M1 informatique – UE réseau et sécurité
Année universitaire : 2021-2022
Consignes : documents interdits, calculatrices interdites, téléphones portables interdits.
Durée : 1h30

Réseau et sécurité – Examen terminal – deuxième session

Exercice 1 : sécurité cryptographique (4 points)

La sécurité cryptographique d'un algorithme est le logarithme du temps d'exécution de son attaque la plus rapide.
Question 1.1 (1 point) : Expliquez pourquoi la sécurité cryptographique d'un algorithme ne peut pas dépasser la longueur en bits de la clé qu'il utilise.
Question 1.2 (1 point) : Les clés RSA de 1024 bits ont une sécurité cryptographique équivalente à des clés symétriques de 80 bits. Expliquez cette différence entre 80 bits et 1024 bits.
Question 1.3 (1 point) : La sécurité cryptographique d'un algorithme décroît avec le temps. Expliquez pourquoi.
Question 1.4 (1 point) : Le principe de Kerckhoff [1] en sécurité indique que la sécurité d'un système cryptographique ne doit se baser que sur la confidentialité de la clé, pas sur la confidentialité de l'algorithme. En d'autres termes, il faut s'attendre à ce que l'attaquant connaisse les détails de fonctionnement du système. Donnez un exemple indiquant pourquoi ce principe est communément accepté.

Exercice 2 : attaque différentielle de DES (5 points)

S-Box 5		4 bits au centre de l'entrée															
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Bits externes	00	0010	1100	0100	0001	0111	1010	1011	0110	1000	0101	0011	1111	1101	0000	1110	1001
	01	1110	1011	0010	1100	0100	0111	1101	0001	0101	0000	1111	1010	0011	1001	1000	0110
	10	0100	0010	0001	1011	1010	1101	0111	1000	1111	1001	1100	0101	0110	0011	0000	1110
	11	1011	1000	1100	0111	0001	1110	0010	1101	0110	1111	0000	1001	1010	0100	0101	0011

Figure 1 : Schéma de la S-Box 5 de DES.

Question 2.1 (1 point) : Décrivez la différence entre le chiffrement par flot et le chiffrement par bloc.
Question 2.2 (1 point) : Donnez la séquence binaire sortant de la S-Box 5, *O*, selon la Figure 1, en sachant que l'entrée *I* vaut *010111*.
Question 2.3 (1 point) : Expliquez pourquoi la conception des S-Box est la source de la sécurité du DES.
Question 2.4 (2 points) : Retrouvez les clés possibles telles que $I_1 \oplus I_2 = 100001$, $O_1 \oplus O_2 = 0111$ et $E_1 = 101110$. (Rappelons que $I = E \oplus K$). Vous pourrez remarquer la valeur particulière $I_1 \oplus I_2$ pour réduire le nombre de calculs.

Exercice 3 : attaque à texte chiffré choisi de RSA (11 points)

Cet exercice se base sur une attaque découverte par D. Bleichenbacher dans [2]. Soit *(n,e)* une clé publique RSA et *(n,d)* la clé privée correspondante. Soit *k* la longueur de *n* en octets. Le standard PKCS #1, qui définit l'encryption RSA, indique qu'un texte clair conforme doit avoir le format suivant : (0x00, 0x02, *PS*, 0x00, *D*), où *D* correspond aux données à encrypter, et *PS* est une chaîne de caractères

aléatoire ne contenant aucun 0x00 et de longueur 8. La longueur en octets de *D* doit être inférieure ou égale à *k*-11.

Question 3.1 (1 point) : Expliquez pourquoi la longueur en octets de *D* doit être inférieure ou égale à *k-11* ?
Question 3.2 (1 point) : En quoi le fait que *PS* est aléatoire augmente la sécurité du chiffrement ?

Un texte clair est conforme si son premier octet est 0x00, son deuxième octet est 0x02, les huit octets suivants (correspondant à *PS*) sont différents de 0x00, et l'octet suivant vaut 0x00. Un texte chiffré est dit conforme si son déchiffrement produit un texte clair conforme.

L'attaque utilise des textes chiffrés choisis et la capacité d'un utilisateur à déterminer si un texte chiffré est conforme ou non. Cette capacité est appelée un oracle. L'attaque fonctionne de la manière suivante. Supposons qu'un attaquant veut trouver $m \equiv c^d \ [n]$ à partir d'un texte chiffré *c*. L'attaquant utilise l'oracle pour savoir si $c.s^e \ [n]$ est un texte chiffré conforme, pour différentes valeurs de *s*. Quand c'est le cas, l'attaquant peut déduire que $m.s \ [n]$ commence par (0x00, 0x02), ce qui réduit les possibilités pour *m*.

Question 3.3 (1 point) : Montrez que si $c.s^e \ [n]$ est un texte chiffré conforme, alors $m.s \ [n]$ commence avec (0x00, 0x02).
Question 3.4 (1 point) : Soit $B=2^{8(k-2)}$. Montrez que si $m.s \ [n]$ est conforme, alors $2B \leq m.s \ [n] < 3B$.
Question 3.5 (1 point) : Expliquez comment il est possible de trouver *m* avec un nombre suffisant de tentatives.
Question 3.6 (1 point) : Soit Pr(*A*) la probabilité qu'un entier aléatoirement choisi dans *[0;n[* ait son premier octet égal à 0x00, et son deuxième octet égal à 0x02. Exprimez Pr(*A*) en fonction de *B* et de *n*.
Question 3.7 (1 point) : Soit Pr(*P*|*A*) la probabilité qu'un entier aléatoirement choisi dans *[0;n[* ait les octets numéros trois à dix différents de 0, et l'octet numéro onze égal à 0, en supposant que l'octet numéro un est 0x00 et que l'octet numéro deux est 0x02. Exprimez Pr(*P*|*A*) sans le simplifier.
Question 3.8 (1 point) : En moyenne, combien de tentatives sont nécessaires pour trouver un nombre *s* tel que $c.s^e \ [n]$ soit un texte chiffré conforme ? Ne simplifiez pas l'expression.
Question 3.9 (1 point) : A partir de cette formule de Pr(*P*|*A*) et d'une heuristique efficace pour choisir la prochaine valeur de *s*, l'auteur de [1] a montré que 2^20 tentatives sont nécessaires pour *k*=128 (ce qui correspond à une valeur de *n* de 1024 bits). Est-ce que ce nombre de tentatives est trop élevé pour être réalisable en pratique ?
Question 3.10 (1 point) : L'oracle peut être implémenté si le système de déchiffrement envoie une erreur à l'utilisateur quand le texte chiffré n'est pas conforme. Pour éviter cela, les auteurs proposent d'ajouter dans le message clair un contrôle d'intégrité, et de vérifier l'intégrité du message clair juste après le déchiffrement. En quoi cela évite l'implémentation de l'oracle ?
Question 3.11 (1 point) : L'oracle peut être implémenté avec une attaque temporelle, si le système de déchiffrement intègre une vérification de signature et fonctionne ainsi : (1) recevoir le texte chiffré *c*, (2) déchiffrer *c* en *m*, (3) envoyer « non » si *m* n'est pas conforme, (4) vérifier la signature, (5) envoyer « oui » si la signature est correcte et « non » si la signature est incorrecte. Expliquez l'attaque temporelle correspondante.

[1] A. Kerckhoff. « La cryptographie militaire ». Journal des sciences militaires, v. 9, pages 5—83, 1983.
[2] D. Bleichenbacher. Chosen ciphertext attacks against protocols based on the RSA encryption standard. Dans *Advances in Cryptology: Proceedings of CRYPTO'98*, volume 1462 de LNCS, pages 1—12, 1998.