# Examen de Cartes à Puce – Sécurité des Implémentations

Février 2020

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.

<u>Remarque</u>: 1 point de la note sera consacré à la présentation et à la justification de vos réponses

## 1) Algorithmes d'exponentiation modulaire réguliers (4 points)

Parmi les différents algorithmes d'exponentiation modulaire, certains sont dits « réguliers ».

- a) Expliquez ce que signifie ce terme, et dites pourquoi il est intéressant de choisir un algorithme régulier.
- b) Dites pourquoi l'algorithme "Always Multiply" basé sur le principe de l'atomicité peut à la fois être considéré comme un algorithme régulier, et comme un algorithme non régulier.
  - c) Quelle(s) contre-mesure(s) permet(tent) d'empêcher l'attaque de Amiel et al. exploitant le poids de Hamming moyen. Expliquez et jusitifiez.

#### 2) Représentation binaire signée (5 points)

- a) Quelle est la représentation binaire signée de type NAF de la valeur hexadécimale 73B ?
- b) Écrivez le pseudo-code du "Square and Multiply" de droite à gauche adapté à une multiplication scalaire et à une représentation signée du scalaire.
  - c) Quelle est la séquence d'additions et de doublements effectués lors de l'exécution de l'algorithme de la question b) lorsque le scalaire a pour valeur celle de la question a) ?
- d) Qu'apprend l'attaquant au sujet du scalaire s'il sait distinguer les additions des doublements? Donnez toutes les valeurs possibles du scalaire (en vue d'une recherche exhaustive) qui résulteraient d'une telle attaque sur la séquence d'opérations identifiée à la question c).

#### 3) Analyse différentielle du courant sur l'AES (2 points)

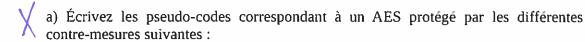
Voici le pseudo-code simplifié d'une implémentation de l'AES :

Input : message M de 16 octets, clé K de 16 octets

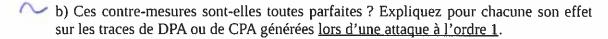
Output : chiffré de 16 octets

- 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 des instructions 4 et 4.1 lorsqu'on réalise une DPA sur les sorties des S-Box du premier tour.

### 4) Contre-mesures pour les chiffrements par blocs (5 points)



- Masquage de données
- Ordre des opérations aléatoire
- Faux calculs en ordre aléatoire



🔪 c) Quel est l'impact de chacune de ces contre-mesures sur le temps d'exécution ?

## **5) Analyse de fautes** (3 points)

- a) Est-il possible de réaliser une analyse différentielle de faute sur le DES lorsque les entrées de l'algorithme ne sont pas connues ? Expliquez.
- b) Est-il possible de réaliser une analyse de fautes par collision sur l'AES lorsque les entrées de l'algorithme sont connues mais pas choisies par l'attaquant ? Expliquez.