution et préparation des clés
- timeant, distinction de ports logiques...

⇒ 802.11i RSN (Robust Security Network)

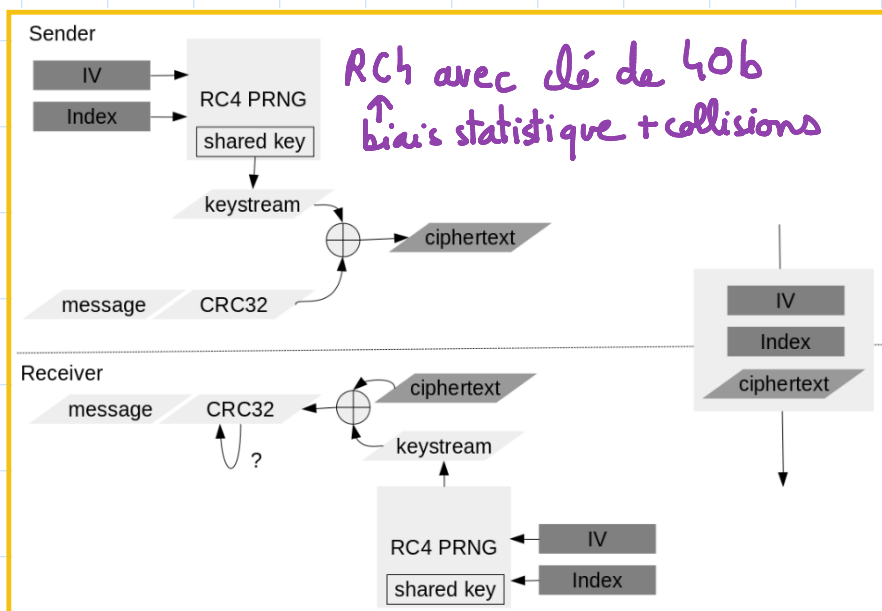
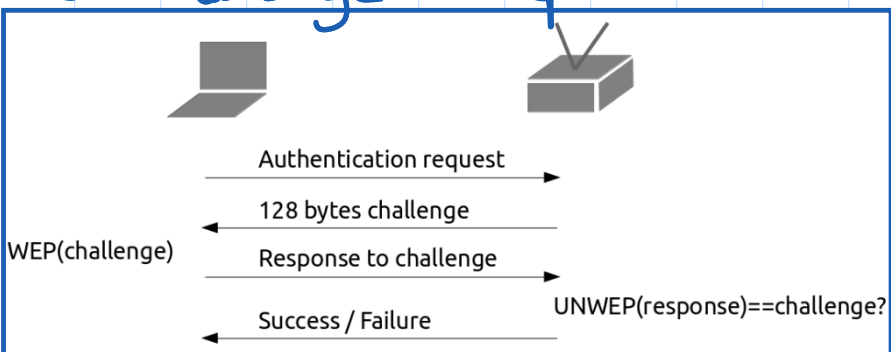
- ↳ négociation politique de sécurité
- ↳ authentification 802.1x
- ↳ distribue les clés
- ↳ protection des communications réseaux

Négotiation politique de sécurité RSN (802.11i)

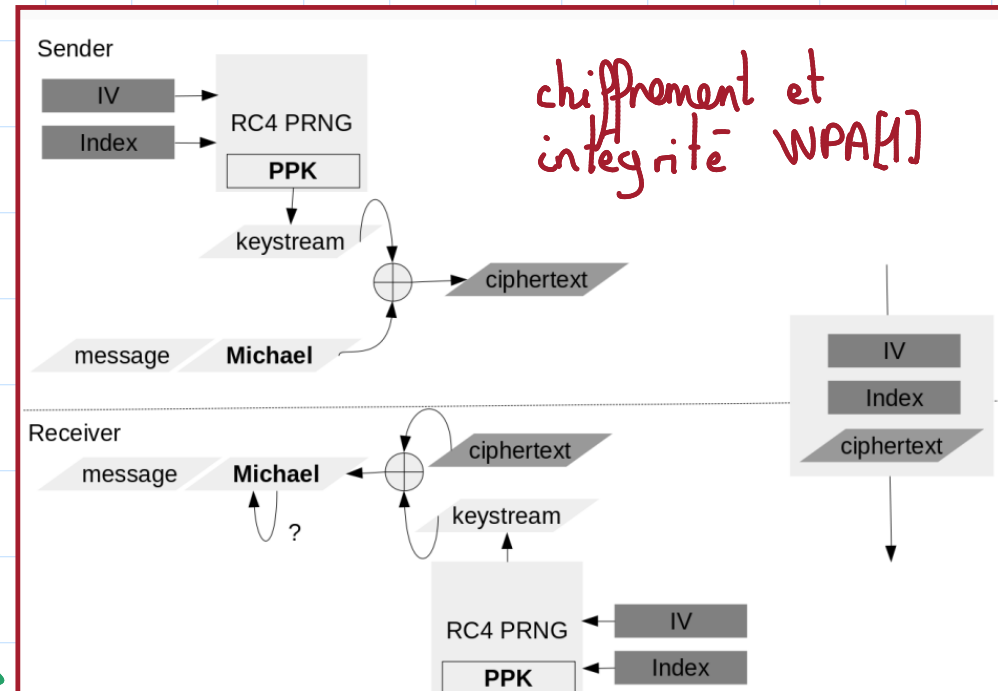


WEP (Wired Equivalent Privacy)

- first try → weak protocol
- basé sur RC4, symétrique
- clé pré-partagée entre AP/Client (40b)
- "intégrité": CRC32
- authentification avec 4way handshake avec challenge et réponse



RC4 avec clé de 40b
↑ biais statistique + collisions



chiffrement et intégrité WPA1

WEP et WPA1 sont d'anciens standards

→ WPA3? → personal → Simultaneous Authentication of Equals (SAE)
↳ protocole d'échange de clés → pas d'attaque offline
↳ confidentialité persistante
→ professional

déjà cassé
donc jamais
vraiment utilisé

- configurer en WPA AES CCMP
- renouveler les mdp en WPA-PSK
- désactiver WPS
- superviser (avec outils)
- MJS OS/firmware

- SSID cloaking
- Filtrage d'@MAC source
- Isolation des stations

- auth des trames de gestion unicast avec PTK
- auth des trames de gestion multicast avec IGTK
- implémentations rare (CISCO MFP, hostpad)

protection trafic de gestion (802.11w)

protection AP

autres mesures (non 802.11)

protection stations

Autre

DOS

- couche P: perturbation électromagnétique
- couche MAC: spoofing de l'@MAC de l'AP
 - envoi: désassociation aux stations
 - envoi continu de trames CTS (mise en attente d'envoi)
- contre-mesure: 802.11w trafic contrôle/gestion
- détection: Alerte Kismet (DEAUTHFLOOD, BCASTDISCON, DISASSOCTRAFFIC)

Attaques sur 802.11: aircrack-ng, Boop Suite, bully ...
Outils de défense: kismet, pidsense, waidps ...

* IV: vecteur d'initialisation

Spécifique WEP

- retrouver clé WEP
- collisions des IV* réduisent l'effort de brute force → aircrack-ng
- attaque chopchop → aireplay-ng
- décrypter les données (sans clé)
- authentification unilatérale donc rogue AP et MITM possible → pas d'auth de l'AP pour la station

Contre-mesure: pas de WEP, éviter que les AP puissent être downgrade

détection: alerte Kismet (CRYPTODROP)

Spécifique WPA aircrack-ng

- dictionnaire contre la PSK *
- KRACK (Key Reinstallation Attack)
- Attaques connues sur PEAP
 - ↳ dictionnaire sur EAP-MD5

Contre-mesures:

- PSK longue et complexe, la renouveler
- utiliser autre auth que EAP-MD5

détection: Kismet (NONCEDEGRADE)
↳ contre KRACK

Ecoute:

- sniffing en mode monitor!
- découverte passive des AP (grâce aux beacons envoyés régulièrement par les AP) → aircrack-ng → kismet
- découverte active (par envoi de probe requests) → iw dev \$iface scan, Netstumbler
- Wardriving
- contre-mesures: masquage SSID
 - ↳ pas de ESSID dans les beacons
 - ↳ on répond aux probe requests que si on a l'ESSID.

Modification du trafic:

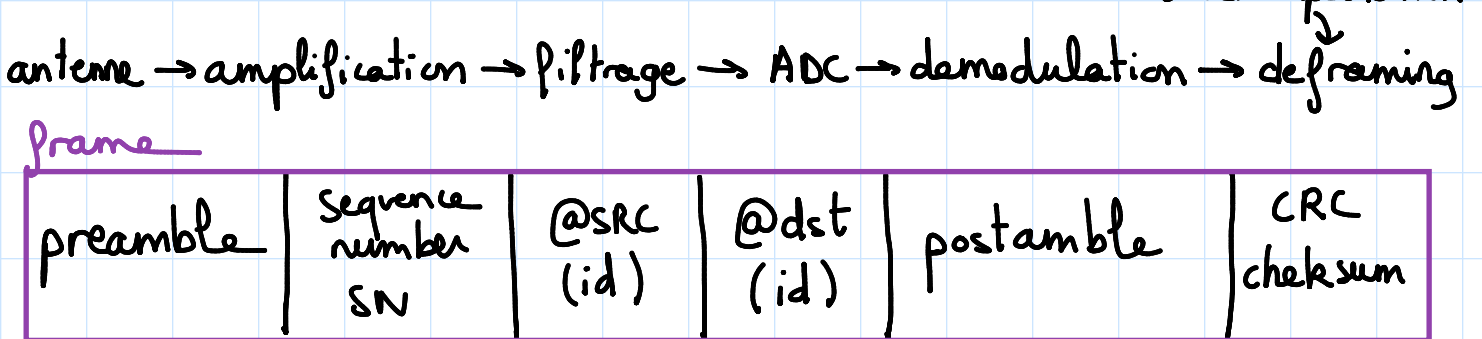
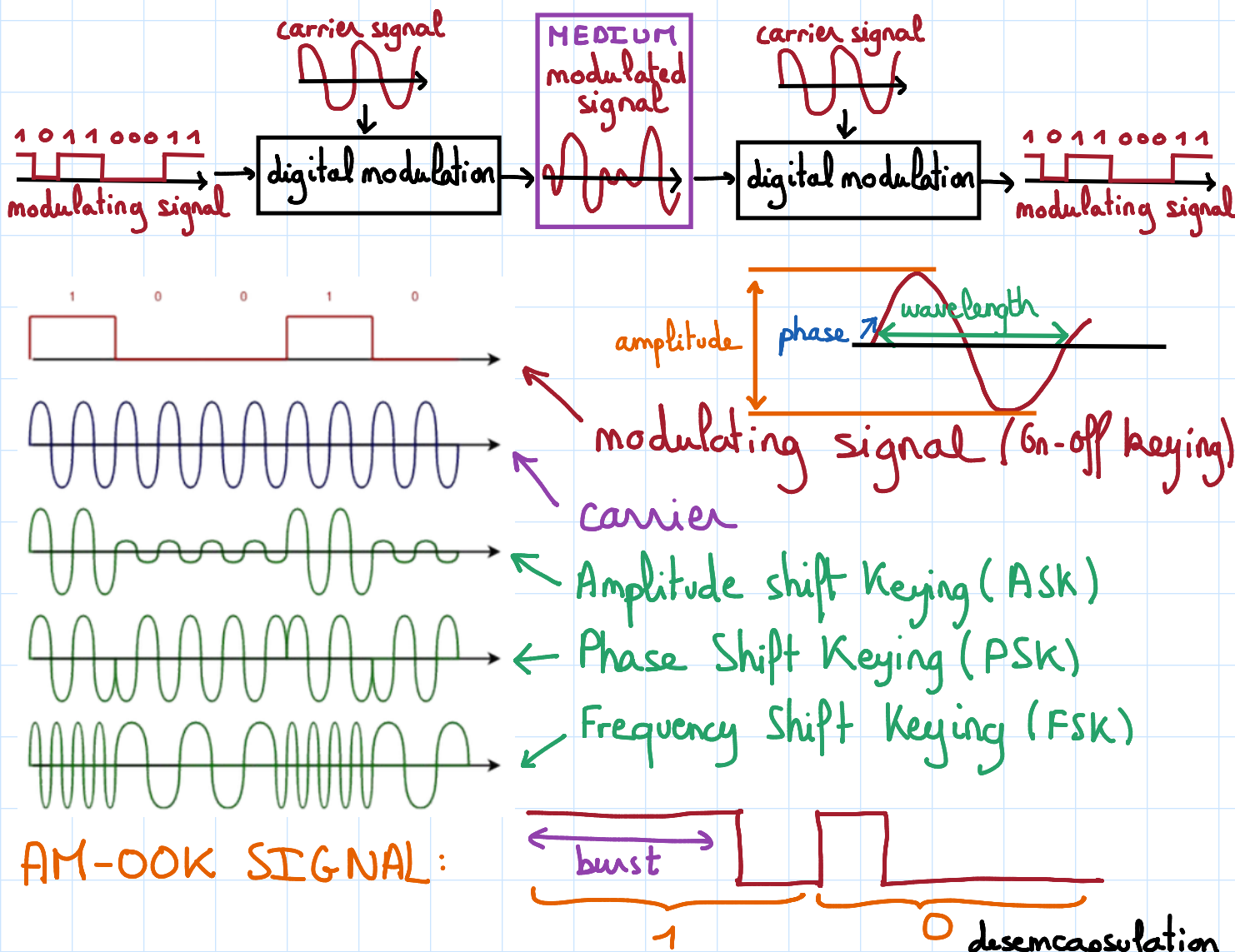
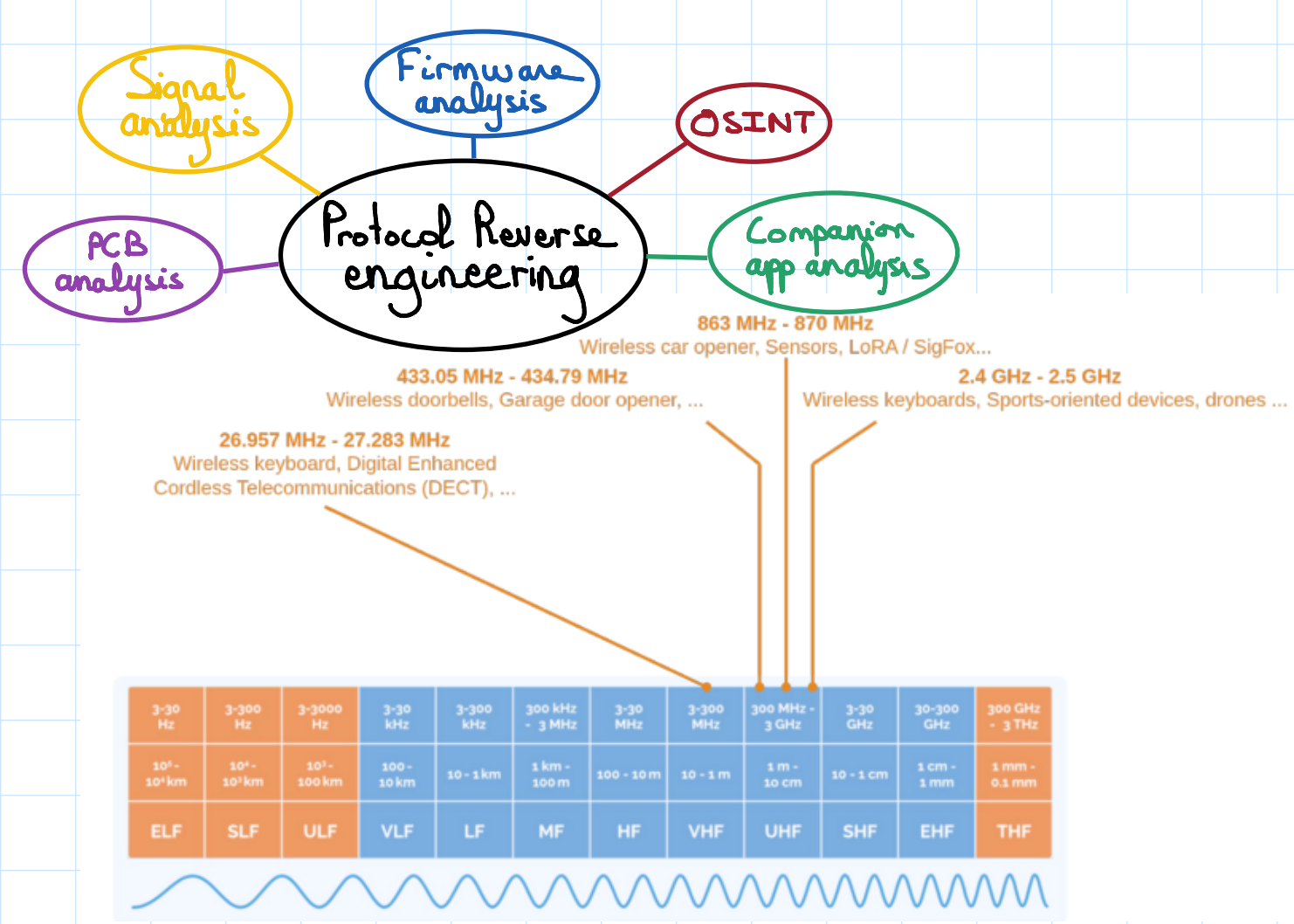
- rejeu en réseau ouvert ou WEP
- injection de trafic arbitraire
 - ↳ peu de vérifications d'intégrité (absence, CRC32)
- contre-mesures: pas de WEP, portails captifs insuffisants en réseaux ouverts (cache supérieure...)

Autre types d'attaques

- découverte d'une commande cachée (RE) dans l'AP
 - ↳ il faut connaître le numéro de série de l'AP
- condition d'arrêt du brute force du mot de passe à passer à cette commande
- résultat: shell root sur tous les AP du fabricant.

* Attaque par dictionnaire de la PSK

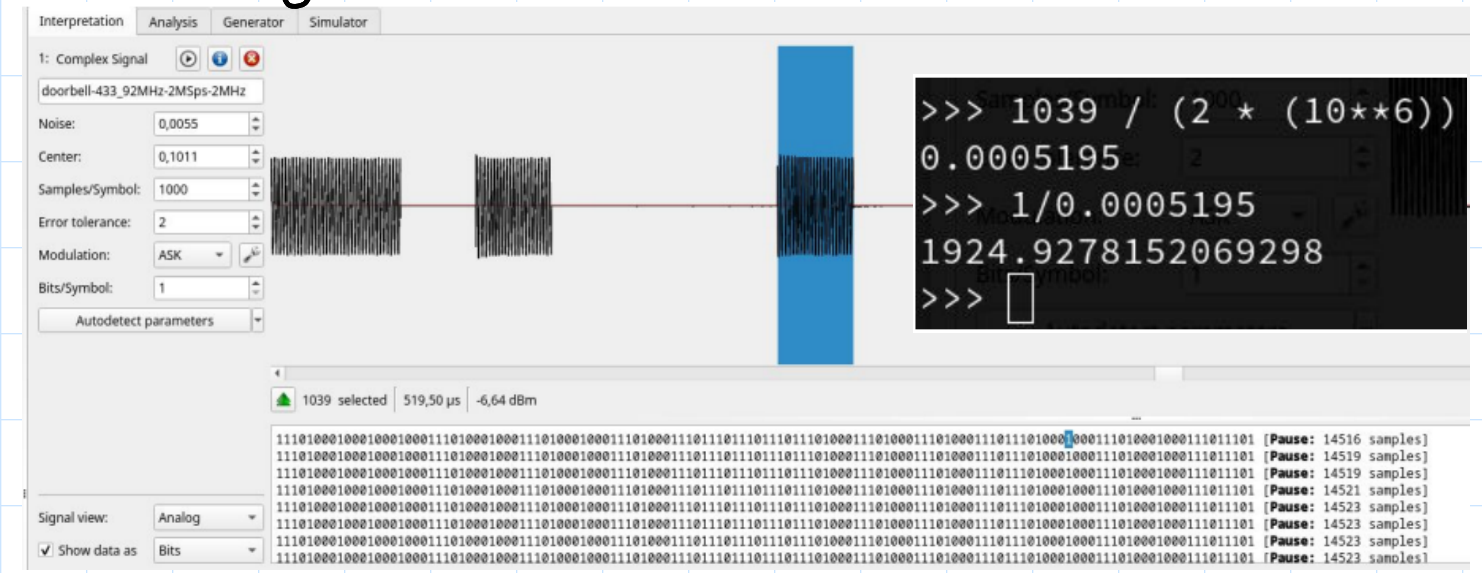
- il faut capturer 1 MIC valide
- identifier une station légitimement associée
- désauthentifier la station
- capturer le 4-way handshake
- comparer le MIC au résultat calculé de la PSK candidate



Wireless doorbell

HackRFOne: Half Duplex Software Define Radio 1MHz ~ 6GHz

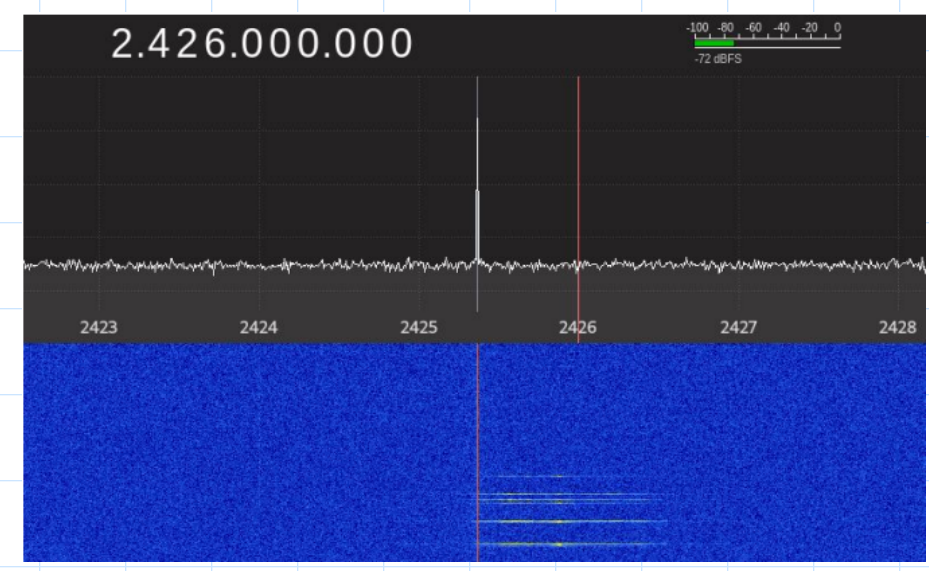
↳ using Universal Radio Hacker (URH)



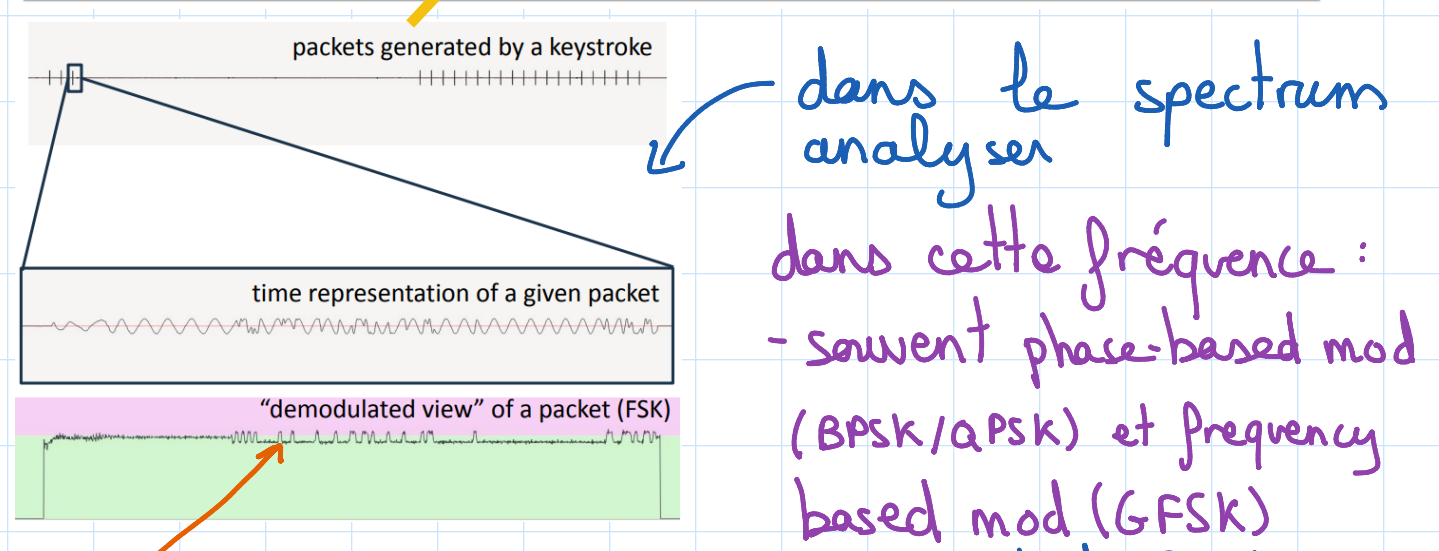
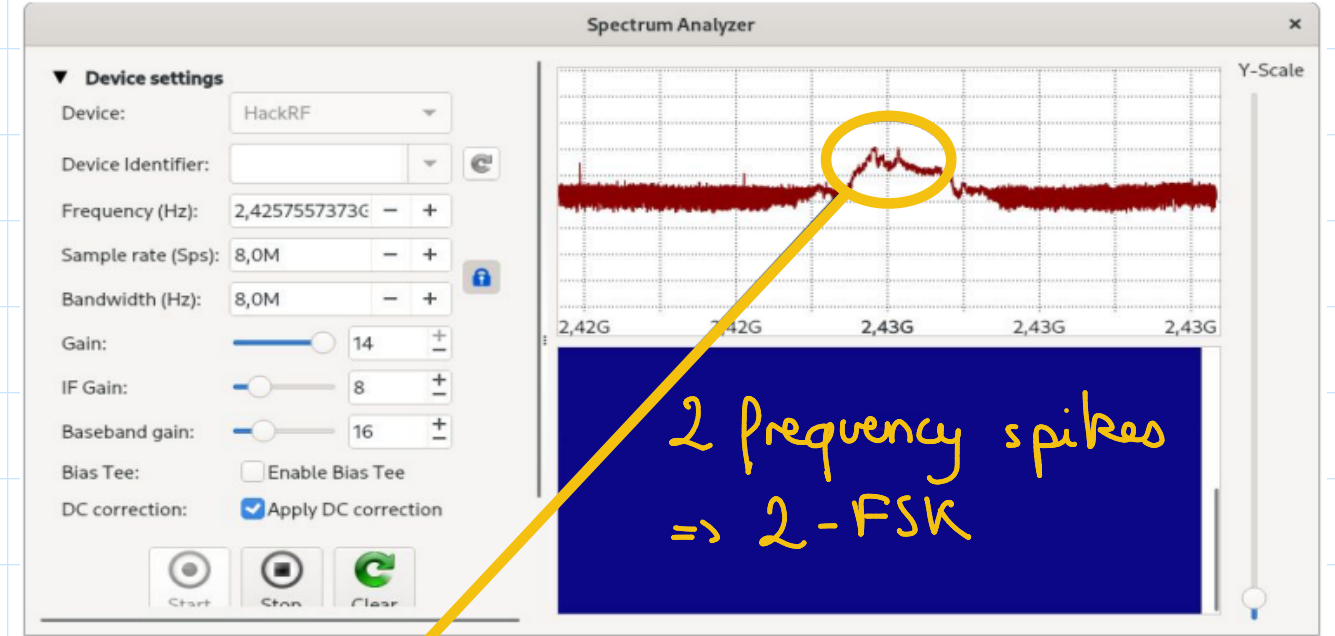
- signal tout au rien ⇒ modulation ON-OFF (OOK/ASK)
- trouver la fréquence: écouter avec un sample rate cohérent
- on a moyen d'estimer le symbol rate: ~ 1925 sps (per sec)
- on cherche le preamble et le postamble et voilà !!
- on prend la frame la plus accuante (celle avec le moins de bitflips probable)

Wireless Keyboard

- proprietary protocol 2GHz
- tool: GQRX → on cherche la fréquence



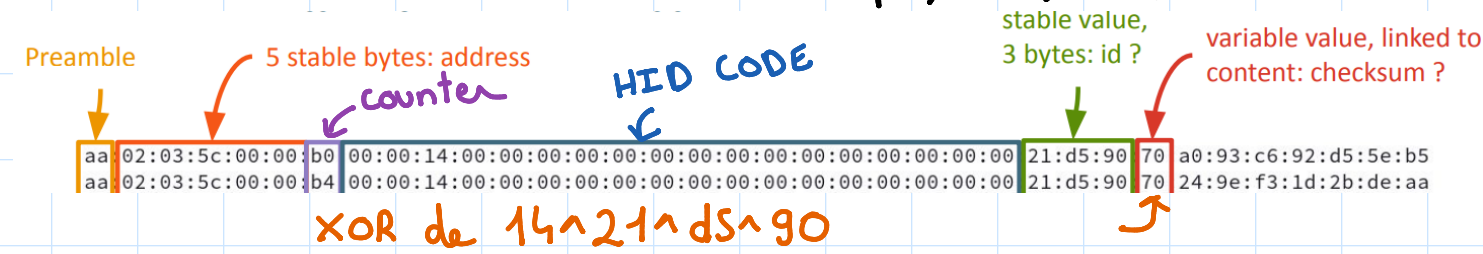
- on gène de l'activité avec la devise
- on se balade dans les fréquence a la recherche du signal → 2462MHz
- signal sporadique ⇒ channel hopping utilise plusieurs fréquences
- 2,4-2,5 GHz très utilisé ⇒ bruité !!!
- tool: URH → on cherche la modulation



le ⊕ petit bit encodé nous donne: 8 samples/bit at 8MSps ⇔ 1Mbits/s

⊗ Méga Sample per seconds

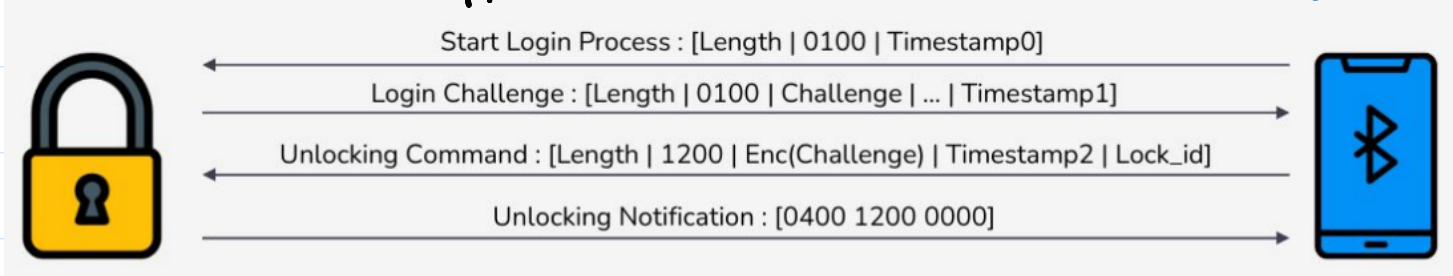
- on cherche le préambule: 0x000000AA
- maintenant on sort le nRF24
- ↳ GFSK modulation at 250kbps, 1M, 2M, 2.4G band



- on sait faire une requête ⇒ injection
- ⇒ KEYLOGGER + KEYSTROKES INJECTION

SMART PADLOCK → BLE

- sniffing non trivial ⇒ channel hopping algorithm
- companion app: android/iOS
- MITM avec l'app: section qui se répète: AES-ECB encryption?
- reverse de l'app: secret hardcoded ~ AES-ECB key



- We need to:**
- Extract the challenge from the first message
 - Send a valid message with the encrypted challenge, a "correct" timestamp (Timestamp0 < Timestamp1 < Timestamp2) and the lock id

ANT PROTOCOL ~> proprietary, security by obscurity

- OSINT : 2.4GHz, ANT+ (with keys), ANTFS (File Sharing over the air)
- OSINT : FCC ID ~> Federal Communications Commission test report, photos, specs
-> ici test report -> GFSK, internal photo: nrf24p2
- reverse -> "network key" genere une valeur de 2 octets en preamble.
- on a moyen de scanner, synchroniser, sniffer, injecter de la donnée arbitraire (heart rate?)

* WEAK CONFIGURATION - ZigBee Default Key

* NO ENCRYPTION - "Mosant" protocol

- propriétaire (HP, Anker, Lenxington, EagleTele...)
 - quand keystroke, la frame a un champ "key code" qui indique la clé utilisé.
- > bruteforce (255 valeurs)

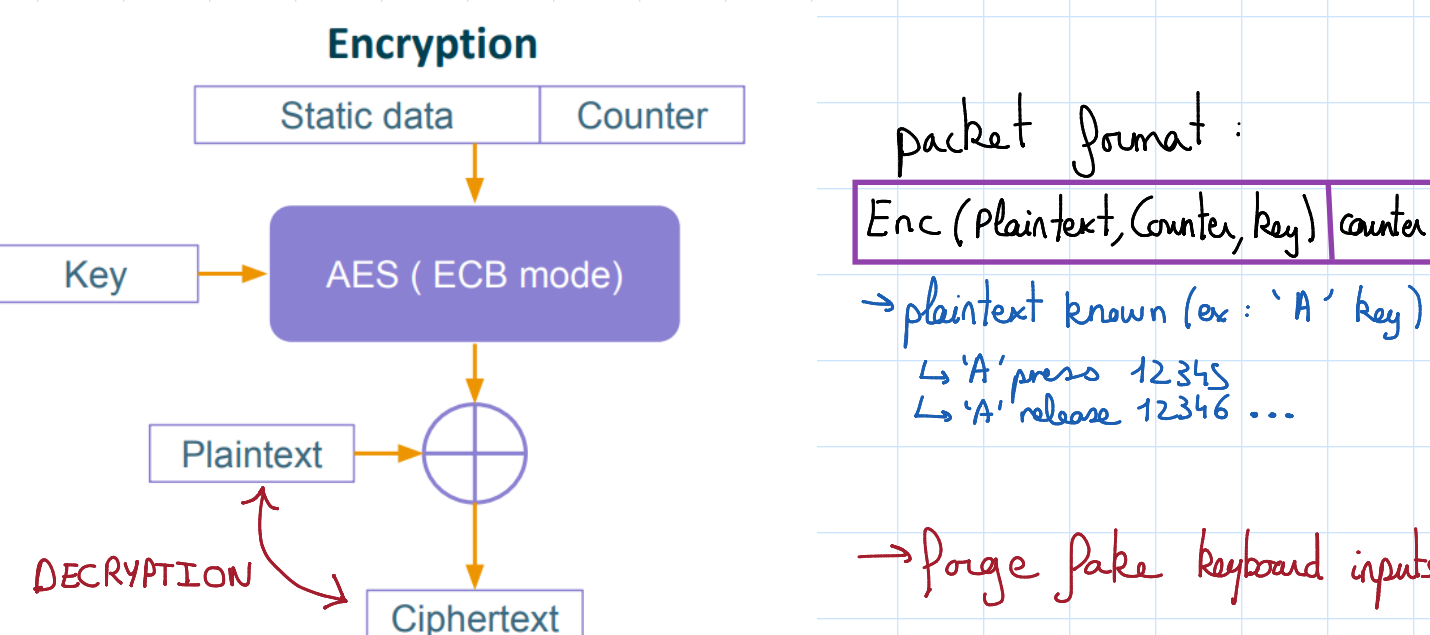
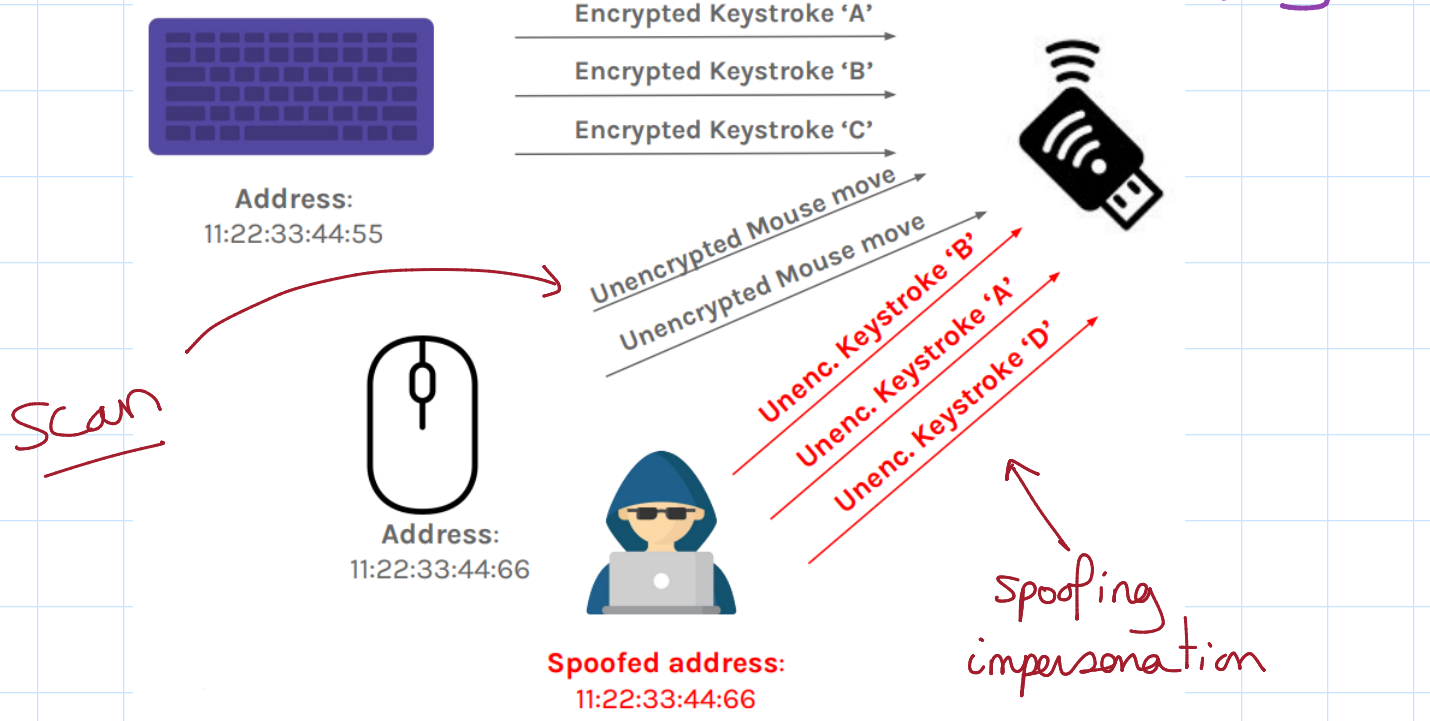
Preamble 2 bytes	Address 4 bytes	Type 4bit	Seq. Num 4bit	Key state 1 byte	Key code 1 byte	CRC 2 bytes	Postamble 2 bytes
---------------------	--------------------	--------------	------------------	---------------------	--------------------	----------------	----------------------

* WEAK ENCRYPTION - Microsoft "encryption"

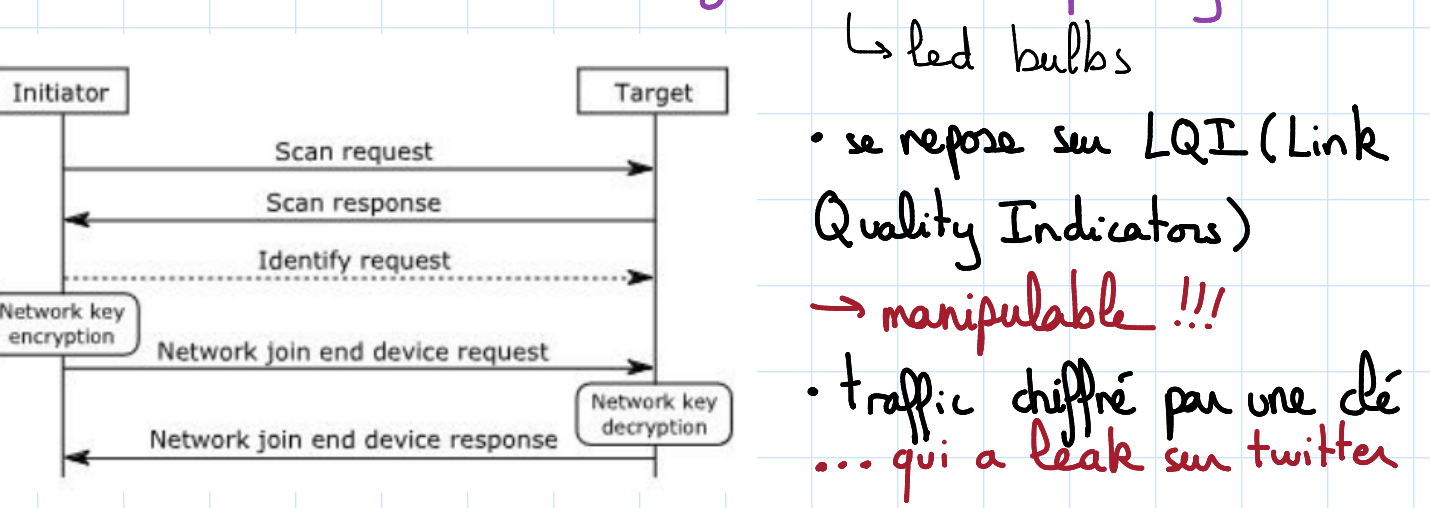
- "chiffré" par XOR par une keystroke.



* INCONSISTENT USE OF ENCRYPTION - Logitech unifying

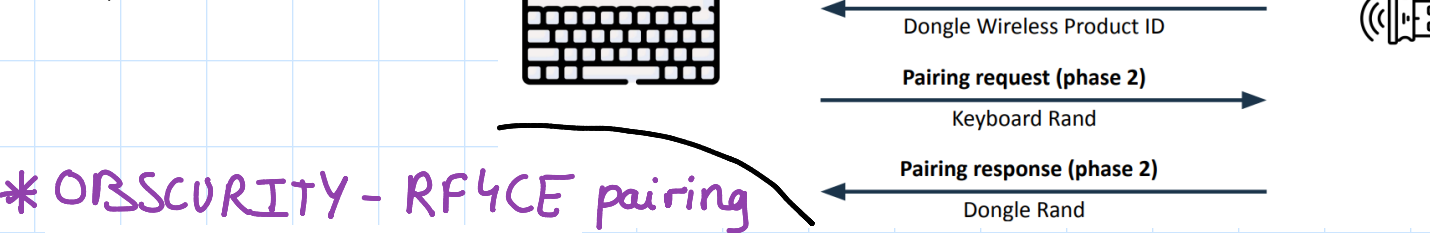


* RELYING ON 4 PARAMETERS - Zigbee touchlink pairing

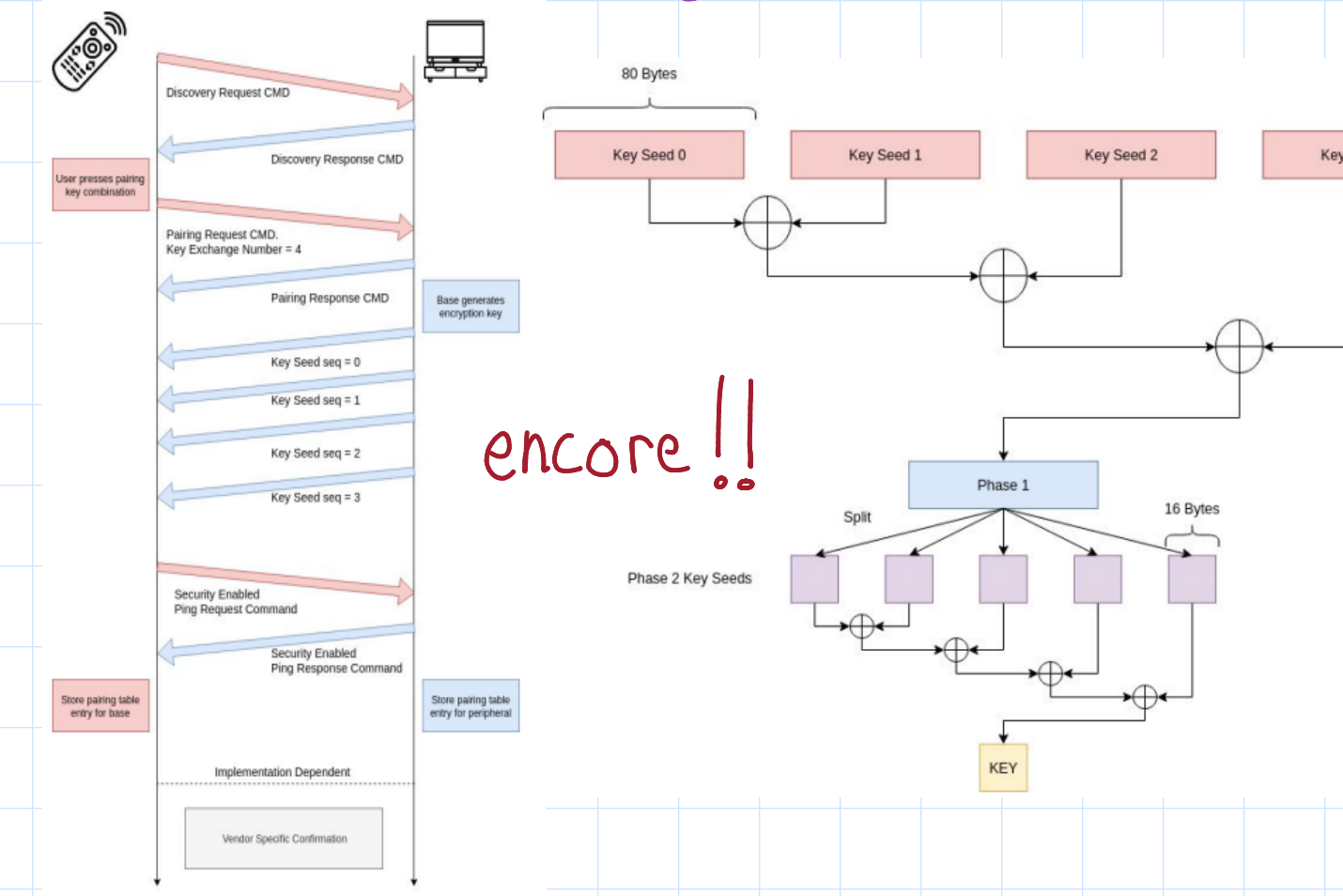


* SECURITY BY OBSCURITY - Logitech unifying pairing

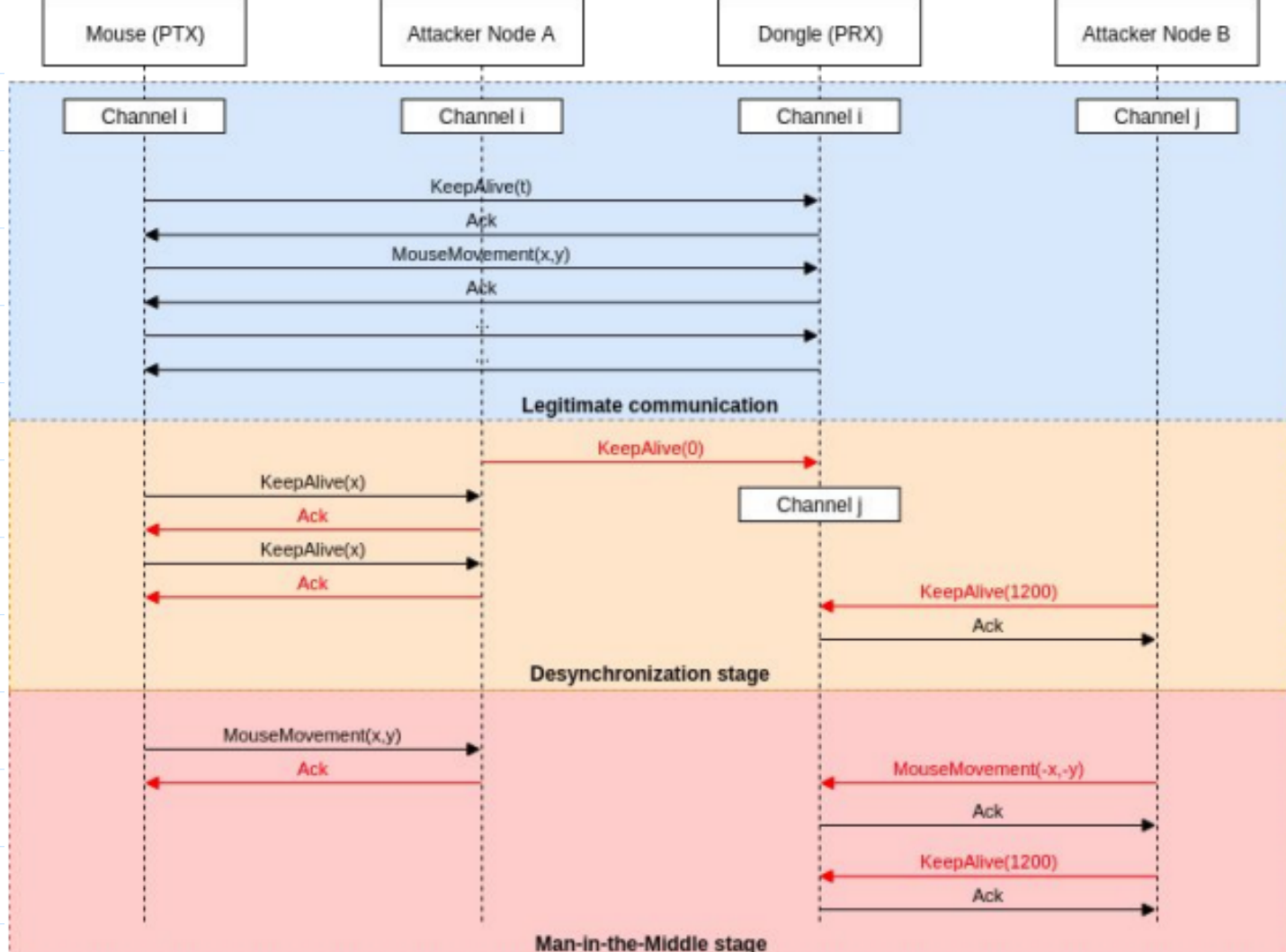
- pas de DH (Diffie-Hellmann), transmission de clé en clair puis petit algo.



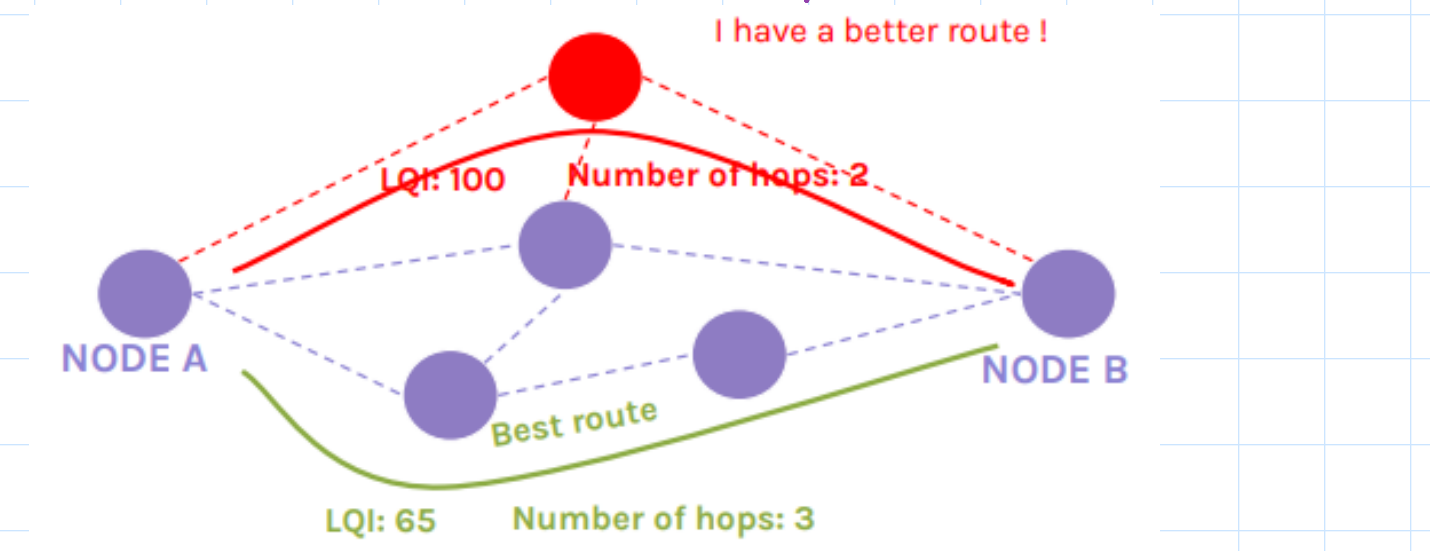
* OBSCURITY - RF4CE pairing



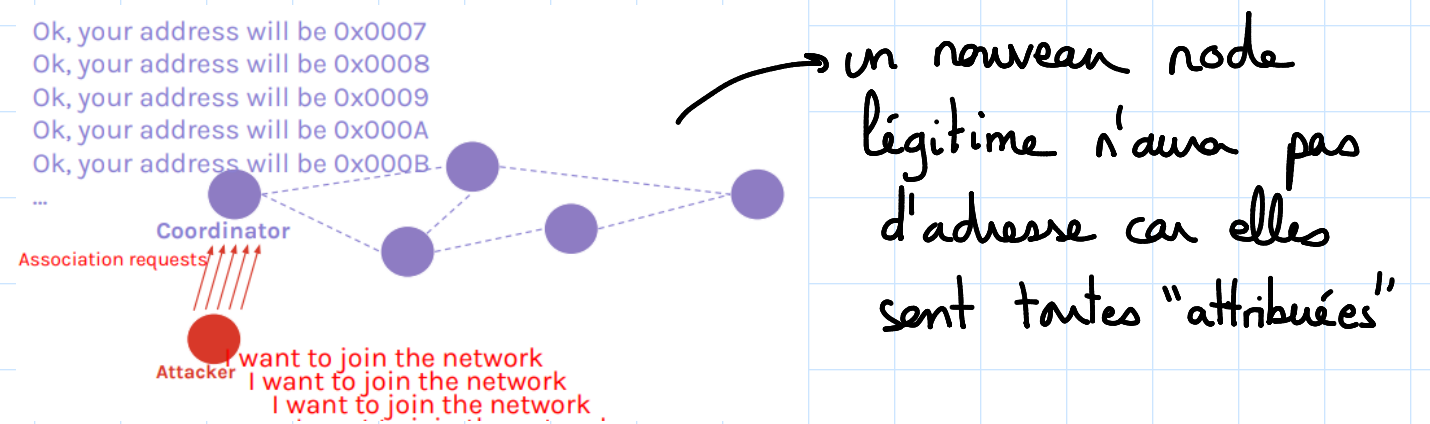
* MITM - Logitech unifying



* ROUTING ATTACKS - example on Zigbee



* RESOURCE EXHAUSTION



CCL: designing protocol is hard
wireless medium ~> open by design
model checking? ex: pro verif -> costly