# Arrachage et dépendance temporelle

Hélène Le Bouder, Ronan Lashermes et **Fabien Autrel**

2023

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Inria

# Attaques

- Attaque par force brute.
- Attaque par arrachage.
- Attaque par dépendance temporelle.

# Objectifs pédagogiques

- Implémenter un compteur d'essais.
- Implémenter en temps constant.

# Attaque par force brute

Terminal

```
pin_verif/> python3 ../L2/B-brute-force.py bin/
```

# Compteur d'essais

- Mémoire non volatile

nvm.h

```
1 #define NVM_ADDRESS 0x50000000
2
3 void      store_counter(uint32_t counter_value);
4 uint32_t    load_counter();
```

- Limiter à 3 essais la vérification de code PIN.
- À vous de jouer !

# Attaque par arrachage

- Interruption du programme.
- Arrachage d'une carte à puce de son lecteur.
- `ARRACHER_ICI`

```
1 if (compare_arrays(candidate_pin,
  ↪  secret_pin, 4) == true ) {
2     //...
3 }
4 else {
5     ARRACHER_ICI
6     //...
7 }
```

**Terminal**

```
pin_verif/> make
pin_verif/> python3 ../L2/C-arrachage-bruteforce.py bin/
pin_verif/> python3 ../L2/D-arrachage-temporelle.py bin/
```

# Implémenter en temps constant

- `D-arrachage-temporelle.py` est plus rapide.
- Fuite d'information dans `compare_arrays`.
- Code PIN trouvé en 40 essais.

# À vous de jouer

Terminal

```
pin_verif/> python3 ../L2/E-temporelle.py bin/
```

- Modifier pour que la comparaison soit faite en temps constant.

# Au revoir

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Inria

*illustrations : Le Mooncat*