VULNÉRABILITÉS SUR LORAWAN

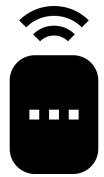
Rejoue de trame sur le LoRa

MQTT

Attaques potentielles

UDP

Gateway

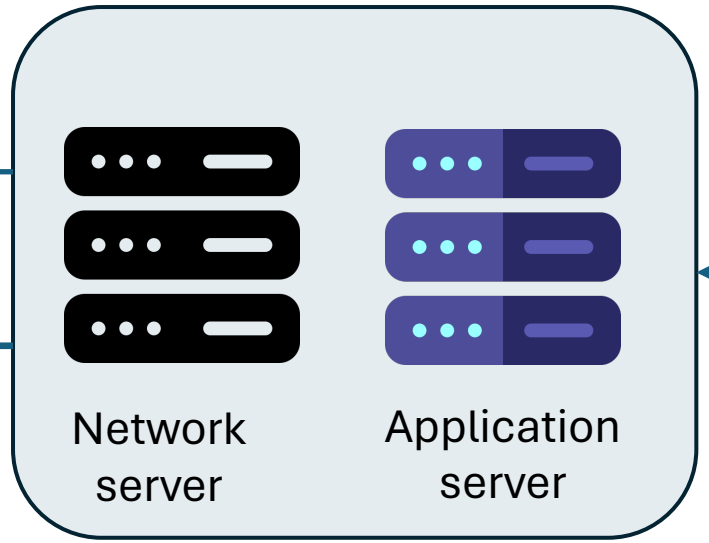


modulation

LoraWan End-device

internet

Gateway



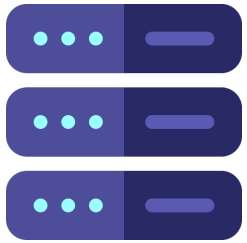
internet

Authenticité
Intégrité

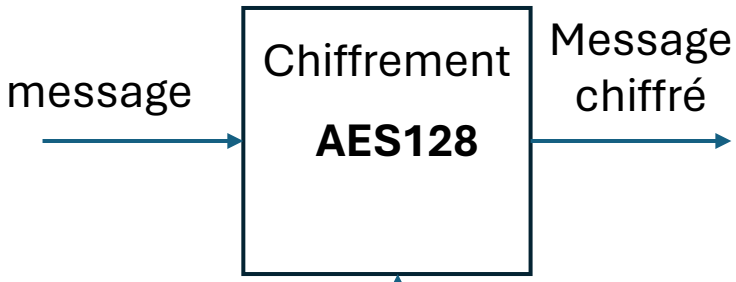
Confidentialité

CHIFFREMENT SYMETRIQUE

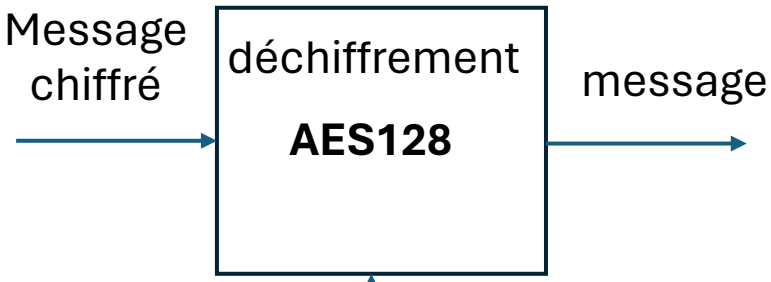
LoraWan
End-device



Application
server

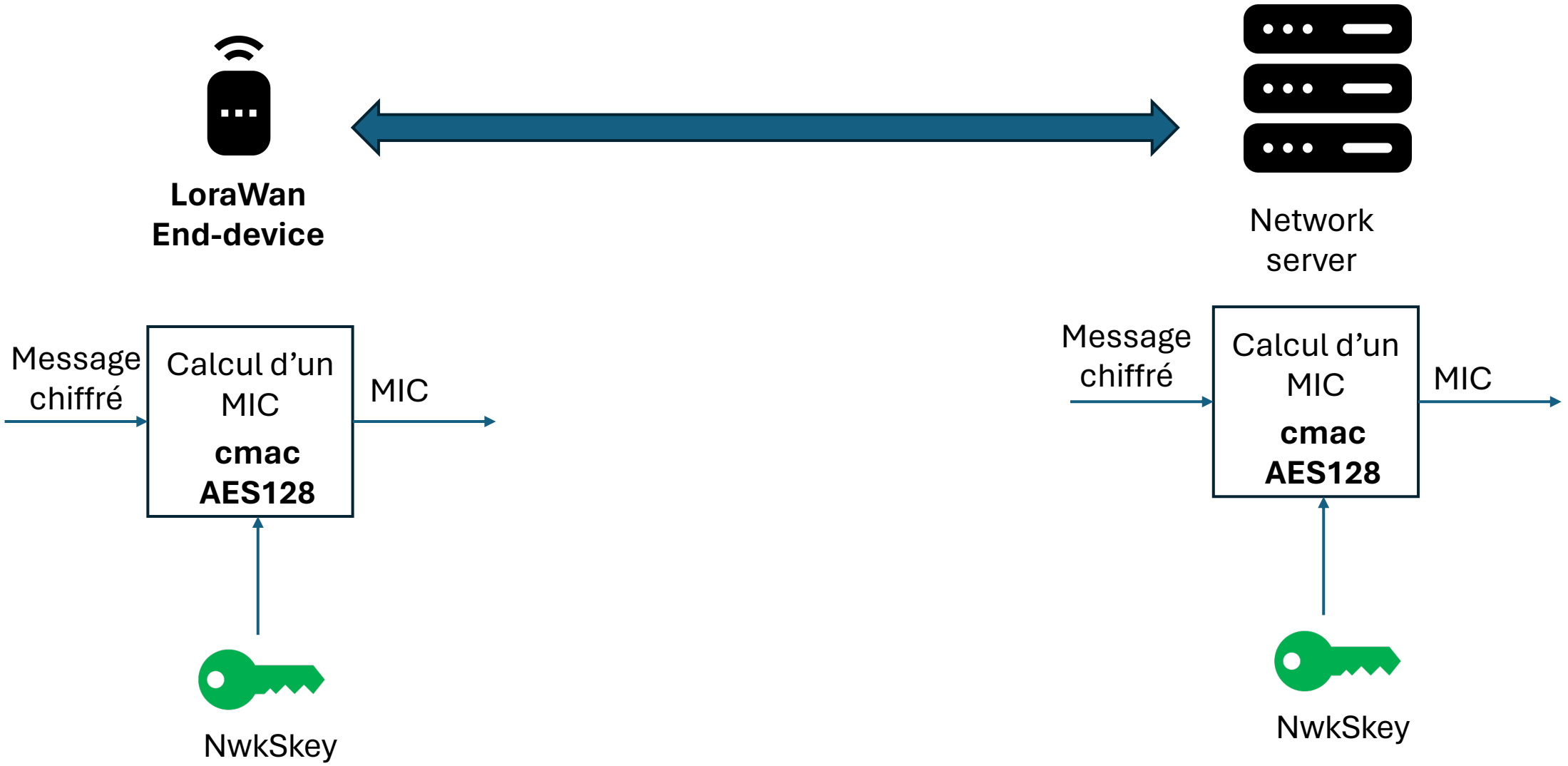


AppkSkey



AppkSkey

AUTHENTIFICATION ET INTÉGRITÉ

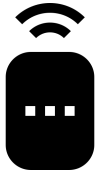


REJOUE DE TRAME LORAWAN

Rejoue de trame sur le LoRa

Attaques potentielles

Gateway



LoraWan End-device

LoRa™ *modulation*

CHIFFREMENT DES DONNÉES ET AUTHENTIFICATION

Attaques potentielles

MQTT

Network server

Application server

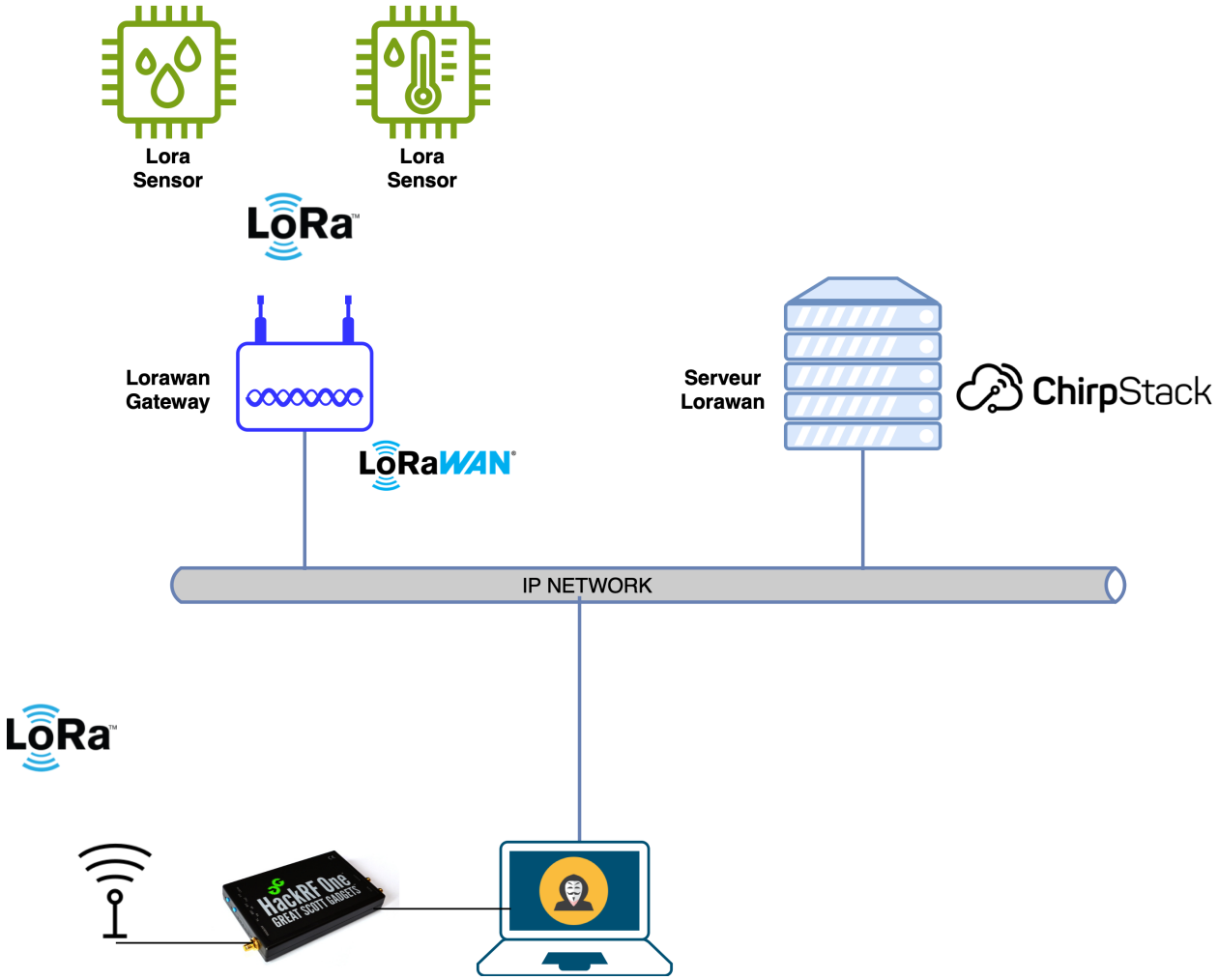
internet

MQTTS avec Certificats TLS

Authenticité
Intégrité

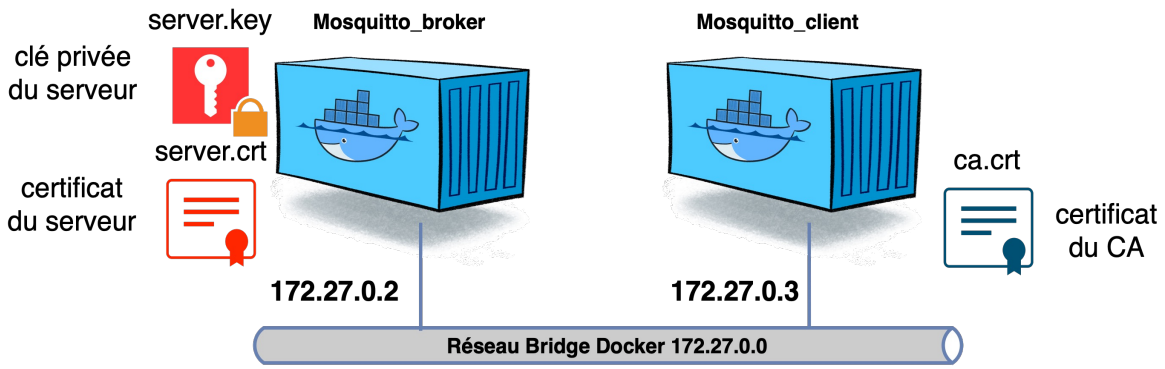
Confidentialité

TP SÉCURITÉ LORAWAN



- Réaliser une attaque Arp spoofing sur la passerelle Lorawan.
- Sniffer les trames UDP avec Wireshark et Scapy.
- Déchiffrer avec lora-packet sur Nodejs.
- Utiliser un hackrf pour rejouer les trames avec URF(Universal radio Hacker)

TP MQTTS




```
mosquitto_sub -h 172.27.0.2 -p 8883 --cafile /ca.crt -t your/topic
```

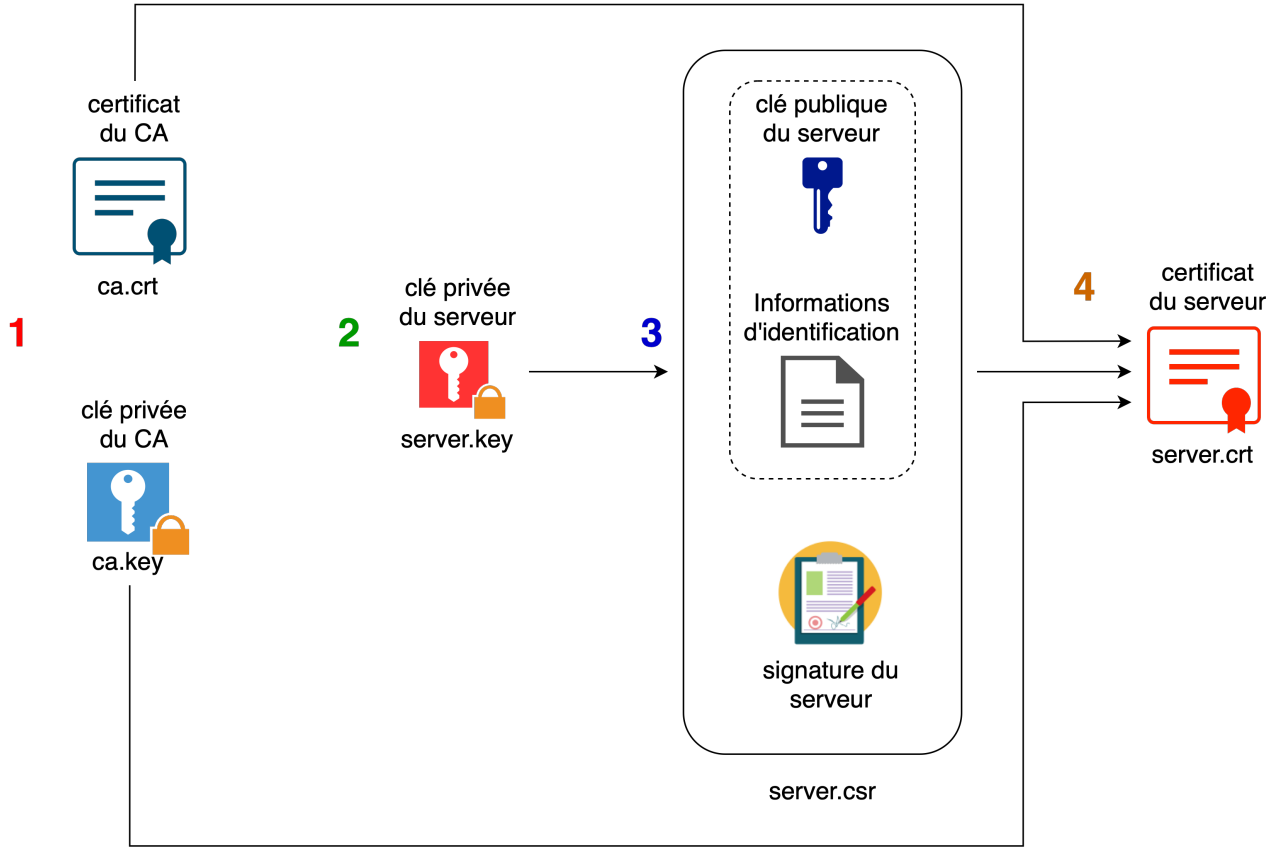
Objectif terminal :

Sécuriser une connexion mqtt pour le client et le serveur.

Objectifs intermédiaires :

- Créer des conteneurs docker en utilisant docker-compose.
- Créer un certificat SSL.
- Configurer un broker MQTT :
  - Configurer le MQTTS.
  - Implémenter les certificats.
  - Implémenter l'authentification par mot de passe.
- Utiliser un certificat pour authentifier le client.

GÉNÉRER DES CERTIFICATS SSL



1 : `openssl req -new -x509 -days 1826 -extensions v3_ca -keyout ca.key -out ca.crt`

2 : `openssl genrsa -out server.key 2048`

3 : `openssl req -out server.csr -key server.key -new`

4 : `openssl x509 -req -in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out server.crt -days 360`