Examen de Cartes à Puce 2

Septembre 2015

Durée : 1h30

Les supports de cours, TD et TP de l'UE Cartes à Puce 2
sont les seuls documents autorisés pour la composition de cet examen.

L'usage d'une calculatrice est autorisé.

1) Algorithmes d'exponentiation modulaire

a) Expliquez à quoi sert une fonction d'exponentiation modulaire dans une carte à puce.

b) Parmi les différents algorithmes d'exponentiation modulaire, certains sont dits « réguliers ». Expliquez ce que signifie ce terme, et dites pourquoi il est intéressant de choisir un algorithme régulier.

c) Un développeur de RSA embarqué sur carte à puce sait très bien que l'algorithme *Square and Multiply* binaire classique est vulnérable à une analyse simple du courant qui permet de retrouver successivement les différents bits de l'exposant secret *d* manipulé.
Souhaitant malgré tout utiliser cet algorithme d'exponentiation pour des raisons de performance, il décide de protéger son implémentation en utilisant la panoplie complète des masquages (masquages d'exposant, de message et de module).

• Expliquez en quoi consistent ces différents masquages.

• Donner votre avis argumenté sur la pertinence de cette contre-mesure dans ce cas précis.

d) On suppose un attaquant capable de mesurer uniquement le temps d'exécution lors du calcul d'une signature RSA utilisant l'algorithme S*quare and Multiply* binaire classique.

• Dans le meilleur des cas, quelle type d'information l'attaquant peut-il apprendre au sujet de l'exposant privé ?

2) Analyse de consommation de courant

• Dites ce que vous savez sur la DPA et la CPA. Comparez ces deux techniques d'attaque : similarités/différences, avantages/inconvénients.

3) Analyse différentielle du courant sur l'AES

Voici le pseudo-code simplifié d'une implémentation de l'AES :

```
Input : message M de 16 octets, clé K de 16 octets
Output : chiffré de 16 octets

1. (K_0, K_1, …, K_{10}) = KeySchedule(K)
2. S = M
3. S = AddRoundKey(S, K_0)
4. Si (rand()%2) == 0 alors  // rand()%2 renvoie 0 ou 1 avec proba ½ chacun
       4.1 attendre 10 micro-secondes
5. Pour i de 1 à 9
       5.1 S = SubBytes(S)
       5.2 S = ShiftRows(S)
       5.3 S = MixColumns(S)
       5.4 S = AddRoundKey(S, K_i)
6. S = SubBytes(S)
7. S = ShiftRows(S)
8. S = AddRoundKey(S, K_{10})
9. Retourner S
```

• Expliquez l'effet qu'auront les instructions 4 et 4.1 lorsqu'on réalise une DPA sur les sorties des S-Box du premier tour.

4) Analyse différentielle du courant sur le DES

• Dans une attaque DPA sur le DES, combien l'attaquant doit-il générer de courbes de DPA pour retrouver les valeurs de toutes les sous-clés du premier tour ? Justifiez votre réponse.