Partie Irek Tobor

Durée : 1h30, 10 pts

Exercice 1, Questions de cours

1. On peut dire qu'une carte à puce est un ordinateur avec son jeu de composants (CPU, mémoire, circuits d'entrée/sortie, unités de calculs). Pourquoi dans les PC "de bureau" on a en général 2 types de mémoire (RAM et ROM) et dans les cartes à puces (et, en général, dans le domaine de l'embarqué) 3 (RAM, EEPROM et ROM) ? A quoi sert chacun de ces types de mémoire de la carte à puce et quel type de données est y stocké.
2. La communication entre un terminal et une carte à puce est faite en mode "maître-esclave" : c'est le terminal qui envoie la commande (APDU) et la carte qui doit répondre (R-APDU). Sauf un cas. Lequel ?
3. Pourquoi les cartes bancaires qui peuvent communiquer en mode "contact" et "sans contact" on ne peut pas toujours lire (dans le sens "c'est la spécification qui l'interdit") les mêmes informations (fichiers) dans ces deux modes.

Exercice 2, Authentification

L'authentification est la procédure qui consiste, pour un système informatique, à vérifier l'identité d'une autre entité afin d'autoriser l'accès de cette entité à des ressources. L'authentification permet donc de valider l'authenticité de l'entité en question. Dans le cadre de cartes à puce la procédure d'authentification permet donc, par exemple de vérifier l'authenticité de la carte du point de vue du terminal ou l'inverse.

Plusieurs exemples peuvent se présenter :

1. Commande Internal Authenticate d'un protocole bancaire qui permet au terminal de s'assurer que la carte bancaire est authentique. La carte chiffre les données aléatoires envoyée par le terminal et le terminal vérifie si le chiffrement est correct.
2. Commande External Authenticate d'une autre application (par exemple du type IAS, dont on a pas parlé en cours) qui permet à la carte de s'assurer que le terminal qui lui envoie des commandes est authentique. Cette fois ci c'est le terminal qui doit chiffrer les données aléatoires envoyées par la carte et la carte à sont tour les vérifie.
3. Commande Mutual Authenticate du protocole GlobalPlatform qui est nécessaire avant le chargement d'une application / applet sur la carte. Elle permet de prouver mutuellement à la carte et au terminal que l'"autre" entité connaît bien les même clefs de chiffrement. Elle regroupe les deux cas précédents.
4. Procédure / protocole (pas une commande APDU !) Passive Authentication de passeport électronique. Elle consiste à vérifier la signature de différents fichiers du passeport (les données perso, la photo, etc). Cette signature est elle-même stockée sur la carte et on suppose que le terminal dispose de la clef publique nécessaire pour la vérifier. On suppose aussi que les fichiers de données et les signatures sont lisibles sans aucune procédure préalable, que la clef privée de signature n'est pas sur la carte et ne peut en aucun cas être trouvée et la clef publique est connue par terminal.
5. Procédure / protocole Active Authentication de passeport électronique. En plus des données personnelles la carte stocke la paire clef privée (pas lisible) et la clef publique. Terminal lit la clef publique, envoie des données à la puce qui les signe avec la clef privée et retourne cette signature. Le terminal peut donc la vérifier avec la clef publique qu'il a récupéré précédemment.

Notez que, pour les besoins de cet exercice, les hypothèses et les suppositions sont simplifiées par rapport aux cas réels. Il n'est pas demandé de donner la réponse exacte (un tel nombre d'octets, un tel format, plutôt des idées ou principes).

Questions :

1. Dans le cas 1) un échange (commande APDU et puis sa réponse) est nécessaire. Quelles données sont envoyés et quelles données sont reçues ?
2. Dans les cas 2) deux échanges sont nécessaires (en général les commandes Get Challenge et External Authenticate). D'où vient cette asymétrie avec le cas 1) ?
3. Dans les cas 3) également deux échanges sont nécessaires (commandes Initialize Update et External Authenticate). Quelles données sont échangées, que font la carte et le terminal, pourquoi les deux échanges sont suffisants ?
4. Dans le cas 4), en quoi cette procédure empêche de forger de fausses données ou permet de détecter une modifications de données d'un vrai passeport ? Et malgré tout peut on cloner un autre vrai passeport ?
5. Même question pour le cas 5), cette procédure permet-elle de cloner ou falsifier un autre passeport ? Pourquoi il faut la coupler avec Passive Authentication ?

Exercice 3, Analyse de log

Les lignes qui suivent correspondent aux 6 derniers records du fichier "Log de transactions" d'une carte bancaire EMV. On peut y trouver des transactions faites en France, Allemagne (en €) et Royaume Uni. En Royaume Uni dans certains distributeurs d'argent on peut choisir entre faire une transaction en £ ou bien en € pour éviter les frais de change de la banque émettrice de la carte (et dans ce cas là c'est la banque propriétaire du distributeur qui applique sa commission).

- `02 50 08 29 15 12 07 40 00 00 00 01 50 00 00 09 78`
- `08 26 08 28 15 12 05 40 00 00 00 00 50 00 00 08 26`
- `08 26 08 27 15 12 05 40 00 00 00 01 41 29 00 09 78`
- `02 76 08 26 15 12 01 40 00 00 00 00 17 53 00 09 78`
- `02 76 08 25 15 11 26 40 00 00 00 00 21 08 00 09 78`
- `02 50 08 24 15 11 25 40 00 00 00 00 92 33 00 09 78`

La commande GET DATA avec l'argument 9F 4F ("Log Format") répond :

- `9f 4f 13 9f 1a 02 9f 36 02 9a 03 9f 27 01 9f 02 06 9c 01 5f 2a 02`

Notez que pour répondre correctement vous n'avez pas besoin de connaître la signification exacte de chacun des tags utilisés. Un peu de réflexion et de bon sens sont suffisants.

Questions :

1. Trouver la date et le montant de chaque transaction (deux derniers chiffres sont des centimes)
2. Quel est le code-pays de Royaume Uni et le code-monnaie de £ ?
3. Question bonus. A votre avis que signifie le tag `9f 36` ? Quel est son nom et son sigle ?

Barème :

- Exercice 1 : 1+1+2
- Exercice 2 : 2+2+2+2+2
- Exercice 3 : 3 + 3 + 2
- Somme est ramenée à une note sur 10 (elle peut même atteindre 11/10 avec la question bonus)