Examen de Cartes à Puce 2

Février 2015

Durée : 1h30

Les supports de cours 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) La Doubling Attack (5 points)

La « doubling attack » permet de retrouver l'exposant privé *d* d'une exponentiation modulaire réalisée avec la méthode *square-and-multiply-always*. Vous disposez des deux procédure et fonction suivantes :

```
procedure acquire_and_split(E m : entier, S trace[] : tableau de segments de traces)

fonction compare_trace_segments(E segment_1, segment_2 : segment de trace) : booléen
```

La procédure `acquire_and_split` acquiert une trace side-channel de l'exécution de l'exponentiation modulaire de la valeur d'entrée *m* (les éléments de la clé utilisée sont implicites) et effectue un post-traitement consistant à « découper » la trace en un tableau de segments correspondant aux produits modulaires successifs.
La fonction `compare_trace_segments` prend en entrée deux segments de traces et renvoie la valeur `vrai` si les opérandes impliquées dans ces segments sont identiques, et `faux` sinon.

a) En utilisant ces procédures et fonctions, écrivez le pseudo-code de la doubling attack appliquée à une exponentiation de type *square-and-multiply-always*.
(Vous choisirez vous-mêmes le prototype de votre fonction ou procédure.)

b) Est-ce que cette attaque continue à être réalisable si l'exponentiation est protégée par :

- un blinding d'exposant seul ?
- un blinding de message seul ?
- un blinding de module seul ?
- un blinding de message conjoint à un blinding de module ?

Dans chaque cas vous justifierez votre réponse.

2) La Doubling Attack 2, le retour (5 points)

a) Écrivez le pseudo-code de la doubling attack dans le cas où l'exponentiation modulaire est réalisée selon la méthode du *Montgomery ladder*.

b) Répondez également à la question concernant l'applicabilité de l'attaque sous les différentes hypothèses de blinding dans le cas du *Montgomery ladder*.

3) Blinding d'exposant (3 points)

a) Expliquez en quoi consiste le blinding d'exposant appliqué à une exponentiation modulaire. De quels types d'attaques cette contre-mesure est-elle sensée protéger ?

b) Expliquez pourquoi le blinding d'exposant est inutile si l'attaquant est capable de retrouver l'exposant utilisé lors d'une exponentiation par SPA en n'utilisant qu'une seule trace.

c) On suppose une clé RSA de 1024 bits, et un aléa de blinding d'exposant de 32 bits. Quel est le coût de cette contre-mesure (surcoût par rapport à une implémentation non protégée) en terme de temps d'exécution ?
(Rappel : l'exponentiation modulaire est de complexité cubique.)

4) Autour de la DPA et de la CPA (5 points)

a) Dans une 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.

b) Dans une CPA sur l'AES avec modèle de consommation en poids de Hamming, combien l'attaquant doit-il générer de courbes de CPA pour retrouver la valeur d'un octet de clé ? Combien de courbes de CPA pour retrouver la clé complète ? Justifiez vos réponses.

c) Répondez à la question b) ci-dessus dans le cas d'un modèle de consommation en distance de Hamming vis-à-vis d'une constante inconnue (état de référence) identique d'un calcul de S-Box à un autre.

d) Dans une CPA sur l'AES avec modèle de consommation en poids de Hamming, quel avantage l'attaquant retire-t-il d'envoyer à la carte uniquement des messages formés de 16 valeurs d'octet identiques ? Pourquoi cette astuce ne s'applique-t-elle pas au DES ?

5) Analyse différentielle de fautes sur le DES (2 points)

L'analyse différentielle de fautes sur le DES permet de retrouver la clé de tour K16 en analysant l'effet de fautes introduites durant l'exécution du 15ème tour.

Selon vous, est-il possible d'adapter cette attaque à un Triple DES ? Si oui, expliquez comment vous mèneriez cette attaque.