La Carte à Microprocesseur

Un système embarqué en plein essor

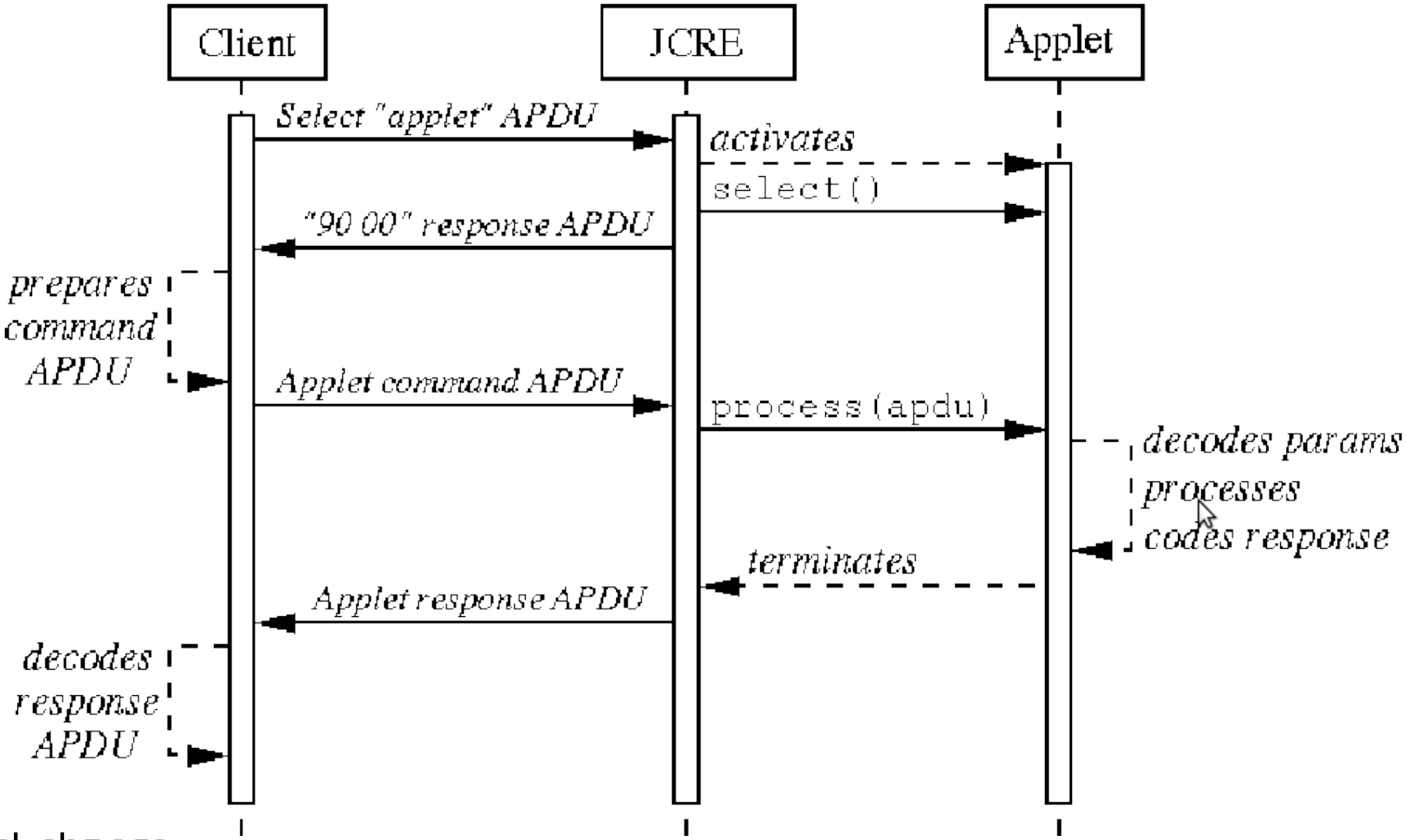
Tegawendé F. Bissyandé
tegawende.bissyande@fasolabs.org

Cours préparé pour
L'Institut Supérieur de Technologie (IST Burkina)

April 14, 2015

FasoLabs.org

Programming....



FasoLabs.org

Programming....

- Un simple Compteur
  - Carte de fidélité, Porte Monnaie Electronique, ...
- APDUs traités par l'applet :
  - `int lire()`
    - Commande : `AA 01 XX XX 00 04`
    - Réponse : `RV3 RV2 RV1 RV0 90 00`
  - `int incrementer( int )`
    - Commande : `AA 02 XX XX 04 AM3 AM2 AM1 AM0 04`
    - Réponse : `RV3 RV2 RV1 RV0 90 00`
  - `int decrementer( int )`
    - Commande : `AA 03 XX XX 04 AM3 AM2 AM1 AM0 04`
    - Réponse : `RV3 RV2 RV1 RV0 90 00`

FasoLabs.org

Programming....

```
package org.carte.compteur ;

import javacard.framework.* ;

public class Compteur extends Applet {
  private int valeur;

  public Compteur() { valeur = 0; register(); }
  public static void install( APDU apdu ) { new Compteur(); }

  public void process( APDU apdu ) {
    byte[] buffer = apdu.getBuffer();
    if ( buffer[ISO.OFFSET_CLA] != 0xAA )
      ISOException.throwIt(ISO.SW_CLA_NOT_SUPPORTED);
    switch ( buffer[ISO.OFFSET_INS] ) {
      case 0x01: ... // Opération de lecture
      case 0x02: ... // Opération d'incrémentation
      case 0x03: ... // Opération de décrémentation
      default:
        ISOException.throwIt(ISO.SW_INS_NOT_SUPPORTED);
    }
  }
}
```

FasoLabs.org

Programming....

```
case 0x03: // Opération de décrémentation
{
  // Réception des donnnées
  byte octetsLus = apdu.setIncomingAndReceive();
  if ( octetsLus != 4 )
    ISOException.throwIt(ISO.SW_WRONG_LENGTH);
  int montant = (buffer[ISO.OFFSET_CDATA]<<24) |
    (buffer[ISO.OFFSET_CDATA+1]<<16) |
    (buffer[ISO.OFFSET_CDATA+2]<<8) |
    buffer[ISO.OFFSET_CDATA+3];
  // Traitement
  if ( montant<0 || valeur-montant<0 )
    ISOException.throwIt((short)0x6910);
  valeur = valeur - montant;
  // Envoie de la réponse
  buffer[0] = (byte)(valeur>>24);
  buffer[1] = (byte)(valeur>>16);
  buffer[2] = (byte)(valeur>>8);
  buffer[3] = (byte)(valeur);
  apdu.setOutgoingAndSend((short)0, (short)4);
  return;
}
```


Programming....

```
package com.banque ;

import javacard.framework.*;

public class Pme extends Applet {
  final static byte Pme_CLA        = (byte)0xB0;
  final static byte Crediter_INS   = (byte)0x10;
  final static byte Debiter_INS    = (byte)0x20;
  final static byte Lire_INS       = (byte)0x30;
  final static byte Valider_INS    = (byte)0x40;
  final static byte MaxEssai_PIN   = (byte)0x03;
  final static byte MaxLg_PIN      = (byte)0x08;
  final static short BalanceNegative_SW = (short)0x6910;

  OwnerPin pin;
  byte balance;
  byte[] buffer;

  private Pme() {
    pin = new OwnerPIN(MaxEssai_PIN, MaxLg_PIN);
    balance = 0;
    register() ;
  }
```


Programming....

```
public static void install(byte[] bArray,
                           short bOffset, byte bLength) {
   Pme p=new Pme();
   pin.updateAndUnblock(bArray, bOffset, bLength);
}

public boolean select() { pin.reset(); return true; }

public void process( APDU apdu ) {
   buffer = apdu.getBuffer();
   if ( buffer[ISO.OFFSET_CLA] != Pme_CLA )
     ISOException.throwIt(ISO.SW_CLA_NOT_SUPPORTED);
   switch ( buffer[ISO.OFFSET_INS] ) {
     case Crediter_INS : crediter(apdu); return;
     case Debiter_INS : debiter(apdu); return;
     case Lire_INS : lire(apdu); return;
     case Valider_INS : valider(apdu); return;
     default:
       ISOEXception.throwIt(ISO.SW_INS_NOT_SUPPORTED);
   }
```

FasoLabs.org

Programming....

```
// Réception de données
private void crediter( APDU apdu ) {
  if ( !pin.isValidated() )
    ISOException.throwIt(ISO.SW_PIN_RIQUIRED);
  byte octetsLus = apdu.setIncomingAndReceive();
  if ( octetsLus != 1 )
    ISOException.throwIt(ISO.SW_WRONG_LENGTH);
  balance = (byte)(balance + buffer[ISO.OFFSET_CDATA]);
}

// Réception de données
private void debiter( APDU apdu ) {
  if ( !pin.isValidated() )
    ISOException.throwIt(ISO.SW_PIN_RIQUIRED);
  byte octetsLus = apdu.setI..comingAndReceive();
  if ( octetsLus != 1 )
    ISOException.throwIt(ISO.SW_WRONG_LENGTH);
  if ( (balance - buffer[ISO.OFFSET_CDATA]) < 0 )
    ISOException.throwIt(BalanceNegative_SW);
  balance = (byte)(balance - buffer[ISO.OFFSET_CDATA]);
}
```

Programming....

```
    // Émission de données
    private void lire( APDU apdu ) {
      if ( !pin.isValidated() )
        ISOException.throwIt(ISO.SW_PIN_RIQUIRED);
      apdu.setOutgoing();
      apdu.setOutgoingLength((byte)1);
      buffer[0] = balance;
      apdu.sendBytes((short)0, (short)1) ;
    }

    // Manipulation du code secret
    private void valider( APDU apdu ) {
      byte octetsLus = apdu.setIncomingAndReceive();
      pin.check(buffer, ISO.OFFSET_CDATA, octetsLus);
    }

}
```

FasoLabs.org

Programming....

```
private DESKey myDESKey;

public static void install(byte[] bArray,
                           short bOffset, byte bLength) {
    new Encryption ();
    pin.updateAndUnblock(bArray, bOffset, bLength);
 }

  public boolean select() { pin.reset(); return true; }

  public void process( APDU apdu ) {
    buffer = apdu.getBuffer();
    if ( buffer[ISO.OFFSET_CLA] != 0x00 )
      ISOException.throwIt(ISO.SW_CLA_NOT_SUPPORTED);
    switch ( buffer[ISO.OFFSET_INS] ) {
      case ENCRYPT_INS : encrypt(apdu); return;
      case PINCHECK_INS : pinCheck(apdu); return;
     default:
       ISOEXception.throwIt(ISO.SW_INS_NOT_SUPPORTED);
    }
```

FasoLabs.org

Programming....

- Classe dérivant de `javacard.framework.Applet`
- Une applet carte
  est un programme serveur de la Java Card
  - APDU de sélection depuis le terminal (select)
  - Sélection par AID (chaque applet doit avoir un AID unique)
    - AID
      - 5 octets identifiant le propriétaire
      - 0-11 octets dépendant du propriétaire

FasoLabs.org

Programming....

- Classe dérivant de `javacard.framework.Applet`
- Une applet carte
  est un programme serveur de la Java Card
  - APDU de sélection depuis le terminal (select)
  - Sélection par AID (chaque applet doit avoir un AID unique)
    - AID
      - 5 octets identifiant le propriétaire
      - 0-11 octets dépendant du propriétaire

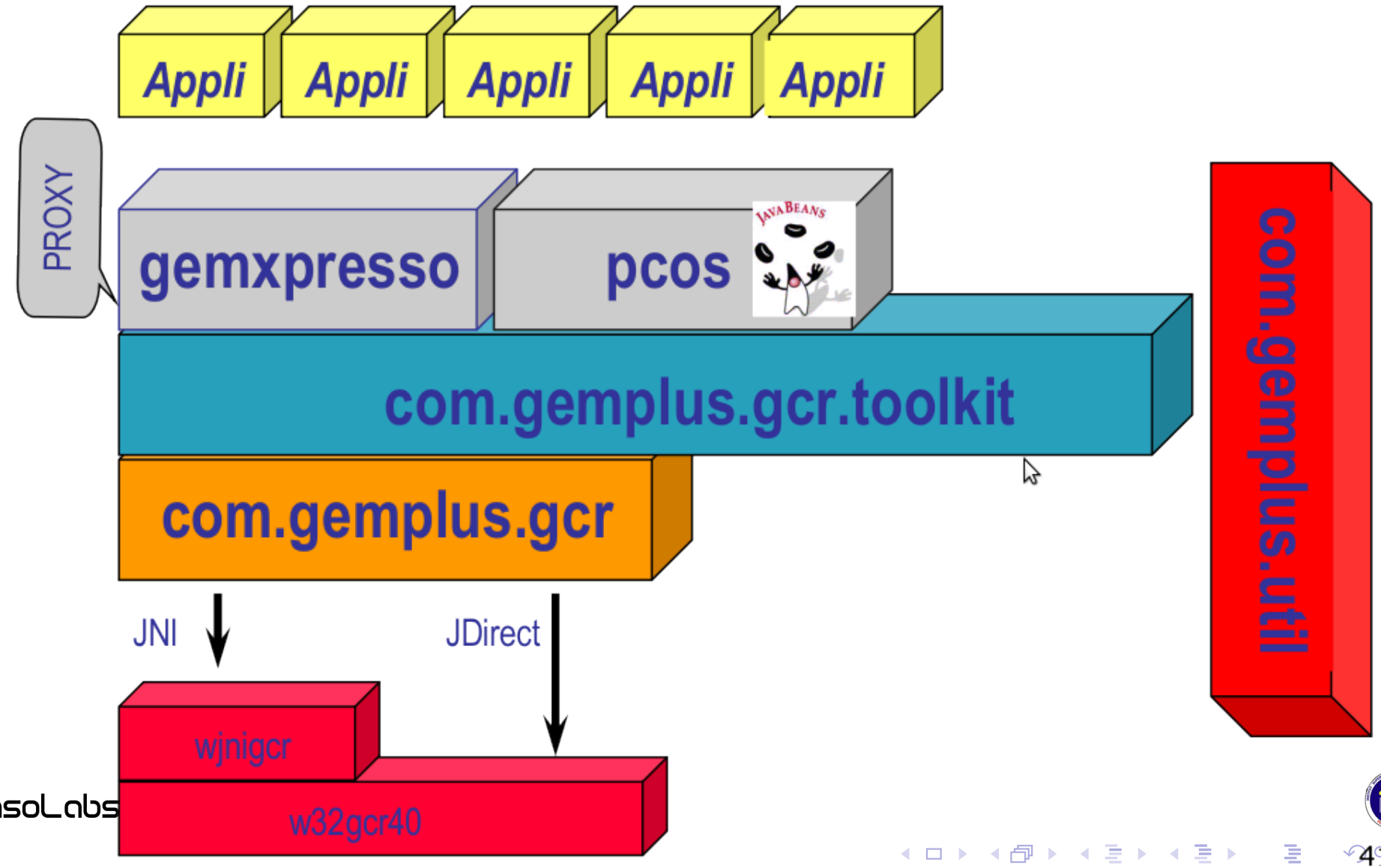
FasoLabs.org

Programming....

- Cycle de vie : Méthodes appelées par JCRE
  - `static void install(bArray,bOffset,bLength)`
    - Crée une instance de la classe avec les paramètres passés dans bArray
    - Puis l'enregistre (register()) auprès du JCRE
  - `boolean select()`
    - Appelé à la sélection
    - peut retourner false si l'initialisation est incomplète (liaison impossible vers des objets partagés, ...)
  - `void deselect()`
    - Appelé à la désélection
  - `void process(APDU apdu)`
    - Méthodes appelées par JCRE

FasoLabs.org

Programming....



Programming....

```
package compteur.client.gemplus.fr ;

import com.gemplus.gcr.* ;

/* Application terminal */
Ifd lecteur =
  new IfdSerial(IFDTYPE.GCR410, SERIALPORT.G_COM1, 9600);
Icc carte = new Icc() ;
try {
  // Connexion à la carte via lecteur GCR410
  short canal = reader.openChannel();
  SessionParameters atr = carte.openSession(canal);
  // Échange d'APDUs (APDU de selection de l'applet Compteur)
  ApduCommand commande = new ApduCommand( /* paramètres */ );
  ApduResponse reponse = carte.exchangeApdu(commande);
  /* etc */
  // Fin de la connexion
  carte.closeSession();
  lecteur.closeChannel(canal);
} catch ( GcrException e ) {
  // Récupération de l'erreur
  System.out.println(``Problème : '' + e.getMessage());
}
```

FasoLabs.org

Programming....

```
// Commande = AA 03 XX XX 04 AM3 AM2 AM1 AM0 04
// Reponse = RV3 RV2 RV1 RV0 90 00 ou 69 10
int montant = System.in.read();
byte[] montantApdu = new byte[4];
montantApdu[0] = (byte)(montant >> 24);
montantApdu[1] = (byte)(montant >> 16);
montantApdu[2] = (byte)(montant >> 8);
montantApdu[3] = (byte)(montant);
ApduCommand commande =
  new ApduCommand(0xAA, 0x03, 0, 0, montantApdu, 4);
ApduResponse reponse = carte.exchangeApdu(commande);
if ( reponse.getShortStatus() == 0x9000 ) {
  byte[] apduValeur = reponse.getDataOut();
  int valeur = (apduValeur[0]<<24) |
    (apduValeur[1] <<16) | (apduValeur[2]<<8) |
    apduValeur[3];
  System.out.println(``Valeur compteur : `` + valeur);
} else {
  if ( reponse.getShortStatus() == 0x6910 )
  { /* Traite l'erreur «Valeur négative» */ }
}
```

FasoLabs.org

Helloworld – basic

pme.java = (/tmp) - VIM 88x47

```
11 >-------private byte counter;
12 >-------/** The String "Hello" */
13 >-------private final static byte[] hello =
14 >-------{ 0x48, 0x65, 0x6c, 0x6c, 0x6f } ;
15
16 >-------/**
17 >------- * Default constructor
18 >------- * Only this class's install method should create the applet object.
19 >------- *  @param baBuffer the array constaining installation parameters
20 >------- *  @param sOffset the starting offset in baBuffer
21 >------- *  @param bLength the length in bytes of the data parameter in baBuffer
22 >------- */
23 >-------protected HelloWorld(byte[] baBuffer, short sOffset, byte bLength) {
24 >------->-------register(baBuffer, (short) (sOffset + 1), (byte) baBuffer[sOffset])
25 >------->-------
26 >------->-------counter = (byte)0x0;
27 >-------}
28
29 >-------/**
30 >------- * Method installing the applet.
31 >------- *  @param baBuffer the array constaining installation parameters
32 >------- *  @param sOffset the starting offset in baBuffer
33 >------- *  @param bLength the length in bytes of the data parameter in baBuffer
34 >------- */
35 >-------public static void install(byte[] baBuffer, short sOffset, byte bLength) {
36 >------->-------// applet  instance creation-
37 >------->-------new HelloWorld(baBuffer, sOffset, bLength);
38 >-------}
39
40 >-------/**
41 >------- * Select method returns true if applet selection is supported.
42 >------- *  @return boolean Should always be true.
43 >------- */
44 >-------public boolean select() {
45
46 >------->-------return true;
47 >-------}
48
49 >-------/**
50 >------- * Deselect method called by the system in the deselection process.
51 >------- */
52 >-------public void deselect() {
53
54 >-------}
55
/tmp/pme.java[RO][108][dos][java][41%][0045,0001]
```

pme.java = (/tmp) - VIM 88x47

```
62 >-------public void process(APDU apdu) {
63
64 >------->-------// Good practice: Return 9000 on SELECT
65 >------->-------if (selectingApplet()) {
66 >------->------->-------return;
67 >------->-------}
68
69 >------->-------byte[] buf = apdu.getBuffer();
70
71 >------->-------switch (buf[ISO7816.OFFSET_INS]) {
72 >------->-------case (byte) 0x40:
73 >------->------->-------Util.arrayCopy(hello,(byte)0,buf,ISO7816.OFFSET_CDATA,(byte
74 >------->------->-------apdu.setOutgoingAndSend(ISO7816.OFFSET_CDATA,(byte)5);
75 >------->------->-------break;
76 >------->-------case (byte)0x41:
77 >------->------->-------read(apdu, buf);
78 >------->------->-------break;
79 >------->-------case (byte)0x42:
80 >------->------->-------increment(buf[ISO7816.OFFSET_P1]);
81 >------->------->-------break;
82 >------->-------case (byte)0x43:
83 >------->------->-------decrement(buf[ISO7816.OFFSET_P1]);
84 >------->------->-------break;
85 >------->-------default:
86 >------->------->-------// good practice: If you don't know the INStruction, say so
87 >------->------->-------ISOException.throwIt(ISO7816.SW_INS_NOT_SUPPORTED);
88 >------->-------}
89 >-------}
90 >-------
91 >-------public void increment(byte value){
92 >------->-------counter += value;
93 >-------}
94 >-------
95 >-------public void decrement(byte value){
96 >------->-------if(counter < value)
97 >------->------->-------ISOException.throwIt(ISO7816.SW_INCORRECT_P1P2);
98 >------->-------counter -= value;
99 >-------}
100 >-------
101 >-------public void read(APDU apdu, byte[] apduBuffer){
102 >------->-------apduBuffer[0] = counter;
103 >------->-------apdu.setOutgoingAndSend((short)0, (short)1);
104 >-------}
105 >-------
106 >-------
/tmp/pme.java[RO][108][dos][java][88%][0096,0001]
```

Attaques... à la limite de la légalité!

Attaques purement logicielles

Elles nécéssitent :

- un analyseur de dialogue
- un logiciel de capture de dialogue (SeriWatcher, AspyCom, etc.)
- fiche technique de la carte (pas indispensable)
- des compétences en cartes à puce et en cryptographie
- du temps!!! beaucoup de temps!!!

FasoLabs.org

Attaques... à la limite de la légalité!

Attaques matérielles destructrices

Il faut du matériel :

- récuperer la puce sur la carte (dissoudre la résine sans abimer la puce)
- examiner la puce au microscope électronique (pour tenter de lire la mémoire morte du programme)
- Seul l'encarteur peut protéger la carte en utilisant de la résine difficile à dissoudre

FasoLabs.org

Attaques... à la limite de la légalité!

Attaques matérielles non destructrices

Encore plus complexes, il faut :

- dissoudre la résine sans altérer la puce car elle doit pouvoir continuer à fonctionner
- Placer des sondes en des points stratégiques de la puce pour analyserr
- reconstituer le comportement du microcontrolleur (reverse engineering)
- Encore une fois c'est à l'encarteur de bien choisir la résine

FasoLabs.org

Attaques... à la limite de la légalité!

Attaques externes de types SPA, DPA, EMA

Presque à la portée d'un hacker :

- SPA : Simple Power Analysis
- DPA : DIfferential Power Analysis
- EMA : Electro Magnetic Analysis
- Pour extraire des clés RSA à partir des signaux de transmissions