CRYPTO

Chiffrements asymétriques

Loïc Rouquette

2025-03-10

Correction du QCM

Question 1

Quelle est la principale caractéristique d'un chiffrement symétrique ?

- Il est plus lent que le chiffrement asymétrique
- Il utilise une paire de clés publique et privée
- Il ne nécessite pas de clé
- Il utilise la même clé pour chiffrer et déchiffrer
- La question n'a pas de sens
- Les informations fournies dans l'énoncé ne sont pas suffisantes pour répondre à la question

Question 1

Quelle est la principale caractéristique d'un chiffrement symétrique ?

- **Il utilise la même clé pour chiffrer et déchiffrer**

> Les chiffrements symétriques utilisent une clé pour chiffrer et déchiffrer. Cette clé est appelée clé secrète.
> Les chiffrement **asymétriques** utilisent deux clés différentes appelées clé privé et clé publique
> Les fonction de hashage et les PRNGs n'utilisent pas de clé.

Question 2

Lequel(s) des algorithmes suivants est (sont) un (des) algorithme(s) de chiffrement symétrique ?

- RSA
- Diffie-Hellman
- AES
- DES
- ECB
- SHA-3
- La question n'a pas de sens
- Les informations fournies dans l'énoncé ne sont pas suffisantes pour répondre à la question

Question 2

Lequel(s) des algorithmes suivants est (sont) un (des) algorithme(s) de chiffrement symétrique ?

- **AES**
- **DES**

> **ECB** est un mode d'opération
>
> **SHA-3** est une fonction de hashage
>
> **RSA** est un chiffrement asymétrique **Diffie-Hellman** est un protocole d'échange de clé

Question 3

Quel est l'un des principaux inconvénients du chiffrement symétrique ?

- Il est trop lent pour les grandes quantités de données
- Il nécessite un échange sécurisé de clé
- Il ne permet pas le chiffrement
- Il ne fonctionne pas avec des données binaires
- La question n'a pas de sens
- Les informations fournies dans l'énoncé ne sont pas suffisantes pour répondre à la question

Question 3

Quel est l'un des principaux inconvénients du chiffrement symétrique ?

- **Il nécessite un échange sécurisé de clé**

Question 4

Quel mode d'opération de chiffrement symétrique est vulnérable à des attaques par substitution de blocs identiques ?

- CBC (Cipher Block Chaining)
- ECB (Electronic Codebook)
- OFB (Output Feedback)
- CTR (Counter)
- La question n'a pas de sens
- Les informations fournies dans l'énoncé ne sont pas suffisantes pour répondre à la question

Question 4

Quel mode d'opération de chiffrement symétrique est vulnérable à des attaques par substitution de blocs identiques ?

- **ECB (Electronic Codebook)**

Question 5

5 personnes souhaitent communiquer entre elles sans que leurs échanges soient révélées aux autres en utilisant un chiffrement symétrique. Combien de clé(s) faut-il générer ?

- 1
- 5
- 10
- 15
- 20

Question 5

5 personnes souhaitent communiquer entre elles sans que leurs échanges soient révélées aux autres en utilisant un chiffrement symétrique. Combien de clé(s) faut-il générer ?

- **10**

> Il faut une clé pour chaque paire de personnes.

Question 6

Pourquoi un CSPRNG doit-il être résistant aux attaques de prédiction ?

- Pour éviter qu'un attaquant puisse deviner les valeurs futures générées
- Pour garantir qu'il est plus rapide que les autres générateurs
- Pour pouvoir être utilisé dans les jeux vidéo
- Pour réduire la consommation d'énergie d'un processeur
- La question n'a pas de sens
- Les informations fournies dans l'énoncé ne sont pas suffisantes pour répondre à la question

Question 6

Pourquoi un CSPRNG doit-il être résistant aux attaques de prédiction ?

- **Pour éviter qu'un attaquant puisse deviner les valeurs futures générées**

Question 7

Quelle(s) source(s) d'entropie peut être utilisée(s) pour un TRNG ?

- Les horloges du processeur
- Le bruit thermique d'un composant électronique
- Une séquence générée par un algorithme déterministe
- Une table de valeurs préenregistrées
- Le Trackpad
- La question n'a pas de sens
- Les informations fournies dans l'énoncé ne sont pas suffisantes pour répondre à la question

Question 7

Quelle(s) source(s) d'entropie peut être utilisée(s) pour un TRNG ?

- **Le bruit thermique d'un composant électronique**
- **Le Trackpad**

Question 8

Quelles sont les propriétés nécessaires pour assurer la sécurité du chiffrement de Vernam ?

- Il faut utiliser un XOR à la place de l'addition avec congruence
- Il faut que la clé face la taille du texte
- Il faut que la clé soit parfaitement aléatoire
- Il faut que la clé ne soit pas réutilisée
- Il faut que la clé reste secrète
- La question n'a pas de sens
- Les informations fournies dans l'énoncé ne sont pas suffisantes pour répondre à la question

Question 8

Quelles sont les propriétés nécessaires pour assurer la sécurité du chiffrement de Vernam ?

- ~~Il faut utiliser un XOR à la place de l'addition avec congruence~~
- **Il faut que la clé face la taille du texte**
- **Il faut que la clé soit parfaitement aléatoire**
- **Il faut que la clé ne soit pas réutilisée**
- **Il faut que la clé reste secrète**

Question 9

Étant donné qu'il est nécessaire d'ajouter un mode d'opération à un chiffrement par bloc pour obtenir un chiffrement par flux. Pouvons-nous affirmer que les chiffrements par flux sont plus sécurisés que les chiffrements par bloc ?

- Oui
- Non
- La question n'a pas de sens
- Les informations fournies dans l'énoncé ne sont pas suffisantes pour répondre à la question

Question 9

Étant donné qu'il est nécessaire d'ajouter un mode d'opération à un chiffrement par bloc pour obtenir un chiffrement par flux. Pouvons-nous affirmer que les chiffrements par flux sont plus sécurisés que les chiffrements par bloc ?

- **Non**
- **La question n'a pas de sens**

Question 10

Quel est l'inconvénient majeur d'un TRNG (True Random Number Generator) ?

- Il est plus lent et coûteux à mettre en œuvre que les PRNG
- Il produit des nombres prévisibles après un certain temps
- Il ne peut pas être utilisé pour générer des clés cryptographiques
- Il est basé sur des algorithmes mathématiques
- La question n'a pas de sens
- Les informations fournies dans l'énoncé ne sont pas suffisantes pour répondre à la question

Question 10

Quel est l'inconvénient majeur d'un TRNG (True Random Number Generator) ?

- **Il est plus lent et coûteux à mettre en œuvre que les PRNG**

Rappels du Dernier Cours

Les 5 propriétés de la sécurité

- Disponibilité
- Authentification
- Intégrité
- Non-Répudiation
- Confidentialité

Les 5 propriétés de la sécurité(ii)

- ~~Disponibilité~~ -> ne concerne pas la cryptographie
- Authentification
- ~~Intégrité~~ -> les fonctions de hashage
- Non-Répudiation
- Confidentialité -> les chiffrements symétriques ?

Les 5 propriétés de la sécurité(iii)

- ~~Disponibilité~~ -> ne concerne pas la cryptographie
- Authentification
- ~~Intégrité~~ -> les fonctions de hashage
- Non-Répudiation
- **Confidentialité**

Les chiffrements asymétriques

Rappel - Canal confidentiel contre les attaques passives

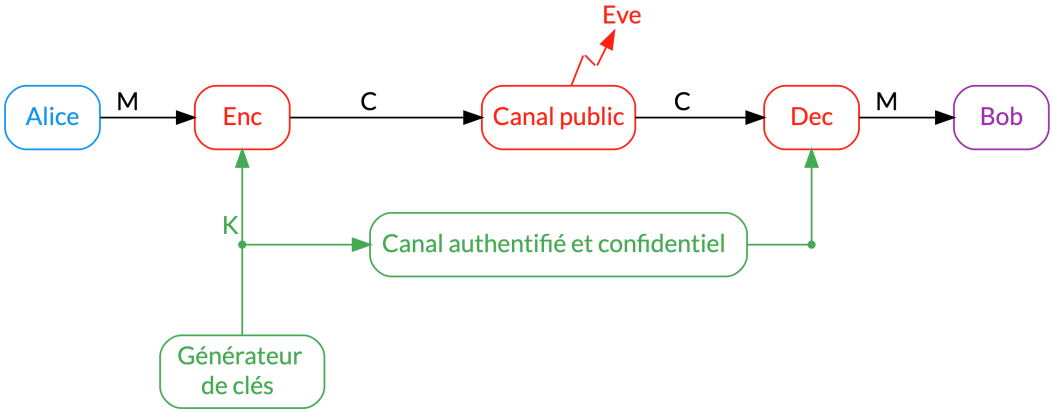

Figure 1. – Une communication utilisant un chiffrement symétrique. On a besoin du canal authentifié et confidentiel **une fois** au préalable de l'envoi d'un **nombre élevé de messages**.

Rappel - Canal confidentiel contre les attaques passives(ii)

Quelle est la problématique ?

Rappel - Canal confidentiel contre les attaques passives(iii)

Comment fait-on ?

Rappel - Canal confidentiel contre les attaques passives(iv)

On échange la clé par divers procédés :

- ambassadeurs, messagers ;
- registres de clés (qui sont connus avant d'initier la communication).

New directions in cryptography

En 1976, **Whitfield Diffie** et **Martin Hellman**, créent une méthode révolutionnaire permettant d'échanger un secret sans utiliser de chiffrement symétrique.

Il s'agit de l'**Échange de clés Diffie-Hellman.**

New directions in cryptography(ii)

Alice	Bob
Génération de clé	
Choisir p un nombre premier et g un générateur de $\mathbb{Z}/p\mathbb{Z}$	
Choisir $a < p$ Envoyer $K_a = g^a \bmod p$ $K = (g^b)^a$	Choisir $b < p$ Envoyer $K_b = g^b \bmod p$ $K = (g^a)^b$
Clé privée : $K = g^{ab} \bmod p = g^{ba} \bmod p$	

New directions in cryptography(iii)

Pourquoi ça marche ?

Les fonctions à sens unique

Une fonction calculable en temps polynomial $f : \{0,1\}^* \to \{0,1\}^*$ est dite à sens unique si pour tout algorithme polynomial probabiliste A, il existe une fonction négligeable $\epsilon : \mathbb{N} \to [0;1]$ telle que pour tout n on ait:

$$\Pr_{x \in_R \{0,1\}^n,\ y=f(x)}[A(y) = x' \text{ telle que } f(x') = y] < \epsilon(n).$$

– Wikipedia

Les fonctions à sens usuelles (pour le moment)

Le problème du logarithme discret

Problème du logarithme discret

Soit $g \in (\mathbb{Z}/p\mathbb{Z})^\times$ un générateur et $b \in (\mathbb{Z}/p\mathbb{Z})^\times$. Le problème du logarithme discret consiste à résoude l'équation : $g^x = b$ pour x dans $\mathbb{Z}/p\mathbb{Z}$, *i.e.* résoude :

$$g^x \equiv b \mod p$$

L'entier $x = log_g(b)$ est appelé logarithme discret de b dans la base g.

Les fonctions à sens usuelles (pour le moment)(ii)

Problème du logarithme discret

Soit $g \in (\mathbb{Z}/p\mathbb{Z})^\times$ un générateur et $b \in (\mathbb{Z}/p\mathbb{Z})^\times$. Le problème du logarithme discret consiste à résoude l'équation : $g^x = b$ pour x dans $\mathbb{Z}/p\mathbb{Z}$, *i.e.* résoude :

$$g^x \equiv b \mod p$$

L'entier $x = log_g(b)$ est appelé logarithme discret de b dans la base g.

Proposition

$$log_g(b_1 b_2) = log_g(b_1) + log_g(b_2)$$

$$log_g(b^n) = n\, log_g(b)$$

Attaques

Attaque par force brute

En général, il s'agit d'essayer toutes les possibilités, ce qui signifie ici calculer toutes les puissances de g jusqu'à la collision

Complexité : $\mathcal{O}(p)$

Solution : Prendre un nombre premier p très grand

Attaques(ii)

Algorithme de Shank

Il s'agit d'un algorithme de collision. On construit deux liste d'éléments de $\mathbb{Z}/p\mathbb{Z}$, et on cherche un élément qui apparaît dans les deux listes (collision).

Complexité : $\mathcal{O}(n\ log(n)) \approx \mathcal{O}(\sqrt{p}\ log(\sqrt{p}) \approx \mathcal{O}(\sqrt{p}\ log(p))$

Solution : Prendre un nombre premier p **vraiment** très grand

Les fonctions à sens usuelles (pour le moment)

Alice	Bob
Génération de clé	
Choisir p un nombre premier et g un générateur de $\mathbb{Z}/p\mathbb{Z}$ et calculer $p = pq$ Calculer $\varphi(n) = (p-1)(q-1)$ Choisir $e < \varphi(n)$ tel que $\gcd(e, \varphi(n)) = 1$ Calculer d tel que $de = 1 \bmod \varphi(n)$ La clé privée $\mathrm{sk} = d$ La clé publique $\mathrm{pk} = (n, e)$	
Chiffrement	
Calculer $c = \mathrm{Enc}(\mathrm{pk}, m) = m^e \bmod n$ Envoyer le message chiffré c	
Déchiffrement	
	Calculer $\mathrm{Dec}(\mathrm{sk}, c) = c^d \bmod n$ Le message clair est $m = c^d$

Résumé de l'algorithme

- $n = pq$ avec p et q étant des nombres premiers ;
- $e \cdot d \equiv 1 \mod (p-1)(q-1)$
- Chiffrement $c = m^e$
- Déchiffrement $m = c^d$
- $(m^e)^d = d$ et $\left(m^d\right)^e = m$

Est-il possible de retrouver m à partir de $c = m^e \mod n$?

RSA(ii)

Réponse : Il faut essayer de factoriser $n = pq$

Solution : Utiliser de grands nombres premiers pour p et q. Est-ce suffisant ?

$\varphi(n) = (p-1)(q-1)$ doit auss rester secret !

En effet :

$$(p-1)(q-1) = pq - p - q + 1 = n - (p+q) + 1$$

Comme n est connu, nous pouvons calculer $p + q$ et résoudre l'équation

$$X^2 - (p+q)X + pq = 0$$

Mais

$$(X-p)(X-q) = X^2 - (p+q)X + pq$$

De ce fait, résoudre l'équation donne p et q

Factorisation de n

- **Attaque par brute force :** tester tous les premiers jusqu'à $\sqrt{n}$
  - Complexité : $\mathcal{O}\left(\sqrt{n}\right)$
- **Algorithme de Fermat :**
  - Complexité : dépend de $|\ p - q\ |$. Il peut être très rapide si $|\ p - q\ |$ est petit
- **Algorithme rho de Pollard (probabiliste):**
  - Complexité : $\mathcal{O}\left(\sqrt{p}\ log^2 n\right) \approx \mathcal{O} n^{1/4} log^2 n$
- **Crible algébriue :** tester tous les premiers jusqu'à $\sqrt{n}$
  - Complexité : $\mathcal{O}\left(c \cdot e^{log(n)^{1/3}(log\ log\ n)^{2/3}}\right)$ avec $c \approx 1.92$

Schéma de communication

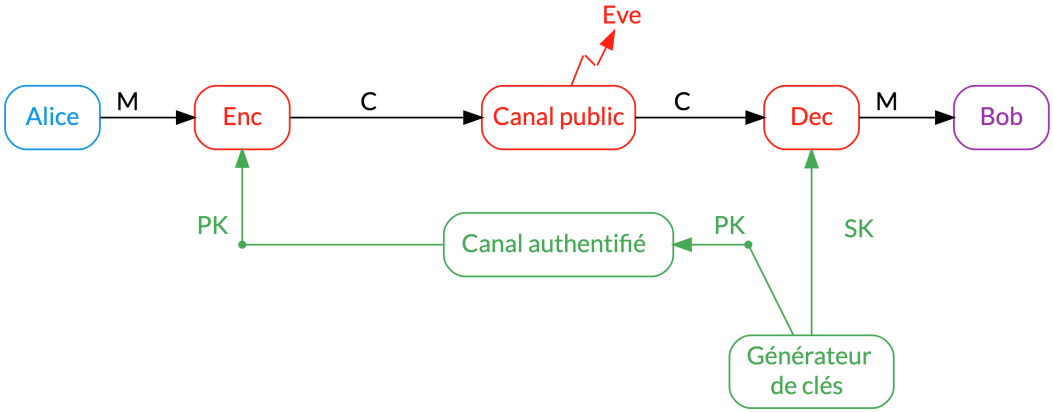

Figure 2. – Grâce au chiffrement asymétrique, nous n'avons plus besoin de canal confidentiel.

Les 5 propriétés de la sécurité

- ~~Disponibilité~~
- Authentification
- ~~Intégrité~~
- Non-Répudiation
- **Confidentialité** -> mix asymétrique, symétrique

Les 5 propriétés de la sécurité(ii)

- ~~Disponibilité~~
- **Authentification**
- ~~Intégrité~~
- **Non-Répudiation**
- ~~Confidentialité~~ -> mix asymétrique, symétrique

Authentification

Créér un canal authentifié

Utiliser un code d'authentification de message

Code d'authentification de message

Un **code d'authentification de message** (**MAC**) est un algorithme qui calcule une valeur de **taille fixe**, appelée aussi **MAC**, à partir de messages de **taille quelconque** et d'une **clé secrète** K partagée enter l'émetteur et le récepteur.

Utilisation d'un code d'authentification de message :

- Un canal public pour transmettre le message et leur code d'authentification ;
- Un canal authentifié **et** confidentiel pour transmettre la **clé secrète**.

Créér un canal authentifié(ii)

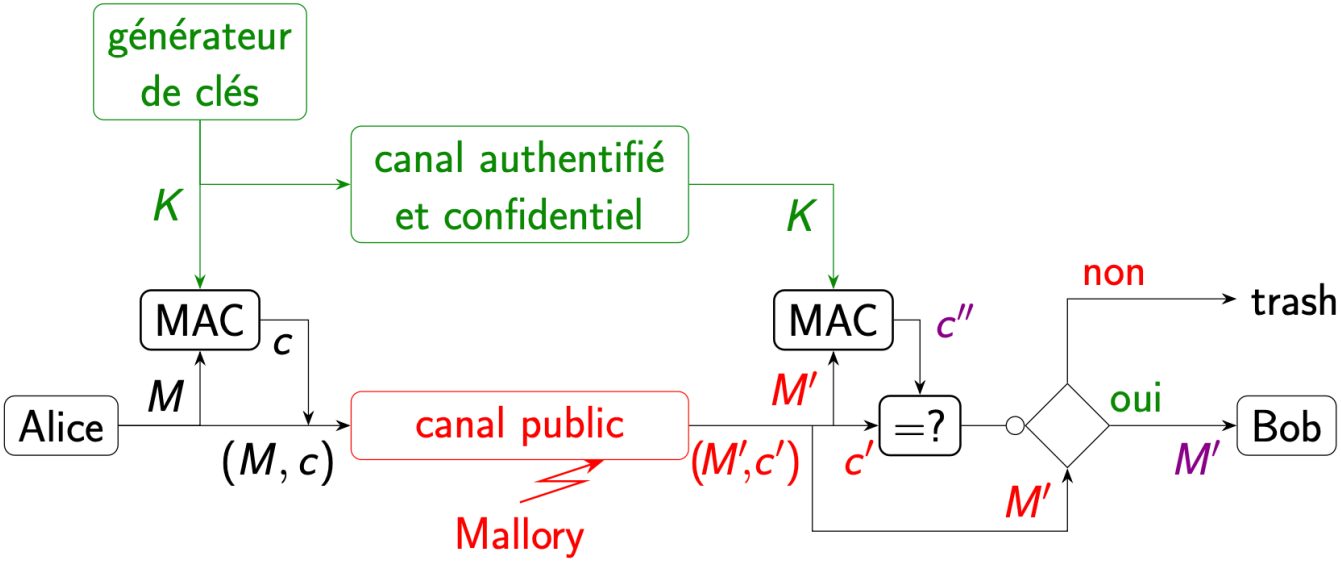

Figure 3. – Canal authentifié à l'aide d'un code d'authentification

Créér un canal authentifié(iii)

Utiliser un algorithme de signature

Signature

Un algorithme de **signature** calcule une valeur appelée **signature**, de **taille fixe**, à partir de messages de **taille quelconque** et de la **clé privée** K_S de l'émetteur. La **vérification** par le récepteur se fait grâce à la clé publique K_P de l'émetteur.

Utilisation d'une signature :

- Un canal public pour transmettre des messages et leur **signature**
- Un canal authentifié pour transmettre la **clé publique**.

Créér un canal authentifié(iv)

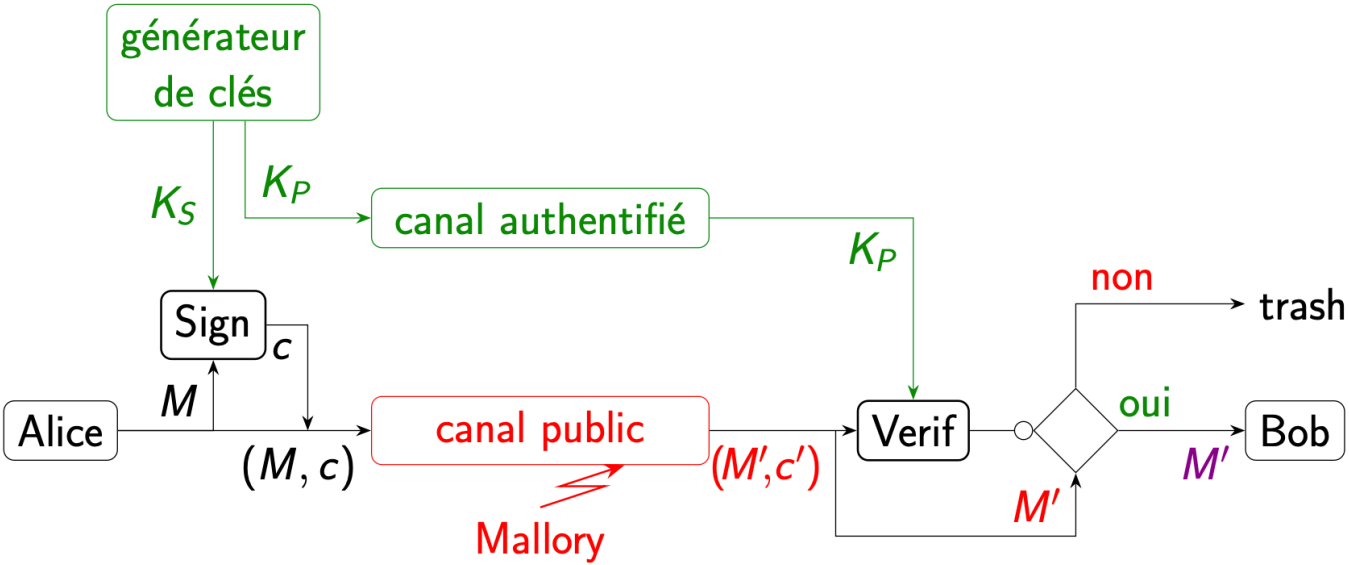

Figure 4. – Canal authentifié à l'aide d'une signature

Authentication $\Leftarrow \not\Rightarrow$ Signature

L'**authentification** permet de répondre à la question : « Qui a émis le message ? »

Pour savoir si on peut parler de **signature**, il faut savoir **qui pose la question ?**

- **MAC** : autres possesseurs de la clé $\Rightarrow$ une seule personne
  - deux personnes peuvent calculer l'authentifiant (donc pas de signature).
- **Signature** : autres possesseurs de la clé (publique) $\Rightarrow$ tout le monde
  - Une seule personne peut calculer l'authentifiant $\Rightarrow$ non-répudiation.