Première implémentation d'un code PIN naïf

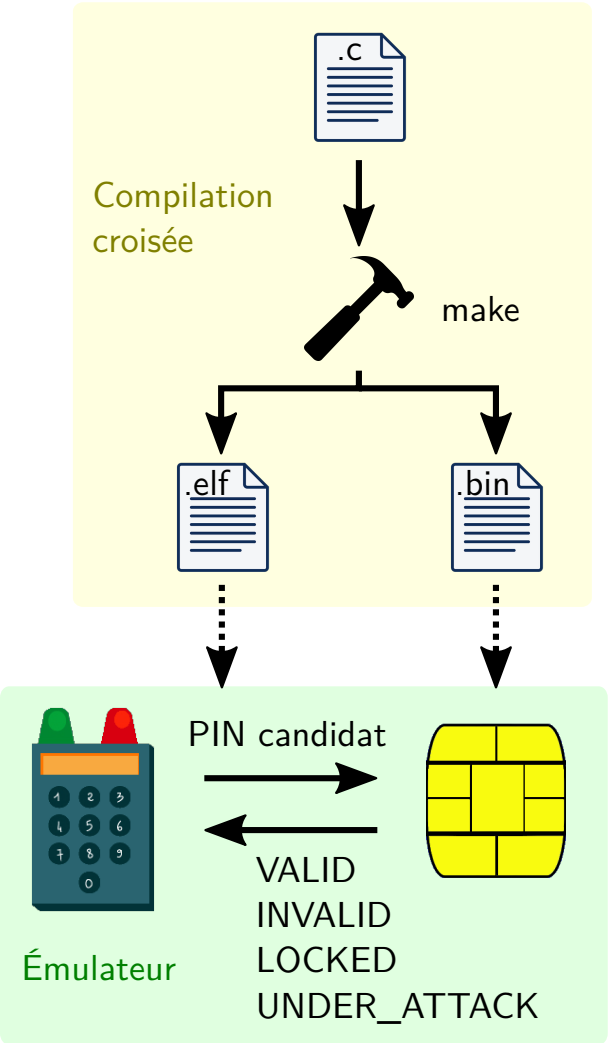
Hélène Le Bouder, Ronan Lashermes et **Fabien Autrel**

2023

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

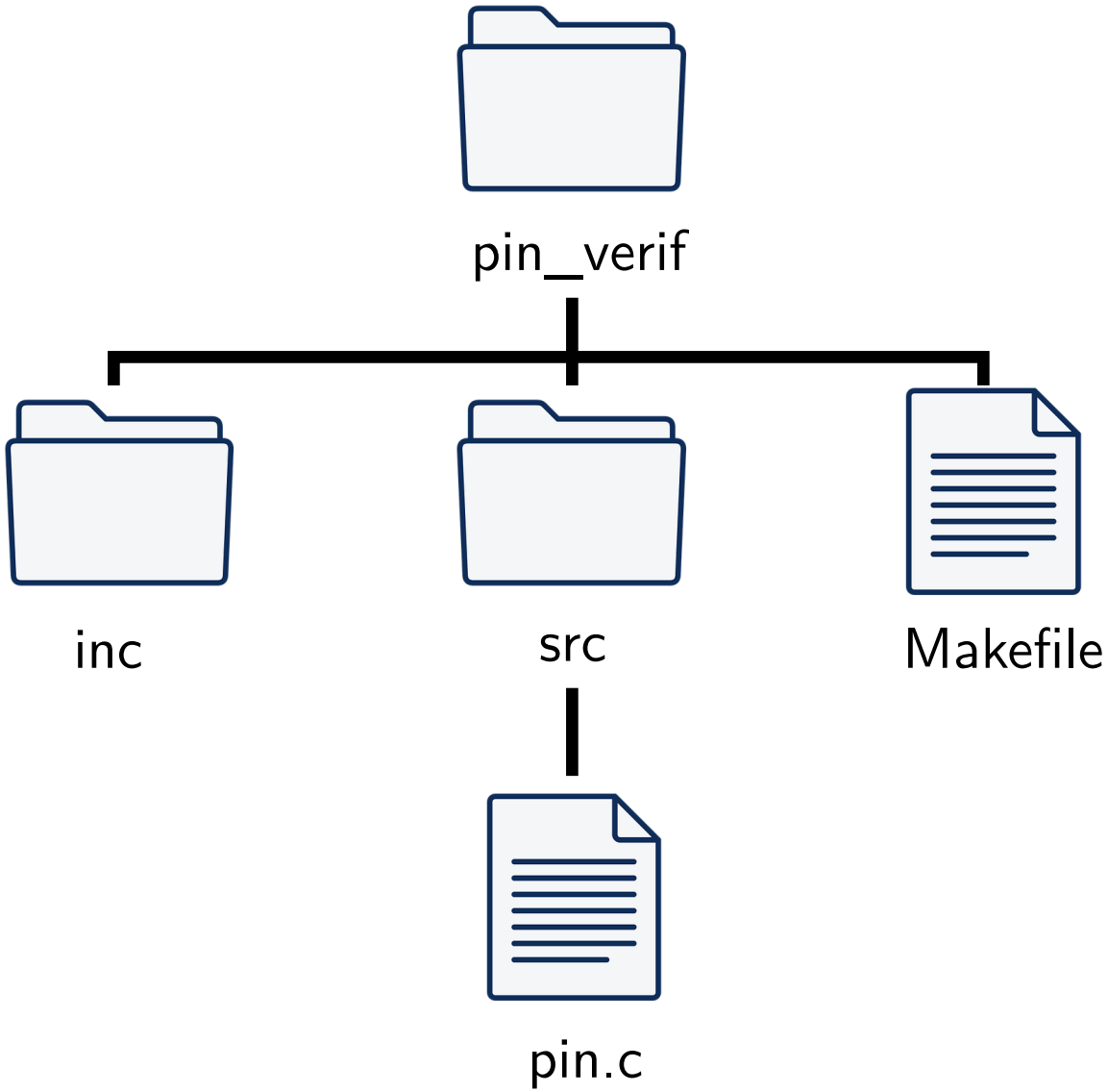
Inria

Objectifs pédagogiques



1. Création de l'application cible de **vérification de code PIN**, en C, à l'aide d'outils de compilation croisée.
2. Exécution de la machine, via un émulateur piloté en Python.

Squelette de l'application



Exercice

pin.c

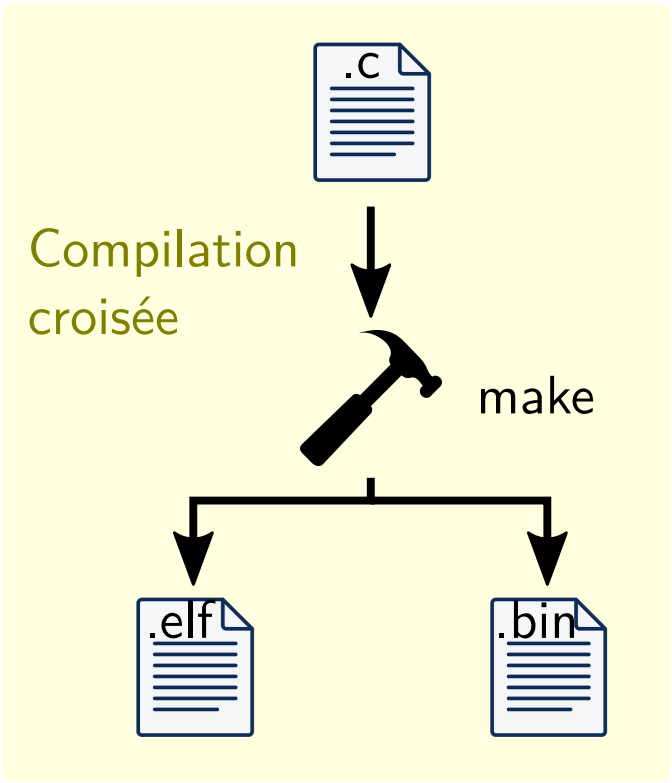
```c
#include "pin.h"
#include "num.h"

//ceci est le code PIN secret
const uint8_t secret_pin[4] = {3, 1, 4, 1};

bool compare_arrays(const uint8_t* a, const uint8_t* b,
    size_t len) {
    return false; // Remplacer par votre propre fonction
}

int verify_pin(const uint8_t* candidate_pin, size_t len) {
    return INVALID; // Remplacer par votre propre fonction
    // Doit retourner VALID, INVALID, LOCKED ou
    UNDER_ATTACK
}
```

Compilation

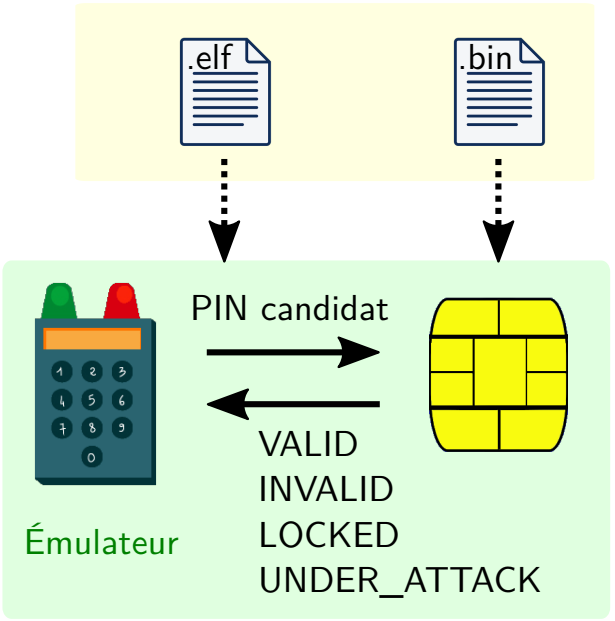


Terminal

```
pin_verif/> make
```

À vous de jouer !

Test de l'implémentation



Dans le dossier `pin_verif`, exécuter la commande :

Terminal

```
pin_verif/> python3 ../L1/A-authentification.py bin/
```

Test de l'implémentation

- Avec un code PIN Faux

Terminal

```
> Veuillez entrer votre PIN: 1111
! PIN incorrect ! Essais restants: 3
> Veuillez entrer votre PIN:
```

- Avec un code PIN Vrai

Terminal

```
> Veuillez entrer votre PIN: 3141
*** PIN accepté *** Essais restants: 3
```

Attaque par force brute

- Test de toutes les combinaisons.

Au revoir

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

Inria

illustrations : Le Mooncat