février 2011

Cartes à puce

Écrit

Exercice 1 : PIN (X points – Damien Sauveron)

Documents autorisés : Le cours et les TDs

1. À quelles attaques l'algorithme suivant de vérification de PIN est-il sensible et pourquoi ?

```
for ( i = 0 ; i <= 7; i++)
    if ( pinCarte [ i ] != pinPresente [ i ] )
        return false ;
    return true;
```

2. Implémenter une version sécurisée de cet algorithme.

Exercice 2 : Java Card (X points – Damien Sauveron) – Jouons un peu !

Documents autorisés : Le cours et les TDs

Le but de cet exercice est d'implanter une applet Java Card qui permet de jouer à un petit jeu entre deux amis. L'un d'eux (ami 1) va utiliser la commande APDU Choisir_Secret pour entrer un nombre secret et l'autre (ami 2) utilisera la commande APDU Deviner_Secret pour essayer de découvrir ce nombre secret. Il aura un maximum de 5 essais pour découvrir le fameux nombre secret. L'applet gérera le compteur d'essais et pour chaque proposition lui indiquera, le nombre d'essais restants et si le nombre secret est supérieur ou inférieur au nombre qu'il a proposé au travers d'un code de retour. Si l'ami 2 échoue dans sa quête (i.e. compteur d'essais à 0), l'applet le lui indiquera au travers d'un code de retour particulier. Les spécifications des commandes et réponses sont données ci-dessous. Le nombre secret sera compris entre 0 et 127.

- Elle acceptera les commandes suivantes :

Commande «Choisir_Secret»						
Commande APDU						
CLA	**INS**	**P1**	**P2**	**Lc**	**Data field**	**Le**
0x80	0x10	secret	N/A	N/A	N/A	0x00
With secret in [0,127]						
Réponse APDU						
Optional data	**Status word**	**Meaning of status word**				
	0x9000	Successful processing				
	0x9010	One argument is invalid (i.e. out of the given range)				

Commande «Deviner_Secret»

février 2011

Commande «Deviner_Secret»						
Commande APDU						
CLA	**INS**	**P1**	**P2**	**Lc**	**Data field**	**Le**
0x80	0x20	v_tried	N/1	N/A	N/A	0x01
With v_tried in [0,127]						
Réponse APDU						
Optional data	**Status word**	**Meaning of status word**				
Remaining tries	0x9000	Successful processing (YOU WIN)				
	0x9010	One argument is invalid (i.e. out of the given range)				
Remaining tries	0x9022	secret > v_tried and number of tries ≠ 0				
Remaining tries	0x9024	secret < v_tried and number of tries ≠ 0				
	0x9030	YOU LOSE (number of tries = 0 and secret ≠ v_tried)				

1. Le code de départ de l'applet est le suivant :

```
package fr.unilim.msi.calc;
import javacard.framework.*;
public class CestPlusCestMoins extends Applet
{
    public static void install(byte[] bArray, short bOffset, byte bLength) throws ISOException
    {
        new CestPlusCestMoins().register();
    }

    public void process(APDU apdu) throws ISOException
    {
        // Insérer ici le code métier
    }
}
```

2. Écrire le code minimal à placer dans la méthode `process(APDU apdu)` pour traiter les commandes APDU spécifiées ci-dessus et pour renvoyer les codes de retour spécifiés après traitement.
3. Écrire les fonctions réalisant les traitements demandés par les commandes APDUs donnés ci-dessus.
4. Modifier le code de l'applet que vous venez d'écrire pour qu'elle enregistre les différentes propositions faites par l'ami 2 lorsqu'il utilise la commande APDU Deviner_Secret (il y en a donc au maximum 5).
   Ajouter également la réinitialisation dans cette « zone » mémoire des propositions avec des « -

février 2011

1 » lorsque l'ami 1 appelle la commande APDU Choisir_Secret pour fixer un nouveau secret. Enfin ajouter le support de la commande APDU Voir_Propositions_Saisies qui permet de récupérer à tout moment les différentes valeurs présentées par l'ami 2 (i.e. les valeurs qui dans la « zone » sont différentes de « -1 »).

Commande «Voir_Propositions_Saisies»						
Commande APDU						
CLA	**INS**	**P1**	**P2**	**Lc**	**Data field**	**Le**
0x80	0x30	N/A	N/A	N/A	N/A	0x05
Réponse APDU						
Optional data		**Status word**		**Meaning of status word**		
The different values tried (between 0 and 5)		0x9000		Successful processing		