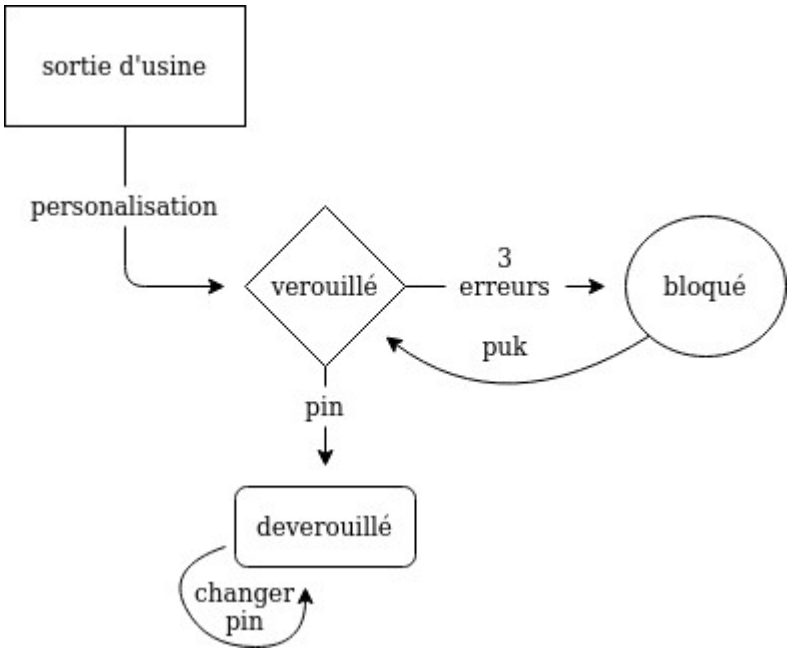
PIN/PUK

1.cycle de vie:



2.commandes:

Nom	Classe	Instruction	P1	P2	P3	argument
Intro PIN	A0	20	0	0	8	Pin
Modif Pin	A0	24	0	0	16	Pin, new Pin
Intro PUK	A0	2C	0	0	16	Puk, new Pin
Perso PUK	A0	40	0	0	8	Puk

3.code erreur:

SW1	SW2	Erreur
90	FF	Mauvais argument
6C	xx	p3 faux, xx = bon p3
6D	00	Ens inconnu
6E	00	Ins inconnu
66	00	Impossible, mauvais etats
90	0x	Echec, x tentatives restantes

4.implementaion:

Nom	Taille	Mémoire
Pin	8 bytes	EEPROM
Puk	8 bytes	EEPROM
Nombre d'essai restant	1 bytes	EEPROM
Etats	1 bytes	RAM

Les 4 etats possibles sont déduits à partir du nombre d'essais restants ainsi:

Nom d'essai restant	Etats
0	Vierge
1,2,3	Verouillé
255	bloqué

La personnalisation n'est possible que si la carte est vierge, le code puk ne peut plus être modifié par la suite.

On arrive ensuite à l'état verouillé, une seule commande permise, intro_pin.

Si on débloque la carte en moins de 3 tentatives on peut modifier le code pin grâce à change_pin, sinon il est nécessaire d'avoir accès au code puk pour pouvoir débloquer la carte puce.

Le code pin initial est 8*[0]

Implementation en detail dans le fichier pinkpuk.c

5.securisation:

Brute force

Dans l'état actuel, la carte est brute-forcable.

Il suffirait de mettre un nombre d'essais maximum restants pour le code puk egalement.

Quand la variable (EEPROM) nombre d'essais puk est égal a 0, la carte est dans l'états «morte».

Attaques temporelles et de consommation

La base de ces attaques découle de la corrélation entre les clefs et le temps d'execution/consommation.

L'utilisation d'un chiffrement par bloc comme TEA est essentielle pour se prémunir de ces attaque là.

Il est également important que la comparaison entre les données et la clef soit faite avec un XOR.

6.Test:

Voir pinpuk.script et imagine en dessous

Commande	Etats	Pin	Puk	Essai restant	commentaire
reader 0	Vierge	8*0	8*0	0	
perso 8*F	Verouillé	8*0	8*F	3	
Pin 8*A	Verouillé	8*0	8*F	2 = SW2	Mauvais pin
Pin 8*0	Deverouillé	8*0	8*F	3	
Modif pin ff+6*0, 8*A	Deverouillé	8*0	8*F	3	Mauvais pin
Modif pin 8*0,8*A	Deverouillé	8*A	8*F	3	Modif pin
reset	Verouillé	8*A	8*F	3	
Modif pin	Verouillé	8*A	8*F	3	Mauvais etats
Pin 8*A	Deverouillé	8*A	8*F	3	
Reset	Verouillé	8*A	8*F	3	
Pin 8*1	Verouillé	8*A	8*F	2	Mauvais pin
Pin 8*1	Verouillé	8*A	8*F	1	Mauvais pin
Pin 8*1	Bloqué	8*A	8*F	FF	Mauvais pin
Modif pin 8*A,8*0	Bloqué	8*A	8*F	FF	Mauvais etats
PUK 8*A ,8*0	Bloqué	8*A	8*F	FF	Mauvais puk
PUK 8*F, 8*0	Verouillé	6*0+AA	8*F	3	
Pin 8*A	Verouillé	6*0+AA	8*F	2	Mauvais pin
Modif pin	Verouillé	6*0+AA	8*F	2	Mauvais etats
Pin 8*0	Verouillé	6*0+AA	8*F	1	Mauvais pin
Pin 6*0+AA	Deverouillé	6*0+AA	8*F	3	

```
  0.000 < a0 40 00 00 08 ff ff ff ff ff ff ff ff              ◆@...◆◆◆◆◆◆◆◆
  0.152 > 90 00                                               ◆.
exécution normale
* a0 20 00 00 08 aa aa aa aa aa aa aa aa

  0.152 < a0 20 00 00 08 aa aa aa aa aa aa aa aa              ◆ ...◆◆◆◆◆◆◆◆
  0.176 > 90 02                                               ◆.
erreur
* a0 20 00 00 08 00 00 00 00 00 00 00 00

  0.176 < a0 20 00 00 08 00 00 00 00 00 00 00 00              ◆ ............
  0.200 > 90 00                                               ◆.
exécution normale
* a0 24 00 00 10 ff 00 00 00 00 00 00 00 aa aa aa aa aa aa aa aa

  0.200 < a0 24 00 00 10 ff 00 00 00 00 00 00 00 aa aa aa     ◆$...◆.......◆◆◆
  0.200 < aa aa aa aa aa                                      ◆◆◆◆◆
  0.232 > 90 ff                                               ◆◆
erreur
* a0 24 00 00 10 00 00 00 00 00 00 00 00 aa aa aa aa aa aa aa aa

  0.232 < a0 24 00 00 10 00 00 00 00 00 00 00 00 aa aa aa     ◆$...........◆◆◆
  0.232 < aa aa aa aa aa                                      ◆◆◆◆◆
  0.288 > 90 00                                               ◆.
exécution normale
* reset

  0.000 > 3b 07 70 69 6e 5f 70 75 6b                          ;.pin_puk
* a0 24 00 00 10 00 00 00 00 00 00 00 00 aa aa aa aa aa aa aa aa

  0.000 < a0 24 00 00 10 00 00 00 00 00 00 00 00 aa aa aa     ◆$...........◆◆◆
  0.000 < aa aa aa aa aa                                      ◆◆◆◆◆
  0.008 > 66 00                                               f.
problème de sécurité
* a0 20 00 00 08 aa aa aa aa aa aa aa aa

  0.008 < a0 20 00 00 08 aa aa aa aa aa aa aa aa              ◆ ...◆◆◆◆◆◆◆◆
  0.032 > 90 00                                               ◆.
exécution normale
* reset

  0.000 > 3b 07 70 69 6e 5f 70 75 6b                          ;.pin_puk
* a0 20 00 00 08 11 11 11 11 11 11 11 11

  0.000 < a0 20 00 00 08 11 11 11 11 11 11 11 11              ◆ ............
  0.024 > 90 02                                               ◆.
erreur
* a0 20 00 00 08 11 11 11 11 11 11 11 11

  0.024 < a0 20 00 00 08 11 11 11 11 11 11 11 11              ◆ ............
  0.044 > 90 01                                               ◆.
erreur
* a0 20 00 00 08 11 11 11 11 11 11 11 11

  0.044 < a0 20 00 00 08 11 11 11 11 11 11 11 11              ◆ ............
  0.068 > 90 ff                                               ◆◆
erreur
* a0 24 00 00 10  aa aa aa aa aa aa aa aa 00 00 00 00 00 00 00 00

  0.068 < a0 24 00 00 10 aa aa aa aa aa aa aa aa 00 00 00     ◆$...◆◆◆◆◆◆◆◆...
  0.068 < 00 00 00 00 00                                      .....
  0.076 > 66 00                                               f.
problème de sécurité
* a0 2c 00 00 10 ff ff ff ff ff ff ff ff 00 00 00 00 00 00 00 aa

  0.076 < a0 2c 00 00 10 ff ff ff ff ff ff ff ff 00 00 00     ◆,...◆◆◆◆◆◆◆◆...
  0.076 < 00 00 00 00 aa                                      ....◆
  0.136 > 90 00                                               ◆.
exécution normale
* a0 20 00 00 08 aa aa aa aa aa aa aa aa

  0.136 < a0 20 00 00 08 aa aa aa aa aa aa aa aa              ◆ ...◆◆◆◆◆◆◆◆
  0.160 > 90 02                                               ◆.
erreur
* a0 24 00 00 10 aa aa aa aa aa aa aa aa 00 00 00 00 00 00 00 00

  0.160 < a0 24 00 00 10 aa aa aa aa aa aa aa aa 00 00 00     ◆$...◆◆◆◆◆◆◆◆...
  0.160 < 00 00 00 00 00                                      .....
  0.172 > 66 00                                               f.
problème de sécurité
* a0 20 00 00 08 00 00 00 00 00 00 00 00
* a0 20 00 00 08 00 00 00 00 00 00 00 00
1156.968 < a0 20 00 00 08 00 00 00 00 00 00 00 00              ◆ ............
1156.988 > 90 01                                               ◆.
erreur
* a0 20 00 00 08 00 00 00 00 00 00 00 aa
* a0 20 00 00 08 00 00 00 00 00 00 00 aa
1257.595 < a0 20 00 00 08 00 00 00 00 00 00 00 aa              ◆ ...........◆
1257.615 > 90 00                                               ◆.
exécution normale
```