Cryptographie à clef publique
Cours 5

Julien Lavauzelle

Université Paris 8

Master 1 mathématiques et applications – parcours ACC et CSSD

05/03/2021

Séance précédente

Vu à la **séance précédente** :

- Notions de courbes elliptiques pour la cryptographie
- Chiffrement ElGamal
- Difficulté du problème du logarithme discret

Questions ?

Devoir à la maison

Lecture rapide du sujet

Questions ?

Aujourd'hui

1. Signatures numériques

2. Intermède

3. Quelques schémas de signature
   - Signature RSA
   - Signature ElGamal

4. TD ElGamal et courbes elliptiques

Plan

Contexte et motivations

On souhaite imiter (voire améliorer) certaines propriétés des signatures manuscrites.

Exemples de ce qu'on souhaite signer **numériquement** :

- des emails
- du code (mise à jour de logiciels)
- des transactions bancaires (ecommerce)
- de la communication publique (sites web)
- des clefs de chiffrement, des certificats

Objectifs :

- **Intégrité** : on peut vérifier si le message a été modifié ou non.
- **Authenticité** : on peut associer un message à un émetteur.
- **Non-répudiation** : on ne peut pas nier avoir émis une signature valide.
- **Infalsifiabilité** : une autre personne ne peut pas prétendre avoir émis la signature.
- **Non-réutilisation :** on ne peut pas utiliser une même signature sur deux messages différents.

Remarque. Ces propriétés ne sont pas toutes vérifiées par la signature « physique ».

Contexte et motivations

Remarque. Une signature n'est pas :

1. du chiffrement (on ne cache pas m),
2. « l'inverse du chiffrement asymétrique »,
3. un MAC (*message authentication code*) : les signatures sont publiquement vérifiables.

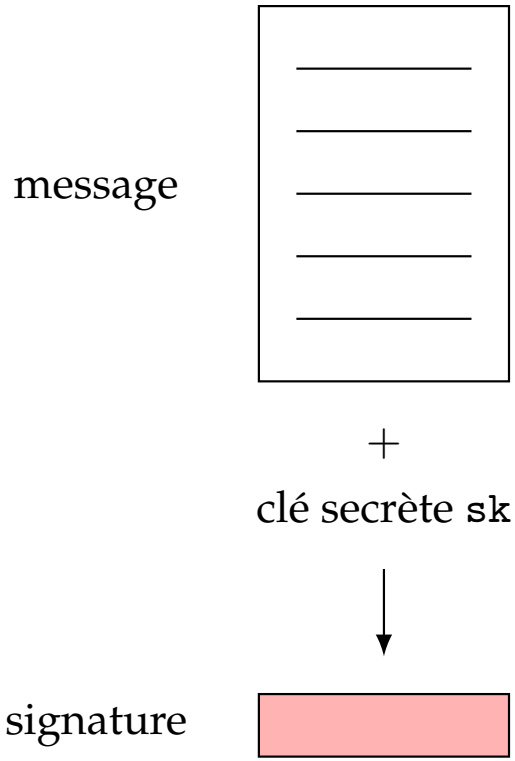
La signature va s'**apposer** au message, et devra dépendre explicitement de lui.

Par conséquent, des propriétés **additionnelles** désirables :

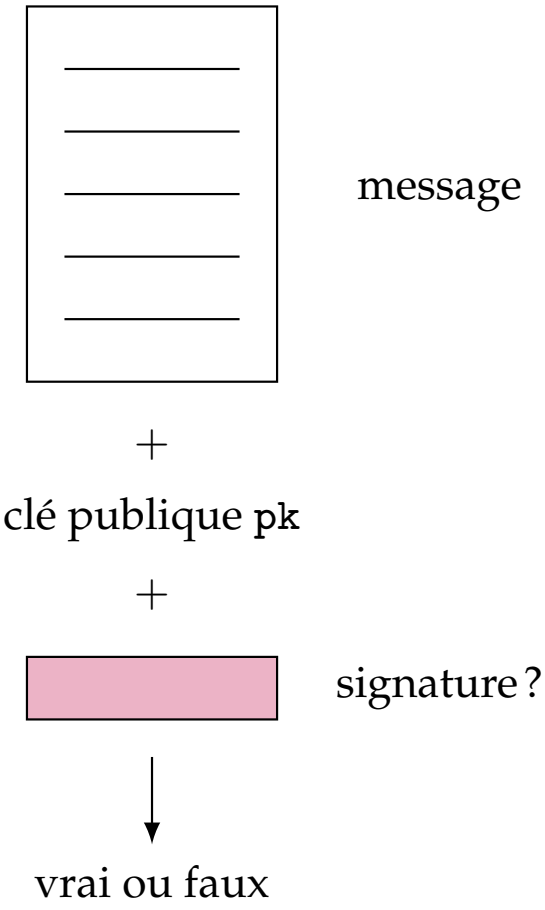
- des signatures courtes,
- des algorithmes de signature et vérification rapides,
- des clés courtes.

Un dessin récapitulatif

Phase de signature opérée par Alice



Phase de vérification opérée par quiconque



Définition

Définition. Un schéma de **signature numérique** (à clef publique) est constitué de deux ensembles et trois algorithmes :

1. $\mathcal{M}$ un **ensemble de messages**.
2. $\mathcal{S}$ un **ensemble de signatures**.
3. KeyGen l'**algorithme de génération de clefs**. Il renvoie un couple $(\mathtt{pk}, \mathtt{sk})$ de clé publique/clé privée.
4. Sign un **algorithme de signature**, qui prend en entrée un message $m \in \mathcal{M}$, la clé privée $\mathtt{sk}$, et retourne une signature $s = \mathsf{Sign}(m, \mathtt{sk}) \in \mathcal{S}$.
5. Verif un **algorithme de vérification**, qui renvoie une valeur booléenne true/false. L'algorithme Verif prend en entrée la clé publique $\mathtt{pk}$, le message m et la signature s.

Le schéma de signature est **valide** si

$$\forall m, s \in \mathcal{M} \times \mathcal{S}, \quad \mathsf{Verif}(m, s, \mathtt{pk}) = \mathsf{true} \iff s = \mathsf{Sign}(m, \mathtt{sk})$$

Modèle d'attaquant (moyens)

Pour la **sécurité** du schéma, il faut définir un modèle d'attaquant (moyens) et un modèle d'attaque (objectifs).

On distingue les moyens suivants :

1. **Attaque à clef seule** (*key-only attack*) : l'attaquant ne dispose que de la clef publique
2. **Attaque à message connu** (*known-message attack*) : l'attaquant dispose d'une liste de messages déjà signés $(m_1, s_1), \ldots, (m_\ell, s_\ell)$. Les signatures sont valides et réalisées avec la même clef.
3. **Attaque à message choisi** (*chosen-message attack*) : l'attaquant choisit des messages $m_1, \ldots, m_\ell$ et demande les signatures $s_1, \ldots, s_\ell$ associées. Les signatures sont valides et réalisées avec la même clef.

Modèle d'attaquant (objectifs)

Les modèles d'attaque sont les suivants :

1. **Cassage total** : l'attaquant détermine une clé privée équivalente à celle d'Alice.
2. **Falsification universelle** : avec probabilité non-négligeable, l'attaquant peut falsifier une signature d'un message choisi par quelqu'un d'autre. Ce message n'aura pas été signé précédemment.
3. **Falsification existentielle** : avec probabilité non-négligeable, l'attaquant peut créer un couple (m, s) où s est une signature valide de m. Ce message n'aura pas été signé précédemment.

Définition (exemple). On dit qu'un schéma satisfait la propriété d'**infalsification existentielle sous une attaque à message choisi** (EUF-CMA, *existential unforgeability* ...) si :

tout attaquant ayant accès à $\begin{cases} \text{la clé publique } \texttt{pk}, \\ \text{une liste de messages } m_1, \ldots, m_\ell \text{ qu'il a choisi,} \\ \text{et les signatures associées } s_1, \ldots, s_\ell, \end{cases}$

a une probabilité négligeable de retourner un message $m' \notin \{m_i\}$ et une signature s' tels que $\mathsf{Verif}(m', s', \texttt{sk}) = \mathsf{true}$.

$\Longrightarrow$ EUF-CMA est le standard de sécurité usuellement requis.

Remarque. Comme l'attaquant a accès à la procédure de vérification (publique), il est impossible d'obtenir un schéma de signature à sécurité inconditionnelle.

Plan

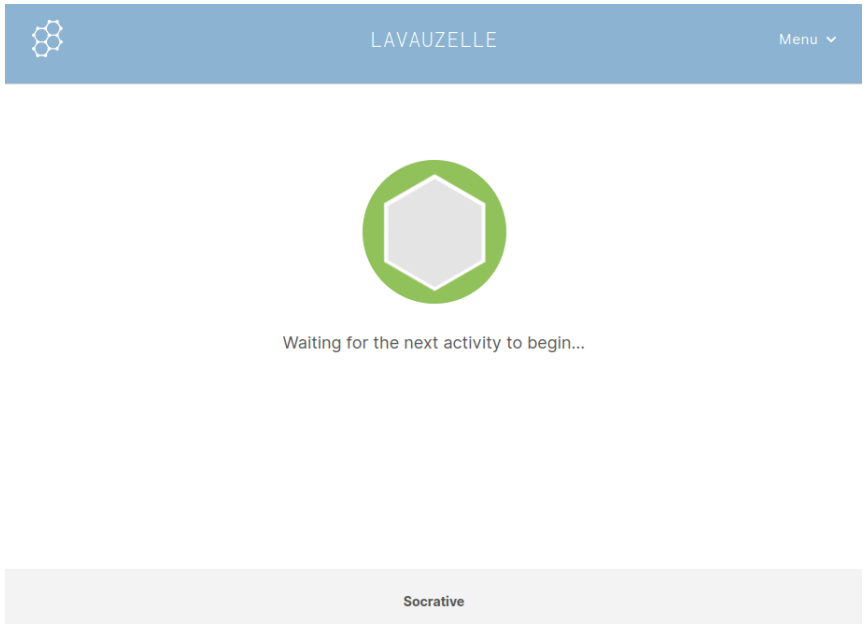
Un quiz

Quiz de révision. Voir ça comme un mini-TD. C'est **anonyme** !

- aller sur `https://b.socrative.com/login/student/`
- entrer LAVAUZELLE pour le nom de la salle,
- répondre aux questions quand elles s'affichent.

Question typique.
Y répondre puis « submit answer ».

Page d'attente de questions.
(si questionnaire terminé par ex.)



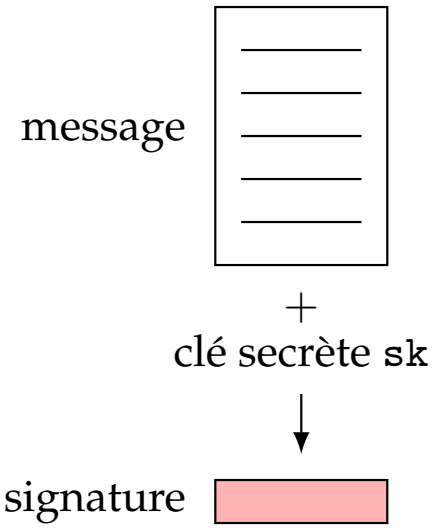
On fera un retour rapide quand tout le monde aura fini (5min max).

Plan

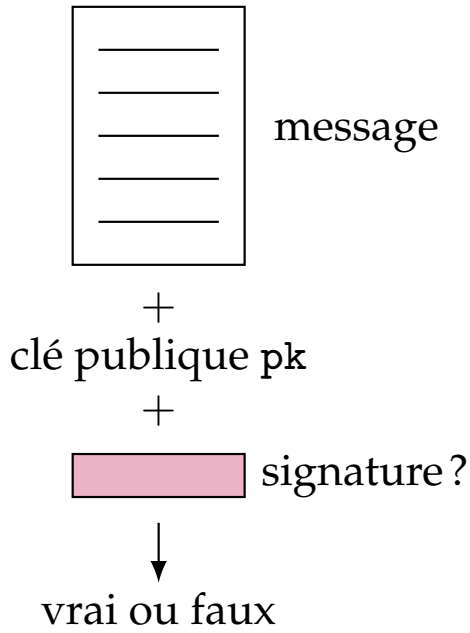
Plan

Première idée

Phase de signature opérée par Alice



Phase de vérification opérée par quiconque



Idée informelle : dans le chiffrement RSA, Alice est la seule à savoir inverser la fonction à sens-unique $f : x \mapsto x^e \mod n$. Mais tout le monde sait calculer f.

- la signature est $s = f^{-1}(m)$
- on vérifie publiquement que $f(s) = m$.

Signature RSA

Signature RSA : KeyGen

1. Calculer $n = pq$ et $\phi(n) = (p-1)(q-1)$, où p et q sont deux grand nombres premiers aléatoires
2. Choisir e et d tels que $ed \equiv 1 \mod \phi(n)$
3. La clé publique est $\texttt{pk} = (e, n)$, la clé privée est $\texttt{sk} = (d, \phi(n))$.

L'espace des messages est $\mathcal{M} = \mathbb{Z}/n\mathbb{Z}$ et celui des signatures est $\mathcal{S} = \mathbb{Z}/n\mathbb{Z}$.

Signature RSA : Sign$(m, \texttt{sk})$

1. Calculer $s = m^d \mod n$.
2. Retourner s.

Signature RSA : Verif$(m, s, \texttt{pk})$

1. Calculer $m' = s^e \mod n$.
2. Faire le test $m' = m$ et retourner le booléen associé.

Validité. $m' \equiv s^e \equiv m^{ed} \equiv m \mod n$.

Sécurité

Résumé.

- Clés : $\texttt{pk} = (n, e)$, $\texttt{sk} = d$,
- Signature : $s = m^d \mod n$
- Vérification : $s^e \mod n \overset{?}{\equiv} m \mod n$

Fait. Il existe une attaque de falsification existentielle sur la signature RSA « brute » avec la clé publique seule.

Preuve. À partir de la clé publique $\texttt{pk} = (n, e)$, on peut forger un message m' et une signature s' valide pour la clé privée $\texttt{sk} = d$ d'Alice.

1. Choisir s' aléatoirement dans $\mathcal{S} = \mathbb{Z}/n\mathbb{Z}$,
2. Calculer $m' = (s')^e \mod n$.

Remarque. La signature RSA brute n'est donc pas EUF-KOA (*key-only attack*).

Fait. Il existe une attaque de falsification universelle sur la signature RSA « brute », par message choisi. Autrement dit, RSA n'est pas UUF-CMA (UUF → *universal unforgeability*).

Preuve. Au prochain TD. *Indication :* 2 *messages préliminaires suffisent.*

Rappels sur les fonctions de hachage

Définition. Une famille de **fonctions de hachage cryptographiques** est un ensemble de fonctions $H_\kappa : \{0,1\}^* \to \mathcal{H}$, où $\mathcal{H}$ est un ensemble de taille fixe, et κ est un paramètre de la famille. L'élément $H_\kappa(\boldsymbol{m})$ est appelé **haché** de $\boldsymbol{m}$.

Une fonction de hachage $H = H_\kappa$ est **résistante aux collisions** si pour tout algorithme polynomial probabiliste $\mathcal{A}$, la probabilité

$$\mathbb{P}\left[\boldsymbol{h} \neq \boldsymbol{h} \text{ et } H_\kappa(\boldsymbol{m}) = H_\kappa(\boldsymbol{m}') \mid \mathcal{A}(t) = (\boldsymbol{h}, \boldsymbol{h}')\right]$$

est négligeable devant un paramètre de sécurité donné.

Exemple : les fonctions SHA-3 (*secure hash algorithm*), dont les sorties sont de taille 224, 256, 384 ou 512 bits (au choix).

Important. Les « anciennes » fonctions MD5 et SHA-1 ne sont pas résistantes aux collisions, mais restent utilisées pour des applications non-cryptographiques (à éviter néanmoins).

Dans tout la suite, on prend $H = H_\kappa : \{0,1\}^* \to \mathcal{H}$ une fonction de hachage résistante aux collisions.

RSA-FDH : *full domain hash*

Idée : au lieu de signer directement le message $\boldsymbol{m}$, on signe $H(\boldsymbol{m})$ avec l'algorithme de signature RSA « brut » vu précédemment.

La génération de clef est identique : $\texttt{pk} = (n, e)$, $\texttt{sk} = d$.

L'espace des messages est $\mathcal{M} = \{0,1\}^*$ et celui des signatures est $\mathcal{S} = \mathbb{Z}/n\mathbb{Z}$.

Signature RSA-FDH : $\mathsf{Sign}(\boldsymbol{m}, \texttt{sk})$

On suppose que $\boldsymbol{m} \in \{0,1\}^*$ et que H est à valeurs dans $\mathcal{S} = \mathbb{Z}/n\mathbb{Z}$.

1. Hacher $\boldsymbol{m}$, c'est-à-dire calculer $h := H(\boldsymbol{m})$.
2. Calculer et retourner $s = h^d \mod n$.

Signature RSA-FDH : $\mathsf{Verif}(\boldsymbol{m}, s, \texttt{pk})$

1. Calculer $h' = s^e \mod n$.
2. Hacher $\boldsymbol{m}$, c'est-à-dire calculer $h := H(\boldsymbol{m})$
3. Faire le test $h' = h$ et retourner le booléen associé.

Validité. $h' \equiv s^e \equiv h^{ed} \equiv h \mod n$.

Théorème. Dans le modèle de l'oracle aléatoire, si le problème RSA est difficile, alors la signature RSA-FDH est EUF-CMA.

Remarque. Le modèle de l'**oracle aléatoire** permet d'idéaliser les fonctions de hachage. C'est une hypothèse plus forte que le **modèle standard**.

En pratique, on peut utiliser la fonction de hachage SHA-3, de sortie 224 ou 256 bits par exemple.

Pour que le résultat de sécurité ci-dessus soit utilisable, il faut que l'espace de définition de la fonction RSA soit le même que l'espace des hachés : « *full domain hash* ».

Problème : pour RSA, il faut choisir n de 2048 bits minimum.

Solutions : il y en a beaucoup. *Padding*, concaténation de hachés successifs $H^{(i)}(\boldsymbol{m})$ ou de hachés avec incrément $H^{(i)}(\boldsymbol{m} \,\|\, \mathsf{ctr})$. Exemple :

$$FDH(\boldsymbol{m}, \mathsf{IV}) = H(\boldsymbol{m} \,\|\, n \,\|\, \mathsf{IV} + 0) \,\|\, H(\boldsymbol{m} \,\|\, n \,\|\, \mathsf{IV} + 1) \,\|\, H(\boldsymbol{m} \,\|\, n \,\|\, \mathsf{IV} + 2) \,\|\, \cdots$$

où IV est un vecteur d'initialisation public apposé au message.

Performance

RSA-FDH est une brique de base pour le standard RSA PKCS#1 v2.1.

Performances :

- Calcul efficace : un haché + $O(1)$ exponentiations modulaires
- Taille de clefs : clé publique $2\log_2 n \simeq 4096$ bits minimum ; clé privée $\log_2 d \simeq 2048$ bits minimum
- Taille de signature : $\log_2 n = 2048$ bits minimum

Plan

Description

On se place dans le groupe multiplicatif d'un corps fini $\mathbb{F}_p$, où p est premier. On note g un générateur de $\mathbb{F}_p^\times$.

Signature ElGamal : KeyGen

1. Choisir aléatoirement $a \in \mathbb{F}_p^\times$.
2. Calculer $\alpha = g^a \mod p$.
3. La clé publique est $\texttt{pk} = \alpha$, la clé privée est $\texttt{sk} = a$.

L'espace des messages est $\mathcal{M} = \mathbb{F}_p^\times$ et celui des signatures est $\mathcal{S} = \mathbb{F}_p^\times \times \mathbb{Z}/(p-1)\mathbb{Z}$.

Signature ElGamal : Sign$(m, \texttt{sk})$

1. Choisir aléatoirement $k \in (\mathbb{Z}/(p-1)\mathbb{Z})^\times$ (c'est-à-dire, iversible modulo $p-1$).
2. Calculer $b = g^k \mod p$.
3. Calculer $c = (m - ab)k^{-1} \mod (p-1)$.
4. Retourner $s = (b, c)$.

Signature ElGamal : Verif$(m, s, \texttt{pk})$

1. Calculer $x = \alpha^b \cdot b^c \mod p$ et $y = g^m \mod p$.
2. Faire le test $x = y$ et retourner le booléen associé.

Validité

Résumé.

- Clés : $\texttt{pk} = \alpha = g^a$, $\texttt{sk} = a$,
- Signature : $s = (b, c)$ où $b = g^k \mod p$ et $c = (m - ab)k^{-1} \mod (p-1)$,
- Vérification : $\alpha^b \cdot b^c \stackrel{?}{\equiv} g^m \mod p$

Remarque. Le premier élément b de la signature s ne dépend pas du message.

Validité. On vérifie que

$$\alpha^b \cdot b^c = g^{ab+k(m-ab)k^{-1} \mod (p-1)} \equiv g^m \mod p$$

Exemple

Paramètres. On prend de petites tailles pour l'exemple : $p = 467$, $g = 2$.

Génération de clefs. Alice engendre la clef privée $a = 127$; la clef publique est donc $\alpha = g^a = 2^{127} \mod 467 \equiv 132$.

Signature. Supposons qu'Alice veuille signer le message $m = 100$.

Elle choisit la valeur aléatoire $k = 213$. On vérifie bien que

$\text{pgcd}(k, p-1) = \text{pgcd}(213, 466) = 1$, et

$k^{-1} \mod (p-1)$ vaut alors 431.

Alice calcule alors

$$b = g^k = 2^{213} \equiv 29 \mod 467$$

$$c = (m - ab)k^{-1} = (100 - 127 \times 29) \times 431 \equiv 51 \mod 466$$

Vérification. On peut alors publiquement vérifier la signature $(29, 51)$ d'Alice pour le message $m = 100$:

d'une part, $\alpha^b \cdot b^c = 132^{29} \cdot 29^{51} \equiv 189 \mod 467$,

d'autre part, $g^m = 2^{100} \equiv 189 \mod 467$.

On va montrer que la signature ElGamal n'est pas EUF-KOA.

But. À partir de la clé publique uniquement, calculer m et $s = (b, c)$ tel que $\mathsf{Verif}(m, s, \mathtt{pk}) = \mathtt{true}$.

Idée : on va écrire $b = g^i \alpha^j$ pour $i, j \in \{0, \dots, p-2\}$. Cette écriture **n'est pas unique**.
Dans ce cas la condition de vérification est :

$$\alpha^b (g^i \alpha^j)^c = g^m \quad \Longleftrightarrow \quad \alpha^{b+jc} = g^{m-ic}$$

Cette condition est vérifiée **en particulier** si on a :

$$b + jc \equiv 0 \mod (p-1) \quad \text{et} \quad m - ic \equiv 0 \mod (p-1)$$

L'idée est alors de construire d'abord i quelconque et j inversible modulo $p-1$ quelconque, puis de définir b, c et m en fonction :

$$\begin{cases} b &= g^i \alpha^j & \mod p \\ c &= -bj^{-1} & \mod (p-1) \\ m &= -ic & \mod (p-1) \end{cases}$$

Remarque. Par l'utilisation d'une fonction de hachage, ces menaces peuvent être levées (voir cours suivant).

Plan

Exercice 1 du TD

Exercice 1. Bob veut envoyer plusieurs messages $(m_1, \ldots, m_\ell)$ à Alice, en utilisant le schéma de chiffrement d'ElGamal. On se place dans un groupe $\mathbb{G}$ d'ordre n, et la clef publique d'Alice est $\alpha \in \mathbb{G}$.

Question 1. Pour économiser des calculs, Bob utilise le même élément aléatoire $k \in \mathbb{Z}/n\mathbb{Z}$ pour chiffrer tous les messages m_i. Expliquer en quoi ce choix fragilise le système.

On suppose maintenant que Bob engendre un nouvel aléa pour chiffrer chaque message m_i. Pour cela, il utilise un générateur d'aléa, mais cet aléa est biaisé. En effet, la i-ème valeur $k_i \in \mathbb{Z}/n\mathbb{Z}$ engendrée par le générateur satisfait la relation

$$k_i = k_{i-1} + r \mod n\,,$$

où $r \in \mathbb{Z}/n\mathbb{Z}$ est est un élément connu de tous.

Question 2. Dans ce contexte, quelle est la relation entre les chiffrés de m_1, m_2, …, m_ℓ ?

Question 3. Supposons qu'un attaquant connaisse m_1. Démontrer qu'il peut alors retrouver tous les m_i.

Exercice 1 – question 1

Question 1. Pour économiser des calculs, Bob utilise le même élément aléatoire $k \in \mathbb{Z}/n\mathbb{Z}$ pour chiffrer tous les messages m_i. Expliquer en quoi ce choix fragilise le système.

Réponse. Le chiffré c_i du message m_i vaut $c_i = (g^k, m_i \alpha^k)$ pour tout i.

Le système n'est plus sûr car :

- Premièrement, le premier élément du chiffré ne dépend pas de i (il est constant égal à g^k). On peut donc distinguer une série de chiffrés d'une série d'élements aléatoires de $\mathbb{G} \times \mathbb{G}$. Le schéma de chiffrement n'est donc pas indistinguable, quel que soit le mode d'attaque.
- Deuxièmement, si l'on connaît l'un des m_i, alors on peut retrouver tous les autres m_j, pour $j \neq i$. En effet, le second terme de c_i permet de retrouver $\alpha^k = c_i[2]/m_i$, puis de calculer $m_j = c_j[2]/\alpha^k$.

Exercice 1 – question 2

On suppose maintenant que Bob engendre un nouvel aléa pour chiffrer chaque message m_i. Pour cela, il utilise un générateur d'aléa, mais cet aléa est biaisé. En effet, la i-ème valeur $k_i \in \mathbb{Z}/n\mathbb{Z}$ engendrée par le générateur satisfait la relation

$$k_i = k_{i-1} + r \mod n\,,$$

où $r \in \mathbb{Z}/n\mathbb{Z}$ est est un élément connu de tous.

Question 2. Dans ce contexte, quelle est la relation entre les chiffrés de $m_1, m_2, \ldots, m_\ell$?

Réponse. Notons $c_i = (\beta_{1,i}, \beta_{2,i})$ le chiffré de m_i. On a alors :

$$\beta_{1,i} = \gamma^{k_i} = \gamma^{k_1 + r(i-1)} \quad \text{et} \quad \beta_{2,i} = m_i \alpha^{k_i} = m_i \alpha^{k_1 + r(i-1)}$$

Si m_1 est inversible dans $\mathbb{Z}/n\mathbb{Z}$ (ce qui est le cas avec grande probabilité), on obtient

$$\beta_{1,i} = \beta_{1,1} \gamma^{r(i-1)} \quad \text{et} \quad \beta_{2,i} = m_1^{-1} \beta_{2,1} m_i \alpha^{r(i-1)}$$

Exercice 1 – question 3

Question 3. Supposons qu'un attaquant connaisse m_1. Démontrer qu'il peut alors retrouver tous les m_i.

Réponse. On utilise :

$$\beta_{2,i} = m_1^{-1} \beta_{2,1} m_i \alpha^{r(i-1)}$$

Comme on connaît m_1 et r, la seule inconnue ici est m_i, que l'on retrouve aisément :

$$m_i = m_1 \frac{\beta_{2,i}}{\beta_{2,1}} \alpha^{-r(i-1)}$$

Exercice 3 du TD

Exercice 3. On rappelle que le problème de Diffie-Hellman calculatoire (CDH) dans un groupe cyclique $\mathbb{G}$ d'ordre n est le suivant.

Problème CDH :

- **Instance.** g générateur de $\mathbb{G}$, et $a, b \in \mathbb{Z}/n\mathbb{Z}$.
- **Objectif.** Étant donné (g, g^a, g^b), calculer g^{ab}.

Pour simplifier, **on suppose que n est un nombre premier**.

Question 1. Démontrer que le problème CDH est équivalent au problème suivant :

Problème sqDH :

- **Instance.** g générateur de $\mathbb{G}$, et $x \in \mathbb{Z}/n\mathbb{Z}$.
- **Objectif.** Étant donné (g, g^x), calculer g^{x^2}.

Question 2. Démontrer que le problème CDH est équivalent au problème suivant :

Problème invDH :

- **Instance.** g générateur de $\mathbb{G}$, et $x \in \mathbb{Z}/n\mathbb{Z}$.
- **Objectif.** Étant donné (g, g^x), calculer $g^{x^{-1} \bmod n}$.

Pour ce faire, on pourra s'aider de l'équivalence démontrée à la question précédente.

Ex3 – question 1

Exercice 3. Pour simplifier, **on suppose que n est un nombre premier**.

Question 1. Démontrer que le problème CDH est équivalent au problème suivant :

Problème sqDH :

- **Instance.** g générateur de $\mathbb{G}$, et $x \in \mathbb{Z}/n\mathbb{Z}$.
- **Objectif.** Étant donné (g, g^x), calculer g^{x^2}.

Réponse. Déjà fait au TD du cours 1. Rappel ?

Pour sqDH $\to$ CDH (l'autre réduction est triviale), on peut donc calculer :

- g^{a^2} grâce à $\mathcal{O}(g, g^a)$
- g^{b^2} grâce à $\mathcal{O}(g, g^b)$
- $g^{(a+b)^2}$ grâce à $\mathcal{O}(g, g^a g^b)$

On obtient alors

$$\frac{g^{(a+b)^2}}{g^{a^2} g^{b^2}} = g^{a^2+b^2+2ab-a^2-b^2} = g^{2ab}.$$

Il reste à extraire une racine carrée de g^{2ab}. Ici, comme r est impair, on peut calculer $m = 2^{-1} \mod r$ (Euclide), puis g^m (exponentiation rapide).

Ex3 – question 2

Exercice 3. Pour simplifier, **on suppose que n est un nombre premier**.

Question 2. Démontrer que le problème CDH est équivalent au problème suivant :

Problème invDH :

- **Instance.** g générateur de $\mathbb{G}$, et $x \in \mathbb{Z}/n\mathbb{Z}$.
- **Objectif.** Étant donné (g, g^x), calculer $g^{x^{-1} \bmod n}$.

Pour ce faire, on pourra s'aider de l'équivalence démontrée à la question précédente.

Réponse (1). [invDH $<$ CDH] Soit A l'oracle qui résout CDH, et (g, g^x) une instance de invDH. Comme on suppose que n est un nombre **premier**, on a en particulier

$$g^{x^{-1} \bmod n} = g^{x^{n-2} \bmod n}.$$

Grâce à A, on sait calculer $(g^x, g^y) \mapsto g^{xy}$. On sait aussi calculer $g^x \mapsto g^{x^2}$ grâce à la question précédente.

Ainsi, on peut calculer $g^{x^{n-2} \bmod n}$ par une méthode de type exponentiation binaire (dans l'exposant).

On peut donc calculer $g^{x^{-1} \bmod n}$ avec un nombre logarithmique d'appels à l'oracle A.

Ex3 – question 2

Exercice 3. Pour simplifier, **on suppose que n est un nombre premier**.

Question 2. Démontrer que le problème CDH est équivalent au problème suivant :

Problème invDH :

- **Instance.** g générateur de $\mathbb{G}$, et $x \in \mathbb{Z}/n\mathbb{Z}$.
- **Objectif.** Étant donné (g, g^x), calculer $g^{x^{-1} \bmod n}$.

Pour ce faire, on pourra s'aider de l'équivalence démontrée à la question précédente.

Réponse (2). [CDH $<$ invDH] Soit B un oracle qui résout invDH. On cherche à résoudre une instance de sqDH, qui est équivalent à CDH.

Soit donc (g, g^x) ; on veut calculer g^{x^2}.

Idée (par exemple) :

$$\frac{1}{x} - \frac{1}{x+1} = \frac{1}{x^2+x}.$$

Donc, $x^{-1} - (x+1)^{-1} \equiv (x^2+x)^{-1} \mod n$. On obtient le résultat par la séquence de calculs :

- On calcule $g^{x+1} = g^x \times g$, puis $a = g^{x^{-1} \bmod n}$ et $b = g^{(x+1)^{-1} \bmod n}$ grâce à l'oracle B.
- On calcule $a/b = g^{(x^2+x)^{-1} \bmod n}$.
- On $c = a/b = g^{(x^2+x) \bmod n}$ grâce à B, puis $g^{x^2} = c/g^x$.

Exercice 5 du TD

Exercice 5. Soit p un nombre premier tel que $p \geq 5$. Soient a et b deux éléments de $\mathbb{F}_p$ tels que l'équation

$$x^3 + ax + b = 0$$

a trois solutions distinctes dans $\mathbb{F}_p$. On note E la courbe elliptique d'équation $y^2 = x^3 + ax + b$, et $E(\mathbb{F}_p)$ son groupe de points dans $\mathbb{F}_p$.

Question 1. Démontrer que les points $P \in E(\mathbb{F}_p)$ tel que $2P = \mathcal{O}$ forment un sous-groupe isomorphe à $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$.

Question 2. En déduire que dans le contexte de l'exercice, $E(\mathbb{F}_p)$ n'est pas cyclique.

Réponse 1. Combien d'éléments dans ce sous-groupe, usuellement noté $E(\mathbb{F}_p)[2]$? Si $P \neq \mathcal{O}$, on a :

$$2P = \mathcal{O} \iff P = -P \iff P = (x, 0)$$

Par hypothèse, il y a exactement trois solutions dans $\mathbb{F}_p$ à l'équation $x^3 + ax + b = 0$. On a donc trois points $P \neq \mathcal{O}$ dans ce sous-groupe, plus $\mathcal{O}$.

Le groupe est abélien (noté additivement), c'est donc ou bien $\mathbb{Z}/4\mathbb{Z}$ ou bien $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$.
Comment savoir ?

L'ordre de tout point P non-nul est 2 $\implies$ $E(\mathbb{F}_p)[2] \simeq \mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$.

Ce groupe est appelé **sous-groupe de 2-torsion** de $E(\mathbb{F}_p)$.

Réponse 2. $E(\mathbb{F}_p)$ admet un sous-groupe non-cyclique, il n'est donc pas cyclique.

Exercice 2 du TD

Résumé de l'exercice. Sur $\mathbb{F}_p^\times$, avec $p = 79$ et $g = 3$, attaquer (par force brute) la clef publique $\alpha = 36$ d'un système d'ElGamal, pour déchiffrer le message :

$$[77,76],[69,22],[50,33],[74,54],[76,10],[50,78],[67,29],[47,40],[74,55],[59,5],$$
$$[65,74],[41,49],[43,34],[48,8],[47,75],[5,59],[5,55],[15,38],[21,20],[78,31],$$
$$[24,26],[32,77],[35,73],[54,30],[25,10],[66,42],[78,53],[74,23],[19,18],[22,58],$$
$$[72,52],[19,54],[4,62],[6,49],[10,10],[52,30],[62,18],[43,4],[1,53],[50,37].$$

Pour l'encodage du message, voir feuille de TD.

Réponse. Voir programmes.

Fin du cours

Questions ?