Université Côte d'Azur M1 Informatique
Sécurité 2023–2024

Examen de juin 2024

Durée : 2h

Note :

Nom :
Prénom :

L'examen comporte 4 parties et 20 questions notées sur 1 point. Répondez sur la copie avec clarté et concision.

1 Quizz sur la sécurité (5 points)

1. Expliquez pourquoi `OpenSSL` demande la clé privée lors de la création d'une *certificate signing request*.

2. Zoom est passé du chiffrement AES-ECB à AES-GCM. Est-ce qu'il s'agit d'un changement important pour la sécurité (justifiez rapidement) ?

3. Listez et expliquez brièvement les différents modèles de confiance utilisés pas les infrastructures à clé publique.

4. Donnez deux exemples de fonctions à sens unique et les mécanismes de sécurité associés.

5. Expliquez la différence entre un protocole de mise en accord et un protocole de transport de clé.

2 Sécurité de la signature par El Gamal [5 points]

Étudions quelques propriétés de la signature par El Gamal. Rappel du fonctionnement : on choisit $k \in \mathbb{Z}^\star_{p-1}$ aléatoire et secret, inversible dans $\mathbb{Z}^\star_{p-1}$. On définit la signature du message M comme :

$$\text{sig}_{SK}(M, k) = (\gamma, \delta) \qquad \text{pour}$$

$$\gamma = \alpha^k \mod p \quad \text{et} \quad \delta = (M - a\gamma)k^{-1} \mod (p-1)$$

2.1 Problème sur la valeur aléatoire

Supposons que le générateur de nombres pseudo-aléatoires d'Alice retourne le paramètre privé a comme valeur aléatoire.

1. Quelle est la probabilité de cet événement ?

2. Comment un indiscret qui intercepte (M, γ, δ) peut-il s'en rendre compte ?

3. Comment notre indiscret peut-il utiliser ce fait à son avantage et retrouver la valeur de a, donc casser le mécanisme de signature d'Alice ?

4. Expliquez comment on peut contrer cette vulnérabilité.

2.2 Variation autour d'El Gamal

Il existe plusieurs variantes du schéma de signature d'El Gamal qui sont toutes obtenues en modifiant l'équation de signature : $\delta = (M - a\gamma)k^{-1} \mod (p-1)$.
1. On considère l'équation de signature $\delta = a\gamma + kM \mod (p-1)$. Montrer la validité du prédicat $\text{Ver}(M, \gamma, \delta) = \texttt{Vrai} \Leftrightarrow \alpha^{\delta} = (\alpha^{a})^{\gamma}\gamma^{m} \mod p$.

3 Les collisions de Luhn [4 points]

On doit l'invention du hachage à H. P. Luhn, ingénieur chez IBM qui a proposé une formule en 1953, toujours utilisée pour vérifier la validité d'un identifiant numérique (e.g. carte de crédit). La formule génère un chiffre de vérification, qui est généralement annexé à la suite de l'identifiant numérique pour construire un identifiant complet. Cet identifiant complet (identifiant numérique et son chiffre de vérification) est soumis à l'algorithme suivant pour vérifier sa validité :

1. On démarre avec le dernier chiffre de l'identifiant (à droite) et on se déplace vers la gauche, en doublant la valeur de tous les chiffres de rang pair : le dernier chiffre est traité en 1er, il n'est pas doublé, l'avant-dernier (2e) sera doublé. Si le double d'un chiffre dépasse 9, on le remplace par la somme de ses chiffres. Par exemple, 8 763 devient 7 733 (car $2 \times 6 = 12$, et 1+2=3 ; $2 \times 8 = 16$, et 1+6=7).
2. On additionne ensemble tous les chiffres du nombre ainsi obtenu. Par exemple, 8763 (transformé en 7733 après l'étape 1) donne 20=7+7+3+3.
3. Si le total est un multiple de 10 le nombre est valide, en accord avec la formule de Luhn. Sinon il est invalide. Ainsi 8763 est valide (cf. ci-dessus, le calcul donne 20).

1. Dites si le nombre 1953 est un identifiant valide ou non (détaillez les calculs).

Pour déterminer le chiffre de vérification, on calcule la somme comme décrit ci-dessus (attention, l'identifiant complet est sous la forme `I...IV`) :

1. Calculer la somme de l'identifiant complet (avec `V`= 0) ;
2. multiplier par 9 ;
3. prendre comme chiffre de vérification le chiffre des unités obtenu.

2. Calculez le chiffre de vérification `V` à ajouter à 2024 pour que 2024`V` soit un identifiant complet valide.

3. On cherche à savoir si la formule de Luhn est robuste aux collisions. Pour cela, calculez combien il faut engendrer d'identifiants incomplets pour obtenir un même chiffre de vérification avec une probabilité de 3/4. On rappelle que $\ln 4 \approx 1,4$ et $\ln 3 \approx 1,1$

4. Saurez-vous trouver une collision pour des identifiants complets sur 4 chiffres ? Expliquez comment vous procédez et donnez deux identifiants qui ont le même chiffre de vérification.

4 Protocole Otway-Rees modifié (6 points)

On considère le protocole d'authentification entre deux parties A et B défini comme suit :

1. $A \to B : I, A, B, \{N_A, I, A, B\}_{K_{AS}}$
2. $B \to S : I, A, B, \{N_A, I, A, B\}_{K_{AS}}, \{N_B, I, A, B\}_{K_{BS}}$
3. $S \to B : I, \{N_A, K_{AB}\}_{K_{AS}}, \{N_B, K_{AB}\}_{K_{BS}}$
4. $B \to A : I, \{N_A, K_{AB}\}_{K_{AS}}$

Dans ce protocole, I désigne un identifiant du protocole, K_{XY} indique une clé secrète partagée entre les entités X et Y, N_X un jeton émis par l'entité X. A et B désignent deux entités et S un serveur. L'opération $\{M\}_K$ représente le chiffrement du message M en utilisant la clé symétrique K.

1. Dessinez ci-dessous les étapes de ce protocole et identifiez s'il s'agit d'un protocole de type "pull" ou "push" et dites à quoi sert ce protocole.

2. Expliquez comment ce protocole permet à A de s'authentifier auprès de B.

3. Dites pourquoi ce protocole n'authentifie pas B auprès de A (et ce que cela signifie pour la sûreté de ce protocole).

4. Comment faudrait-il modifier ce protocole pour le transformer en un modèle dit "mixte" ?

5. Pensez-vous que la sécurité de ce protocole serait renforcée en utilisant des chiffres asymétriques (justifiez) ?

6. En utilisant des chiffres asymétriques, quelle seraient les hypothèses sur les clés nécessaires au bon fonctionnement ?