

Cryptographie à clé publique

Cours 6

Julien Lavauzelle

Université Paris 8

Master 1 mathématiques et applications – parcours ACC

11/03/2026

1. Identification et signature

Schémas d'identification

Construction à partir de signature

Construction sans signature

De l'identification à la signature

2. Chiffrement basé sur l'identité

Définitions

IBE de Cocks

Pour aller plus loin : IBE de Boneh-Franklin

1. Identification et signature

Schémas d'identification

Construction à partir de signature

Construction sans signature

De l'identification à la signature

2. Chiffrement basé sur l'identité

Définitions

IBE de Cocks

Pour aller plus loin : IBE de Boneh-Franklin

Objectif d'un schéma d'identification : prouver à l'autre son identité.

Objectif d'un schéma d'identification : prouver à l'autre son identité.

Motivations :

- essentiellement pour du **droit d'accès** : à un réseau, à un site web, à des ressources, ou même à un lieu physique (passeport)

Objectif d'un schéma d'identification : prouver à l'autre son identité.

Motivations :

- essentiellement pour du **droit d'accès** : à un réseau, à un site web, à des ressources, ou même à un lieu physique (passeport)
- brique de base pour d'autres protocoles (ex : signature)

Objectif d'un schéma d'identification : prouver à l'autre son identité.

Motivations :

- essentiellement pour du **droit d'accès** : à un réseau, à un site web, à des ressources, ou même à un lieu physique (passeport)
- brique de base pour d'autres protocoles (ex : signature)

Essentiellement 3 manières de procéder à une **vérification d'identité**.

Objectif d'un schéma d'identification : prouver à l'autre son identité.

Motivations :

- essentiellement pour du **droit d'accès** : à un réseau, à un site web, à des ressources, ou même à un lieu physique (passeport)
- brique de base pour d'autres protocoles (ex : signature)

Essentiellement 3 manières de procéder à une **vérification d'identité**.

1. Par ce que l'on **est**.

→ Exemple : biométrie (empreintes digitales, reconnaissance faciale, etc.)

Objectif d'un schéma d'identification : prouver à l'autre son identité.

Motivations :

- essentiellement pour du **droit d'accès** : à un réseau, à un site web, à des ressources, ou même à un lieu physique (passeport)
- brique de base pour d'autres protocoles (ex : signature)

Essentiellement 3 manières de procéder à une **vérification d'identité**.

1. Par ce que l'on **est**.
→ Exemple : biométrie (empreintes digitales, reconnaissance faciale, etc.)
2. Par ce que l'on **possède**.
→ Exemple : documents d'identité, clés, etc.

Objectif d'un schéma d'identification : prouver à l'autre son identité.

Motivations :

- essentiellement pour du **droit d'accès** : à un réseau, à un site web, à des ressources, ou même à un lieu physique (passeport)
- brique de base pour d'autres protocoles (ex : signature)

Essentiellement 3 manières de procéder à une **vérification d'identité**.

1. Par ce que l'on **est**.
→ Exemple : biométrie (empreintes digitales, reconnaissance faciale, etc.)
2. Par ce que l'on **possède**.
→ Exemple : documents d'identité, clés, etc.
3. Par ce que l'on **sait**.
→ Exemple : mots de passe.

Définition. Un **schéma d'identification** implique un **prouveur** \mathcal{P} et un **vérifieur** \mathcal{V} .

Définition. Un **schéma d'identification** implique un **prouveur** \mathcal{P} et un **vérifieur** \mathcal{V} .
Il est constitué d'un **algorithme** de génération de clefs et d'un **protocole** de vérification.

Définition. Un **schéma d'identification** implique un **prouveur** \mathcal{P} et un **vérifieur** \mathcal{V} .

Il est constitué d'un **algorithme** de génération de clefs et d'un **protocole** de vérification.

- Lors de la **génération de clefs** KeyGen , le prouveur engendre une paire de clefs publique/privée et émet la clé publique.

Définition. Un **schéma d'identification** implique un **prouveur** \mathcal{P} et un **vérifieur** \mathcal{V} .

Il est constitué d'un **algorithme** de génération de clés et d'un **protocole** de vérification.

- Lors de la **génération de clés** KeyGen , le prouveur engendre une paire de clés publique/privée et émet la clé publique.
- Lors du **protocole de vérification** noté $[\mathcal{P} \longleftrightarrow \mathcal{V}]$, le vérifieur retourne un booléen $b \in \{\text{true}, \text{false}\}$ suivant s'il est convaincu ou non de l'identité du prouveur

Définition. Un **schéma d'identification** implique un **prouveur** \mathcal{P} et un **vérifieur** \mathcal{V} .

Il est constitué d'un **algorithme** de génération de clefs et d'un **protocole** de vérification.

- Lors de la **génération de clefs** KeyGen , le prouveur engendre une paire de clefs publique/privée et émet la clé publique.
- Lors du **protocole de vérification** noté $[\mathcal{P} \longleftrightarrow \mathcal{V}]$, le vérifieur retourne un booléen $b \in \{\text{true}, \text{false}\}$ suivant s'il est convaincu ou non de l'identité du prouveur

Remarque. Contrairement à la signature, on a ici un **protocole** de vérification, c'est-à-dire une échange de données **interactif**.

Définition. Un **schéma d'identification** implique un **prouveur** \mathcal{P} et un **vérifieur** \mathcal{V} .

Il est constitué d'un **algorithme** de génération de clefs et d'un **protocole** de vérification.

- Lors de la **génération de clefs** KeyGen , le prouveur engendre une paire de clefs publique/privée et émet la clé publique.
- Lors du **protocole de vérification** noté $[\mathcal{P} \longleftrightarrow \mathcal{V}]$, le vérifieur retourne un booléen $b \in \{\text{true}, \text{false}\}$ suivant s'il est convaincu ou non de l'identité du prouveur

Remarque. Contrairement à la signature, on a ici un **protocole** de vérification, c'est-à-dire une échange de données **interactif**.

Définition.

1. On dit qu'un prouveur est **honnête** s'il il possède la clé privée et s'il suit le protocole.

Définition. Un **schéma d'identification** implique un **prouveur** \mathcal{P} et un **vérifieur** \mathcal{V} .

Il est constitué d'un **algorithme** de génération de clefs et d'un **protocole** de vérification.

- Lors de la **génération de clefs** KeyGen , le prouveur engendre une paire de clefs publique/privée et émet la clé publique.
- Lors du **protocole de vérification** noté $[\mathcal{P} \longleftrightarrow \mathcal{V}]$, le vérifieur retourne un booléen $b \in \{\text{true}, \text{false}\}$ suivant s'il est convaincu ou non de l'identité du prouveur

Remarque. Contrairement à la signature, on a ici un **protocole** de vérification, c'est-à-dire une échange de données **interactif**.

Définition.

1. On dit qu'un prouveur est **honnête** s'il il possède la clé privée et s'il suit le protocole.
2. Un schéma d'identification est **valide** si tout prouveur honnête convainc le vérifieur de son identité avec probabilité 1.

Définition. Un **schéma d'identification** implique un **prouveur** \mathcal{P} et un **vérifieur** \mathcal{V} .

Il est constitué d'un **algorithme** de génération de clefs et d'un **protocole** de vérification.

- Lors de la **génération de clefs** KeyGen , le prouveur engendre une paire de clefs publique/privée et émet la clé publique.
- Lors du **protocole de vérification** noté $[\mathcal{P} \longleftrightarrow \mathcal{V}]$, le vérifieur retourne un booléen $b \in \{\text{true}, \text{false}\}$ suivant s'il est convaincu ou non de l'identité du prouveur

Remarque. Contrairement à la signature, on a ici un **protocole** de vérification, c'est-à-dire une échange de données **interactif**.

Définition.

1. On dit qu'un prouveur est **honnête** s'il il possède la clé privée et s'il suit le protocole.
2. Un schéma d'identification est **valide** si tout prouveur honnête convainc le vérifieur de son identité avec probabilité 1.

Sauf si le contraire est indiqué : dans ces slides, Alice = prouveur et Bob = vérifieur.

Pour la **sécurité**, le but d'une attaque est l'**imposture**, *i.e.* se faire passer pour le prouveur \mathcal{P} auprès d'un vérifieur.

Pour la **sécurité**, le but d'une attaque est l'**imposture**, *i.e.* se faire passer pour le prouveur \mathcal{P} auprès d'un vérifieur.

Il y a différents modèles d'attaques, mais tous se caractérisent par deux étapes.

Pour la **sécurité**, le but d'une attaque est l'**imposture**, *i.e.* se faire passer pour le prouveur \mathcal{P} auprès d'un vérifieur.

Il y a différents modèles d'attaques, mais tous se caractérisent par deux étapes.

1. **Étape d'apprentissage.** L'attaquant observe des échanges entre un prouveur \mathcal{P} et un vérifieur \mathcal{V} lors de plusieurs itérations du protocole de vérification $[\mathcal{P} \longleftrightarrow \mathcal{V}]$.

Pour la **sécurité**, le but d'une attaque est l'**imposture**, *i.e.* se faire passer pour le prouveur \mathcal{P} auprès d'un vérifieur.

Il y a différents modèles d'attaques, mais tous se caractérisent par deux étapes.

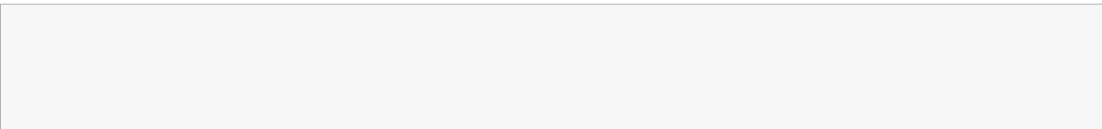
1. **Étape d'apprentissage.** L'attaquant observe des échanges entre un prouveur \mathcal{P} et un vérifieur \mathcal{V} lors de plusieurs itérations du protocole de vérification $[\mathcal{P} \longleftrightarrow \mathcal{V}]$.
2. **Étape d'imposture.** L'attaquant produit une simulation de prouveur $\tilde{\mathcal{P}}$. L'attaque est réussie si, lors d'une exécution de $[\tilde{\mathcal{P}} \longleftrightarrow \mathcal{V}']$ avec un vérifieur \mathcal{V}' , la probabilité que le vérifieur \mathcal{V}' soit convaincu d'interagir avec \mathcal{P} est non-négligeable.

Pour la **sécurité**, le but d'une attaque est l'**imposture**, *i.e.* se faire passer pour le prouveur \mathcal{P} auprès d'un vérifieur.

Il y a différents modèles d'attaques, mais tous se caractérisent par deux étapes.

1. **Étape d'apprentissage.** L'attaquant observe des échanges entre un prouveur \mathcal{P} et un vérifieur \mathcal{V} lors de plusieurs itérations du protocole de vérification $[\mathcal{P} \longleftrightarrow \mathcal{V}]$.
2. **Étape d'imposture.** L'attaquant produit une simulation de prouveur $\tilde{\mathcal{P}}$. L'attaque est réussie si, lors d'une exécution de $[\tilde{\mathcal{P}} \longleftrightarrow \mathcal{V}']$ avec un vérifieur \mathcal{V}' , la probabilité que le vérifieur \mathcal{V}' soit convaincu d'interagir avec \mathcal{P} est non-négligeable.

Selon l'activité de l'attaquant dans l'**étape d'apprentissage**, on différencie deux moyens d'attaque :



Pour la **sécurité**, le but d'une attaque est l'**imposture**, *i.e.* se faire passer pour le prouveur \mathcal{P} auprès d'un vérifieur.

Il y a différents modèles d'attaques, mais tous se caractérisent par deux étapes.

1. **Étape d'apprentissage.** L'attaquant observe des échanges entre un prouveur \mathcal{P} et un vérifieur \mathcal{V} lors de plusieurs itérations du protocole de vérification $[\mathcal{P} \longleftrightarrow \mathcal{V}]$.
2. **Étape d'imposture.** L'attaquant produit une simulation de prouveur $\tilde{\mathcal{P}}$. L'attaque est réussie si, lors d'une exécution de $[\tilde{\mathcal{P}} \longleftrightarrow \mathcal{V}']$ avec un vérifieur \mathcal{V}' , la probabilité que le vérifieur \mathcal{V}' soit convaincu d'interagir avec \mathcal{P} est non-négligeable.

Selon l'activité de l'attaquant dans l'**étape d'apprentissage**, on différencie deux moyens d'attaque :

- Les attaques **passives** sont celles où l'attaquant ne fait qu'observer les échanges entre le prouveur et un vérifieur externe.

Pour la **sécurité**, le but d'une attaque est l'**imposture**, *i.e.* se faire passer pour le prouveur \mathcal{P} auprès d'un vérifieur.

Il y a différents modèles d'attaques, mais tous se caractérisent par deux étapes.

1. **Étape d'apprentissage.** L'attaquant observe des échanges entre un prouveur \mathcal{P} et un vérifieur \mathcal{V} lors de plusieurs itérations du protocole de vérification $[\mathcal{P} \longleftrightarrow \mathcal{V}]$.
2. **Étape d'imposture.** L'attaquant produit une simulation de prouveur $\tilde{\mathcal{P}}$. L'attaque est réussie si, lors d'une exécution de $[\tilde{\mathcal{P}} \longleftrightarrow \mathcal{V}']$ avec un vérifieur \mathcal{V}' , la probabilité que le vérifieur \mathcal{V}' soit convaincu d'interagir avec \mathcal{P} est non-négligeable.

Selon l'activité de l'attaquant dans l'**étape d'apprentissage**, on différencie deux moyens d'attaque :

- Les attaques **passives** sont celles où l'attaquant ne fait qu'observer les échanges entre le prouveur et un vérifieur externe.
- Les attaques **actives** sont celles où l'attaquant est également autorisé à jouer le rôle d'un vérifieur (il va donc « conduire » les exécutions du protocole de vérification).

1. Identification et signature

Schémas d'identification

Construction à partir de signature

Construction sans signature

De l'identification à la signature

2. Chiffrement basé sur l'identité

Définitions

IBE de Cocks

Pour aller plus loin : IBE de Boneh-Franklin

Théorème (informel). Il est possible de construire un schéma d'identification sûr contre des adversaires actifs, **à partir** d'une signature numérique EUF-CMA.

Théorème (informel). Il est possible de construire un schéma d'identification sûr contre des adversaires actifs, **à partir** d'une signature numérique EUF-CMA.

Essayons!

Théorème (informel). Il est possible de construire un schéma d'identification sûr contre des adversaires actifs, **à partir** d'une signature numérique EUF-CMA.

Essayons!

Mise en place. On suppose que Alice est munie d'une paire de clefs (pk_A, sk_A) certifiée, pour un schéma de signature numérique quelconque.

Théorème (informel). Il est possible de construire un schéma d'identification sûr contre des adversaires actifs, **à partir** d'une signature numérique EUF-CMA.

Essayons!

Mise en place. On suppose que Alice est munie d'une paire de clefs (pk_A, sk_A) certifiée, pour un schéma de signature numérique quelconque.

On associe également à Alice son identité id_A .

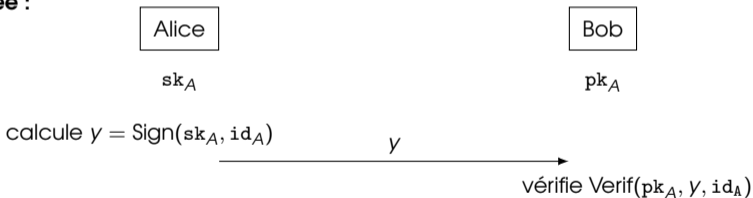
Théorème (informel). Il est possible de construire un schéma d'identification sûr contre des adversaires actifs, à partir d'une signature numérique EUF-CMA.

Essayons!

Mise en place. On suppose que Alice est munie d'une paire de clefs (pk_A, sk_A) certifiée, pour un schéma de signature numérique quelconque.

On associe également à Alice son identité id_A .

Une première idée :



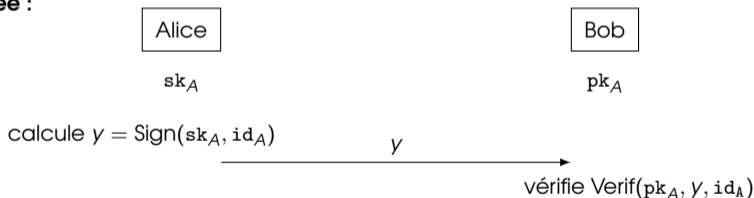
Théorème (informel). Il est possible de construire un schéma d'identification sûr contre des adversaires actifs, à partir d'une signature numérique EUF-CMA.

Essayons!

Mise en place. On suppose que Alice est munie d'une paire de clefs (pk_A, sk_A) certifiée, pour un schéma de signature numérique quelconque.

On associe également à Alice son identité id_A .

Une première idée :



Problème. Attaque par rejeu.

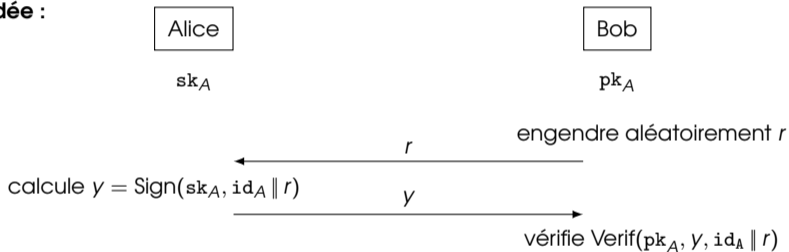
→ Oscar, qui observe le canal de transmission, peut réutiliser ailleurs la signature y , afin d'être identifié comme Alice.

Mise en place. On suppose que Alice est munie d'une paire de clefs (pk_A, sk_A) certifiée, pour un schéma de signature numérique quelconque. On associe également à Alice son identité id_A .

Identification à partir d'une signature

Mise en place. On suppose que Alice est munie d'une paire de clefs (pk_A, sk_A) certifiée, pour un schéma de signature numérique quelconque. On associe également à Alice son identité id_A .

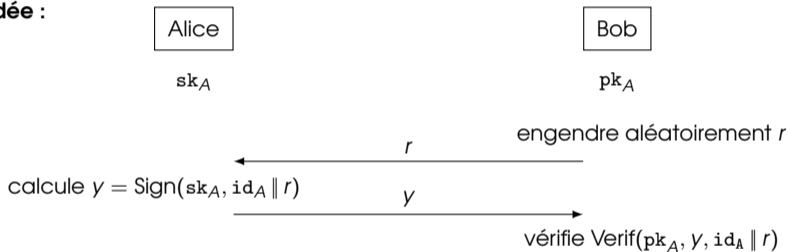
Une seconde idée :



Identification à partir d'une signature

Mise en place. On suppose que Alice est munie d'une paire de clefs (pk_A, sk_A) certifiée, pour un schéma de signature numérique quelconque. On associe également à Alice son identité id_A .

Une seconde idée :

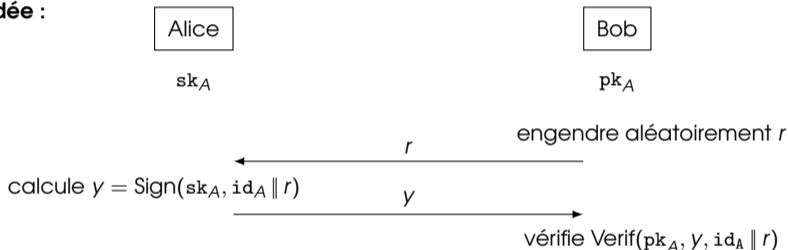


La valeur de y ne peut pas être réutilisée, car elle dépend du "challenge" r engendré par Bob.

Identification à partir d'une signature

Mise en place. On suppose que Alice est munie d'une paire de clefs (pk_A, sk_A) certifiée, pour un schéma de signature numérique quelconque. On associe également à Alice son identité id_A .

Une seconde idée :

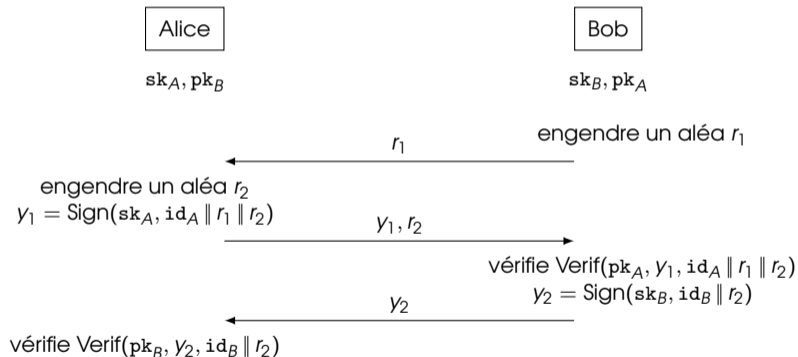


La valeur de y ne peut pas être réutilisée, car elle dépend du "challenge" r engendré par Bob.

Théorème (informel). Il est possible de construire un schéma d'identification sûr contre des adversaires actifs, **à partir** d'une signature numérique EUF-CMA.

Remarque. On peut également construire un protocole d'identification **mutuelle**, où Alice et Bob sont mutuellement convaincus de leurs identités réciproques.

Remarque. On peut également construire un protocole d'identification **mutuelle**, où Alice et Bob sont mutuellement convaincus de leurs identités réciproques.



1. Identification et signature

Schémas d'identification

Construction à partir de signature

Construction sans signature

De l'identification à la signature

2. Chiffrement basé sur l'identité

Définitions

IBE de Cocks

Pour aller plus loin : IBE de Boneh-Franklin

But : construire un schéma d'identification sans signature préalable.

But : construire un schéma d'identification sans signature préalable.

On suppose qu'Alice a émis une clé publique $\alpha = g^{-a}$ qui est **certifiée**, où

- $g \in \mathbb{F}_p^\times$ est d'ordre premier q dans \mathbb{F}_p^\times
- $a \in \{0, \dots, q-1\}$ est gardé secrètement par Alice.

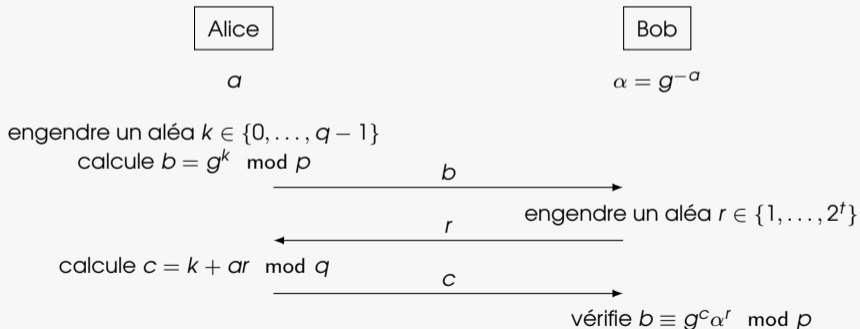
Schéma d'identification de Schnorr

But : construire un schéma d'identification sans signature préalable.

On suppose qu'Alice a émis une clé publique $\alpha = g^{-a}$ qui est **certifiée**, où

- $g \in \mathbb{F}_p$ est d'ordre premier q dans \mathbb{F}_p^\times
- $a \in \{0, \dots, q-1\}$ est gardé secrètement par Alice.

SCHÉMA D'IDENTIFICATION DE SCHNORR



Validité. Si le protocole est respecté, alors on a bien

$$g^{c\alpha^r} \equiv g^{k+ar} g^{-ar} \equiv g^k \equiv b \pmod{p}.$$

Validité. Si le protocole est respecté, alors on a bien

$$g^{c\alpha^r} \equiv g^{k+ar} g^{-ar} \equiv g^k \equiv b \pmod{p}.$$

Sécurité. On peut démontrer le résultat suivant.

Validité. Si le protocole est respecté, alors on a bien

$$g^{c\alpha^r} \equiv g^{k+ar} g^{-ar} \equiv g^k \equiv b \pmod{p}.$$

Sécurité. On peut démontrer le résultat suivant.

Théorème (informel). S'il existe un algorithme polynomial qui réalise une imposture sur le schéma d'identification de Schnorr avec probabilité $\varepsilon > 2^{-t+1}$, alors il existe un algorithme probabiliste qui résout le logarithme discret avec complexité $O(1/\varepsilon)$.

Validité. Si le protocole est respecté, alors on a bien

$$g^{c\alpha^r} \equiv g^{k+ar} g^{-ar} \equiv g^k \equiv b \pmod{p}.$$

Sécurité. On peut démontrer le résultat suivant.

Théorème (informel). S'il existe un algorithme polynomial qui réalise une imposture sur le schéma d'identification de Schnorr avec probabilité $\varepsilon > 2^{-t+1}$, alors il existe un algorithme probabiliste qui résout le logarithme discret avec complexité $O(1/\varepsilon)$.

Ce type de schéma d'identification est appelé « **identification en trois passes** ».

Validité. Si le protocole est respecté, alors on a bien

$$g^{c\alpha^r} \equiv g^{k+ar} g^{-ar} \equiv g^k \equiv b \pmod{p}.$$

Sécurité. On peut démontrer le résultat suivant.

Théorème (informel). S'il existe un algorithme polynomial qui réalise une imposture sur le schéma d'identification de Schnorr avec probabilité $\varepsilon > 2^{-t+1}$, alors il existe un algorithme probabiliste qui résout le logarithme discret avec complexité $O(1/\varepsilon)$.

Ce type de schéma d'identification est appelé « **identification en trois passes** ».

1. Phase d'**engagement** (*commitment*) : Alice s'engage sur une valeur aléatoire (k) qu'elle transmet de manière cachée ($g^k \pmod{p}$) à Bob.

Validité. Si le protocole est respecté, alors on a bien

$$g^{c\alpha^r} \equiv g^{k+ar} g^{-ar} \equiv g^k \equiv b \pmod{p}.$$

Sécurité. On peut démontrer le résultat suivant.

Théorème (informel). S'il existe un algorithme polynomial qui réalise une imposture sur le schéma d'identification de Schnorr avec probabilité $\varepsilon > 2^{-t+1}$, alors il existe un algorithme probabiliste qui résout le logarithme discret avec complexité $O(1/\varepsilon)$.

Ce type de schéma d'identification est appelé « **identification en trois passes** ».

1. Phase d'**engagement** (*commitment*) : Alice s'engage sur une valeur aléatoire (k) qu'elle transmet de manière cachée ($g^k \pmod{p}$) à Bob.
2. Phase de **défi/challenge** : Bob construit un défi aléatoire (r) auquel Alice doit répondre. La résolution de ce défi dépend de l'engagement d'Alice.

Validité. Si le protocole est respecté, alors on a bien

$$g^{c\alpha^r} \equiv g^{k+ar} g^{-ar} \equiv g^k \equiv b \pmod{p}.$$

Sécurité. On peut démontrer le résultat suivant.

Théorème (informel). S'il existe un algorithme polynomial qui réalise une imposture sur le schéma d'identification de Schnorr avec probabilité $\varepsilon > 2^{-t+1}$, alors il existe un algorithme probabiliste qui résout le logarithme discret avec complexité $O(1/\varepsilon)$.

Ce type de schéma d'identification est appelé « **identification en trois passes** ».

1. Phase d'**engagement** (*commitment*) : Alice s'engage sur une valeur aléatoire (k) qu'elle transmet de manière cachée ($g^k \pmod{p}$) à Bob.
2. Phase de **défi/challenge** : Bob construit un défi aléatoire (r) auquel Alice doit répondre. La résolution de ce défi dépend de l'engagement d'Alice.
3. Phase de **réponse** : Alice répond au défi aléatoire de Bob, qui vérifie ensuite si la réponse est correcte.

Validité. Si le protocole est respecté, alors on a bien

$$g^{c\alpha^r} \equiv g^{k+ar} g^{-ar} \equiv g^k \equiv b \pmod{p}.$$

Sécurité. On peut démontrer le résultat suivant.

Théorème (informel). S'il existe un algorithme polynomial qui réalise une imposture sur le schéma d'identification de Schnorr avec probabilité $\varepsilon > 2^{-t+1}$, alors il existe un algorithme probabiliste qui résout le logarithme discret avec complexité $O(1/\varepsilon)$.

Ce type de schéma d'identification est appelé « **identification en trois passes** ».

1. Phase d'**engagement** (*commitment*) : Alice s'engage sur une valeur aléatoire (k) qu'elle transmet de manière cachée ($g^k \pmod{p}$) à Bob.
2. Phase de **défi/challenge** : Bob construit un défi aléatoire (r) auquel Alice doit répondre. La résolution de ce défi dépend de l'engagement d'Alice.
3. Phase de **réponse** : Alice répond au défi aléatoire de Bob, qui vérifie ensuite si la réponse est correcte.

Remarque. Il existe des schémas d'identification à 5 passes, comportant deux challenges et deux réponses.

Un **exemple-jouet** (petites valeurs) pour le schéma d'identification de Schnorr.

Un exemple-jouet (petites valeurs) pour le schéma d'identification de Schnorr.

Paramètres. Soit $p = 88667$ et $q = 1031$. On fixe $t = 10$ (taille des défis). Un sous-groupe de \mathbb{F}_p^\times d'ordre q est engendré par $g = 70322$ par exemple.

Un exemple-jouet (petites valeurs) pour le schéma d'identification de Schnorr.

Paramètres. Soit $p = 88667$ et $q = 1031$. On fixe $t = 10$ (taille des défis). Un sous-groupe de \mathbb{F}_p^\times d'ordre q est engendré par $g = 70322$ par exemple.

Clés. La clé privée d'Alice est $a = 755$. La clé publique correspondante est :

$$\alpha = g^{-a} = 70322^{1031-755} \equiv 13136 \pmod{88667}$$

Un exemple-jouet (petites valeurs) pour le schéma d'identification de Schnorr.

Paramètres. Soit $p = 88667$ et $q = 1031$. On fixe $t = 10$ (taille des défis). Un sous-groupe de \mathbb{F}_p^\times d'ordre q est engendré par $g = 70322$ par exemple.

Clés. La clé privée d'Alice est $a = 755$. La clé publique correspondante est :

$$\alpha = g^{-a} = 70322^{1031-755} \equiv 13136 \pmod{88667}$$

Engagement. Alice choisit $k = 543$ et envoie

$$b = g^k \equiv 70322^{543} \equiv 84109 \pmod{88667}$$

Un exemple-jouet (petites valeurs) pour le schéma d'identification de Schnorr.

Paramètres. Soit $p = 88667$ et $q = 1031$. On fixe $t = 10$ (taille des défis). Un sous-groupe de \mathbb{F}_p^\times d'ordre q est engendré par $g = 70322$ par exemple.

Clés. La clé privée d'Alice est $a = 755$. La clé publique correspondante est :

$$\alpha = g^{-a} = 70322^{1031-755} \equiv 13136 \pmod{88667}$$

Engagement. Alice choisit $k = 543$ et envoie

$$b = g^k \equiv 70322^{543} \equiv 84109 \pmod{88667}$$

Défi. Bob envoie le défi $r = 1000 < 2^t$.

Un exemple-jouet (petites valeurs) pour le schéma d'identification de Schnorr.

Paramètres. Soit $p = 88667$ et $q = 1031$. On fixe $t = 10$ (taille des défis). Un sous-groupe de \mathbb{F}_p^\times d'ordre q est engendré par $g = 70322$ par exemple.

Clés. La clé privée d'Alice est $a = 755$. La clé publique correspondante est :

$$\alpha = g^{-a} = 70322^{1031-755} \equiv 13136 \pmod{88667}$$

Engagement. Alice choisit $k = 543$ et envoie

$$b = g^k \equiv 70322^{543} \equiv 84109 \pmod{88667}$$

Défi. Bob envoie le défi $r = 1000 < 2^t$.

Réponse. Alice calcule

$$c = k + ar = 543 + 755 \times 1000 \equiv 851 \pmod{1031}$$

Un exemple-jouet (petites valeurs) pour le schéma d'identification de Schnorr.

Paramètres. Soit $p = 88667$ et $q = 1031$. On fixe $t = 10$ (taille des défis). Un sous-groupe de \mathbb{F}_p^\times d'ordre q est engendré par $g = 70322$ par exemple.

Clés. La clé privée d'Alice est $a = 755$. La clé publique correspondante est :

$$\alpha = g^{-a} = 70322^{1031-755} \equiv 13136 \pmod{88667}$$

Engagement. Alice choisit $k = 543$ et envoie

$$b = g^k \equiv 70322^{543} \equiv 84109 \pmod{88667}$$

Défi. Bob envoie le défi $r = 1000 < 2^t$.

Réponse. Alice calcule

$$c = k + ar = 543 + 755 \times 1000 \equiv 851 \pmod{1031}$$

Vérification. Bob vérifie que $b \equiv g^c \alpha^r \pmod{p}$, c'est-à-dire

$$84109 \equiv 70322^{851} 13136^{1000} \pmod{88667}.$$

1. Identification et signature

Schémas d'identification

Construction à partir de signature

Construction sans signature

De l'identification à la signature

2. Chiffrement basé sur l'identité

Définitions

IBE de Cocks

Pour aller plus loin : IBE de Boneh-Franklin

La **transformation** (ou **heuristique**) de **Fiat–Shamir** permet de convertir un protocole de vérification interactif en une preuve non-interactive.

Informellement, l'idée est la suivante : l'aléa provenant du vérifieur est **remplacé** par l'utilisation d'une **fonction de hachage**.

La **transformation** (ou **heuristique**) de **Fiat–Shamir** permet de convertir un protocole de vérification interactif en une preuve non-interactive.

Informellement, l'idée est la suivante : l'aléa provenant du vérifieur est **remplacé** par l'utilisation d'une **fonction de hachage**.

Si le protocole initial est un **schéma d'identification**, alors la transformation de Fiat-Shamir permet d'obtenir un **schéma de signature numérique**.

La **transformation** (ou **heuristique**) de **Fiat–Shamir** permet de convertir un protocole de vérification interactif en une preuve non-interactive.

Informellement, l'idée est la suivante : l'aléa provenant du vérifieur est **remplacé** par l'utilisation d'une **fonction de hachage**.

Si le protocole initial est un **schéma d'identification**, alors la transformation de Fiat-Shamir permet d'obtenir un **schéma de signature numérique**.

Pour des protocoles en trois passes (engagement, défi, réponse), l'idée est la suivante :

La **transformation** (ou **heuristique**) de **Fiat–Shamir** permet de convertir un protocole de vérification interactif en une preuve non-interactive.

Informellement, l'idée est la suivante : l'aléa provenant du vérifieur est **remplacé** par l'utilisation d'une **fonction de hachage**.

Si le protocole initial est un **schéma d'identification**, alors la transformation de Fiat-Shamir permet d'obtenir un **schéma de signature numérique**.

Pour des protocoles en trois passes (engagement, défi, réponse), l'idée est la suivante :

- le prouveur s'engage sur un élément x

La **transformation** (ou **heuristique**) de **Fiat–Shamir** permet de convertir un protocole de vérification interactif en une preuve non-interactive.

Informellement, l'idée est la suivante : l'aléa provenant du vérifieur est **remplacé** par l'utilisation d'une **fonction de hachage**.

Si le protocole initial est un **schéma d'identification**, alors la transformation de Fiat-Shamir permet d'obtenir un **schéma de signature numérique**.

Pour des protocoles en trois passes (engagement, défi, réponse), l'idée est la suivante :

- le prouveur s'engage sur un élément x
- plutôt que de recevoir un défi aléatoire r du vérifieur, le prouveur hache x et le message m

La **transformation** (ou **heuristique**) de **Fiat–Shamir** permet de convertir un protocole de vérification interactif en une preuve non-interactive.

Informellement, l'idée est la suivante : l'aléa provenant du vérifieur est **remplacé** par l'utilisation d'une **fonction de hachage**.

Si le protocole initial est un **schéma d'identification**, alors la transformation de Fiat-Shamir permet d'obtenir un **schéma de signature numérique**.

Pour des protocoles en trois passes (engagement, défi, réponse), l'idée est la suivante :

- le prouveur s'engage sur un élément x
- plutôt que de recevoir un défi aléatoire r du vérifieur, le prouveur hache x et le message m
- le prouveur calcule ensuite une réponse c au défi $r = H(x, m)$, qui constitue alors la signature de m .

La **transformation** (ou **heuristique**) de **Fiat–Shamir** permet de convertir un protocole de vérification interactif en une preuve non-interactive.

Informellement, l'idée est la suivante : l'aléa provenant du vérifieur est **remplacé** par l'utilisation d'une **fonction de hachage**.

Si le protocole initial est un **schéma d'identification**, alors la transformation de Fiat-Shamir permet d'obtenir un **schéma de signature numérique**.

Pour des protocoles en trois passes (engagement, défi, réponse), l'idée est la suivante :

- le prouveur s'engage sur un élément x
- plutôt que de recevoir un défi aléatoire r du vérifieur, le prouveur hache x et le message m
- le prouveur calcule ensuite une réponse c au défi $r = H(x, m)$, qui constitue alors la signature de m .

Si l'on instancie cette idée avec le protocole d'identification de Schnorr, on obtient la **signature de Schnorr**.

La **transformation** (ou **heuristique**) de **Fiat–Shamir** permet de convertir un protocole de vérification interactif en une preuve non-interactive.

Informellement, l'idée est la suivante : l'aléa provenant du vérifieur est **remplacé** par l'utilisation d'une **fonction de hachage**.

Si le protocole initial est un **schéma d'identification**, alors la transformation de Fiat-Shamir permet d'obtenir un **schéma de signature numérique**.

Pour des protocoles en trois passes (engagement, défi, réponse), l'idée est la suivante :

- le prouveur s'engage sur un élément x
- plutôt que de recevoir un défi aléatoire r du vérifieur, le prouveur hache x et le message m
- le prouveur calcule ensuite une réponse c au défi $r = H(x, m)$, qui constitue alors la signature de m .

Si l'on instancie cette idée avec le protocole d'identification de Schnorr, on obtient la **signature de Schnorr**.

Remarque. Dans certains cas, la transformation de Fiat-Shamir permet également de préserver une propriété de **non-divulgateion** du protocole interactif.

SIGNATURE DE SCHNORR : KeyGen

1. Choisir a aléatoirement dans $\mathbb{Z}/q\mathbb{Z}^\times$.
2. Calculer $\alpha = g^{-a}$.
3. La clé publique est $\mathbf{pk} = \alpha$, la clé privée est $\mathbf{sk} = a$.

SIGNATURE DE SCHNORR : KeyGen

1. Choisir a aléatoirement dans $\mathbb{Z}/q\mathbb{Z}^\times$.
2. Calculer $\alpha = g^{-a}$.
3. La clé publique est $\mathbf{pk} = \alpha$, la clé privée est $\mathbf{sk} = a$.

SIGNATURE DE SCHNORR : Sign(\mathbf{m} , \mathbf{sk})

1. Choisir $k \in \{1, \dots, q-1\}$ aléatoirement.
2. Calculer $b = (g^k \bmod p) \bmod q$.
3. Calculer $r = H(b \parallel \mathbf{m})$ (r correspond à un "défi" aléatoire issu de la fonction de hachage).
4. Calculer $c = k + ar \bmod q$.
5. Retourner $s = (r, c)$.

SIGNATURE DE SCHNORR : KeyGen

1. Choisir a aléatoirement dans $\mathbb{Z}/q\mathbb{Z}^\times$.
2. Calculer $\alpha = g^{-a}$.
3. La clé publique est $\mathbf{pk} = \alpha$, la clé privée est $\mathbf{sk} = a$.

SIGNATURE DE SCHNORR : Sign(\mathbf{m} , \mathbf{sk})

1. Choisir $k \in \{1, \dots, q-1\}$ aléatoirement.
2. Calculer $b = (g^k \bmod p) \bmod q$.
3. Calculer $r = H(b \parallel \mathbf{m})$ (r correspond à un "défi" aléatoire issu de la fonction de hachage).
4. Calculer $c = k + ar \bmod q$.
5. Retourner $s = (r, c)$.

SIGNATURE DE SCHNORR : Verif(\mathbf{m} , s , \mathbf{pk})

1. Calculer $b' = g^c \alpha^r$
2. Calculer $r' = H(b' \parallel \mathbf{m})$
3. Faire le test $r' \equiv r \bmod q$ et retourner le booléen associé.

SIGNATURE DE SCHNORR : KeyGen

1. Choisir a aléatoirement dans $\mathbb{Z}/q\mathbb{Z}^\times$.
2. Calculer $\alpha = g^{-a}$.
3. La clé publique est $\text{pk} = \alpha$, la clé privée est $\text{sk} = a$.

SIGNATURE DE SCHNORR : Sign(\mathbf{m} , sk)

1. Choisir $k \in \{1, \dots, q-1\}$ aléatoirement.
2. Calculer $b = (g^k \bmod p) \bmod q$.
3. Calculer $r = H(b \parallel \mathbf{m})$ (r correspond à un "défi" aléatoire issu de la fonction de hachage).
4. Calculer $c = k + ar \bmod q$.
5. Retourner $s = (r, c)$.

SIGNATURE DE SCHNORR : Verif(\mathbf{m} , s, pk)

1. Calculer $b' = g^c \alpha^r$
2. Calculer $r' = H(b' \parallel \mathbf{m})$
3. Faire le test $r' \equiv r \bmod q$ et retourner le booléen associé.

Performances. \simeq DSA : clés courtes et adaptable sur les courbes elliptiques.

1. Identification et signature

Schémas d'identification

Construction à partir de signature

Construction sans signature

De l'identification à la signature

2. Chiffrement basé sur l'identité

Définitions

IBE de Cocks

Pour aller plus loin : IBE de Boneh-Franklin

1. Identification et signature

Schémas d'identification

Construction à partir de signature

Construction sans signature

De l'identification à la signature

2. Chiffrement basé sur l'identité

Définitions

IBE de Cocks

Pour aller plus loin : IBE de Boneh-Franklin

Une nouvelle primitive : **chiffrement basé sur l'identité** (*identity-based encryption*, IBE)

Une nouvelle primitive : **chiffrement basé sur l'identité** (*identity-based encryption*, IBE)

Motivation. Dans un système de chiffrement, on souhaite utiliser l'**identité d'un utilisateur comme clé publique** (par exemple, son adresse email).

Une nouvelle primitive : **chiffrement basé sur l'identité** (*identity-based encryption*, IBE)

Motivation. Dans un système de chiffrement, on souhaite utiliser l'**identité d'un utilisateur comme clé publique** (par exemple, son adresse email).

Avantages.

- L'expéditeur du message peut s'implement utiliser l'identité publique de son destinataire.
- Évite une prédistribution des clefs publiques (pas de PKI).

Une nouvelle primitive : **chiffrement basé sur l'identité** (*identity-based encryption*, IBE)

Motivation. Dans un système de chiffrement, on souhaite utiliser l'**identité d'un utilisateur comme clé publique** (par exemple, son adresse email).

Avantages.

- L'expéditeur du message peut s'implément utiliser l'identité publique de son destinataire.
- Évite une prédistribution des clefs publiques (pas de PKI).

Inconvénient. Une autorité centrale, possédant des **clés maîtresses**, doit délivrer des clés privées pour le déchiffrement.

Une nouvelle primitive : **chiffrement basé sur l'identité** (*identity-based encryption*, IBE)

Motivation. Dans un système de chiffrement, on souhaite utiliser l'**identité d'un utilisateur comme clé publique** (par exemple, son adresse email).

Avantages.

- L'expéditeur du message peut s'implement utiliser l'identité publique de son destinataire.
- Évite une prédistribution des clefs publiques (pas de PKI).

Inconvénient. Une autorité centrale, possédant des **clés maîtresses**, doit délivrer des clés privées pour le déchiffrement.

Historique. En 2001 :

- ▶ Cocks propose un IBE fondé sur le problème de la résiduosit  quadratique, mais peu efficace.
- ▶ Boneh et Franklin proposent un IBE fond  sur le probl me de Diffie-Hellman bilin aire, qui est alors un probl me peu  tudi .

Définition. Un schéma de **chiffrement basé sur l'identité** (*identity-based encryption*, IBE) est constitué de 4 algorithmes :

Définition. Un schéma de **chiffrement basé sur l'identité** (*identity-based encryption*, IBE) est constitué de 4 algorithmes :

1. Un algorithme de **génération de clés maîtresses** $\text{MasterKeyGen}()$, qui engendre mpk/msk , où la clé maîtresse privée msk est détenue par un tiers de confiance.

Définition. Un schéma de **chiffrement basé sur l'identité** (*identity-based encryption*, IBE) est constitué de 4 algorithmes :

1. Un algorithme de **génération de clés maîtresses** $\text{MasterKeyGen}()$, qui engendre mpk/msk , où la clé maîtresse privée msk est détenue par un tiers de confiance.
2. Un algorithme de **génération de clés personnelles** $\text{KeyGen}(\text{msk}, \text{id}_U)$ qui produit une clé privée sk_U destiné à un utilisateur U , à partir de son identité (publique).

Définition. Un schéma de **chiffrement basé sur l'identité** (*identity-based encryption*, IBE) est constitué de 4 algorithmes :

1. Un algorithme de **génération de clés maîtresses** $\text{MasterKeyGen}()$, qui engendre mpk/msk , où la clé maîtresse privée msk est détenue par un tiers de confiance.
2. Un algorithme de **génération de clés personnelles** $\text{KeyGen}(\text{msk}, \text{id}_U)$ qui produit une clé privée sk_U destiné à un utilisateur U , à partir de son identité (publique).
3. Un algorithme de **chiffrement** $\text{Enc}(\text{id}_U, m)$, avec un fonctionnement similaire au chiffrement à clef publique « classique », mais avec une clé publique correspondant à l'identité de U (parfois, on notera pk_U une clé publique dérivée de id_U),

Définition. Un schéma de **chiffrement basé sur l'identité** (*identity-based encryption*, IBE) est constitué de 4 algorithmes :

1. Un algorithme de **génération de clés maîtresses** $\text{MasterKeyGen}()$, qui engendre mpk/msk , où la clé maîtresse privée msk est détenue par un tiers de confiance.
2. Un algorithme de **génération de clés personnelles** $\text{KeyGen}(\text{msk}, \text{id}_U)$ qui produit une clé privée sk_U destiné à un utilisateur U , à partir de son identité (publique).
3. Un algorithme de **chiffrement** $\text{Enc}(\text{id}_U, \mathbf{m})$, avec un fonctionnement similaire au chiffrement à clef publique « classique », mais avec une clé publique correspondant à l'identité de U (parfois, on notera pk_U une clé publique dérivée de id_U),
4. Un algorithme de **déchiffrement** $\text{Dec}(\text{sk}_U, \mathbf{c})$, avec un fonctionnement similaire au déchiffrement à clef publique « classique ».

Définition. Un schéma de **chiffrement basé sur l'identité** (*identity-based encryption*, IBE) est constitué de 4 algorithmes :

1. Un algorithme de **génération de clés maîtresses** $\text{MasterKeyGen}()$, qui engendre mpk/msk , où la clé maîtresse privée msk est détenue par un tiers de confiance.
2. Un algorithme de **génération de clés personnelles** $\text{KeyGen}(\text{msk}, \text{id}_U)$ qui produit une clé privée sk_U destiné à un utilisateur U , à partir de son identité (publique).
3. Un algorithme de **chiffrement** $\text{Enc}(\text{id}_U, \mathbf{m})$, avec un fonctionnement similaire au chiffrement à clef publique « classique », mais avec une clé publique correspondant à l'identité de U (parfois, on notera pk_U une clé publique dérivée de id_U),
4. Un algorithme de **déchiffrement** $\text{Dec}(\text{sk}_U, \mathbf{c})$, avec un fonctionnement similaire au déchiffrement à clef publique « classique ».

Les **modèles de sécurité** sont adaptables à partir de ceux du chiffrement à clef publique vus dans les cours précédents.

1. Identification et signature

Schémas d'identification

Construction à partir de signature

Construction sans signature

De l'identification à la signature

2. Chiffrement basé sur l'identité

Définitions

IBE de Cocks

Pour aller plus loin : IBE de Boneh-Franklin

Soit $n = pq$ où p et q sont deux nombres premiers distincts tels que $p \equiv q \equiv 3 \pmod{4}$.

Soit $n = pq$ où p et q sont deux nombres premiers distincts tels que $p \equiv q \equiv 3 \pmod{4}$.

On note QR_n^\times les résidus quadratiques inversibles modulo n , c'est-à-dire

$$\text{QR}_n^\times := \left\{ x \in (\mathbb{Z}/n\mathbb{Z})^\times \mid \left(\frac{x}{p} \right) = \left(\frac{x}{q} \right) = 1 \right\}.$$

Soit $n = pq$ où p et q sont deux nombres premiers distincts tels que $p \equiv q \equiv 3 \pmod{4}$.

On note QR_n^\times les résidus quadratiques inversibles modulo n , c'est-à-dire

$$\text{QR}_n^\times := \left\{ x \in (\mathbb{Z}/n\mathbb{Z})^\times \mid \left(\frac{x}{p}\right) = \left(\frac{x}{q}\right) = 1 \right\}.$$

Les **pseudo-résidus quadratiques** modulo n sont

$$\overline{\text{QR}}_n^\times := \left\{ x \in (\mathbb{Z}/n\mathbb{Z})^\times \mid \left(\frac{x}{p}\right) = \left(\frac{x}{q}\right) = -1 \right\}.$$

Soit $n = pq$ où p et q sont deux nombres premiers distincts tels que $p \equiv q \equiv 3 \pmod{4}$.

On note QR_n^\times les résidus quadratiques inversibles modulo n , c'est-à-dire

$$\text{QR}_n^\times := \left\{ x \in (\mathbb{Z}/n\mathbb{Z})^\times \mid \left(\frac{x}{p}\right) = \left(\frac{x}{q}\right) = 1 \right\}.$$

Les **pseudo-résidus quadratiques** modulo n sont

$$\overline{\text{QR}}_n^\times := \left\{ x \in (\mathbb{Z}/n\mathbb{Z})^\times \mid \left(\frac{x}{p}\right) = \left(\frac{x}{q}\right) = -1 \right\}.$$

On note enfin

$$\mathcal{Q} := \text{QR}_n^\times \cup \overline{\text{QR}}_n^\times = \left\{ x \in (\mathbb{Z}/n\mathbb{Z})^\times \mid \left(\frac{x}{n}\right) = 1 \right\}$$

Description de l'IBE de Cocks. On suppose qu'on a à notre disposition une fonction de hachage $H : \{0, 1\}^* \rightarrow \mathcal{Q}$.

Description de l'IBE de Cocks. On suppose qu'on a à notre disposition une fonction de hachage $H : \{0, 1\}^* \rightarrow \mathcal{Q}$.

Clés maîtresses. La clé publique est $mpk = n$ et la clé privée est $msk = (p, q)$.

Description de l'IBE de Cocks. On suppose qu'on a à notre disposition une fonction de hachage $H : \{0, 1\}^* \rightarrow \mathcal{Q}$.

Clés maîtresses. La clé publique est $\text{mpk} = n$ et la clé privée est $\text{msk} = (p, q)$.

Génération de clés pour l'utilisateur U d'identité id_U .

1. La clé publique pk_U est $H(\text{id}_U) \in \mathcal{Q}$.
2. La clé privée sk_U est une racine carrée de $\begin{cases} \text{pk}_U & \text{si } \text{pk}_U \in \text{QR}_n^\times, \\ -\text{pk}_U & \text{sinon} . \end{cases}$

Description de l'IBE de Cocks. On suppose qu'on a à notre disposition une fonction de hachage $H : \{0, 1\}^* \rightarrow \mathcal{Q}$.

Clés maîtresses. La clé publique est $mpk = n$ et la clé privée est $msk = (p, q)$.

Génération de clés pour l'utilisateur U d'identité id_U .

1. La clé publique pk_U est $H(id_U) \in \mathcal{Q}$.
2. La clé privée sk_U est une racine carrée de $\begin{cases} pk_U & \text{si } pk_U \in \mathbb{QR}_n^\times, \\ -pk_U & \text{sinon.} \end{cases}$

Remarque. Seul le détenteur de la factorisation (p, q) de n est capable de décider si pk_U est dans \mathbb{QR}_n^\times ou dans $\overline{\mathbb{QR}_n^\times}$. C'est le problème de la **résiduosit  quadratique** qui est suppos  difficile.

L'espace des clairs est $\mathcal{M} = \{-1, 1\}$, et celui des chiffrés est $\mathcal{C} = (\mathbb{Z}/n\mathbb{Z})^2$.

L'espace des clairs est $\mathcal{M} = \{-1, 1\}$, et celui des chiffrés est $\mathcal{C} = (\mathbb{Z}/n\mathbb{Z})^2$.

Chiffrement. On veut chiffrer un élément $m \in \{-1, 1\}$.

1. Choisir aléatoirement $t_1, t_2 \in \mathbb{Z}/n\mathbb{Z}$, tels que $\left(\frac{t_1}{n}\right) = \left(\frac{t_2}{n}\right) = m$.

2. Calculer

$$\begin{cases} y_1 &= t_1 + \text{pk}_U \times t_1^{-1} \pmod n \\ y_2 &= t_2 - \text{pk}_U \times t_2^{-1} \pmod n \end{cases}$$

3. Le chiffré est $y = (y_1, y_2)$.

L'espace des clairs est $\mathcal{M} = \{-1, 1\}$, et celui des chiffrés est $\mathcal{C} = (\mathbb{Z}/n\mathbb{Z})^2$.

Chiffrement. On veut chiffrer un élément $m \in \{-1, 1\}$.

1. Choisir aléatoirement $t_1, t_2 \in \mathbb{Z}/n\mathbb{Z}$, tels que $\left(\frac{t_1}{n}\right) = \left(\frac{t_2}{n}\right) = m$.
2. Calculer

$$\begin{cases} y_1 &= t_1 + \text{pk}_U \times t_1^{-1} \pmod n \\ y_2 &= t_2 - \text{pk}_U \times t_2^{-1} \pmod n \end{cases}$$

3. Le chiffré est $y = (y_1, y_2)$.

Déchiffrement. On veut déchiffrer $y = (y_1, y_2) \in (\mathbb{Z}/n\mathbb{Z})^2$.

1. Tester si $\text{pk}_U = (\text{sk}_U)^2$:
 - si c'est vrai, définir $s = y_1$,
 - sinon, définir $s = y_2$.

2. Calculer et retourner :

$$m' = \left(\frac{s + 2\text{sk}_U}{n}\right)$$

L'espace des clairs est $\mathcal{M} = \{-1, 1\}$, et celui des chiffrés est $\mathcal{C} = (\mathbb{Z}/n\mathbb{Z})^2$.

Chiffrement. On veut chiffrer un élément $m \in \{-1, 1\}$.

1. Choisir aléatoirement $t_1, t_2 \in \mathbb{Z}/n\mathbb{Z}$, tels que $\left(\frac{t_1}{n}\right) = \left(\frac{t_2}{n}\right) = m$.
2. Calculer

$$\begin{cases} y_1 &= t_1 + \text{pk}_U \times t_1^{-1} \pmod n \\ y_2 &= t_2 - \text{pk}_U \times t_2^{-1} \pmod n \end{cases}$$

3. Le chiffré est $y = (y_1, y_2)$.

Déchiffrement. On veut déchiffrer $y = (y_1, y_2) \in (\mathbb{Z}/n\mathbb{Z})^2$.

1. Tester si $\text{pk}_U = (\text{sk}_U)^2$:
 - si c'est vrai, définir $s = y_1$,
 - sinon, définir $s = y_2$.

2. Calculer et retourner :

$$m' = \left(\frac{s + 2\text{sk}_U}{n}\right)$$

Remarque. On ne chiffre qu'un bit à la fois, mais on peut réitérer le procédé.

Validité. Supposons que $(sk_U)^2 = pk_U$ (par exemple).

Validité. Supposons que $(sk_U)^2 = pk_U$ (par exemple). Il faut démontrer que $\left(\frac{s+2sk_U}{n}\right) = m$ où $s = t_1 + pk_U(t_1^{-1}) \pmod n$ et $\left(\frac{t_1}{n}\right) = m$.

Validité. Supposons que $(sk_U)^2 = pk_U$ (par exemple). Il faut démontrer que $\left(\frac{s+2sk_U}{n}\right) = m$ où

$$s = t_1 + pk_U(t_1^{-1}) \pmod n \text{ et } \left(\frac{t_1}{n}\right) = m.$$

On a :

$$\left(\frac{s + 2sk_U}{n}\right) =$$

Validité. Supposons que $(sk_U)^2 = pk_U$ (par exemple). Il faut démontrer que $\left(\frac{s+2sk_U}{n}\right) = m$ où $s = t_1 + pk_U(t_1^{-1}) \pmod n$ et $\left(\frac{t_1}{n}\right) = m$.

On a :

$$\left(\frac{s + 2sk_U}{n}\right) = \left(\frac{t_1 + pk_U(t_1^{-1}) + 2sk_U}{n}\right) =$$

Validité. Supposons que $(sk_U)^2 = pk_U$ (par exemple). Il faut démontrer que $\left(\frac{s+2sk_U}{n}\right) = m$ où $s = t_1 + pk_U(t_1^{-1}) \pmod n$ et $\left(\frac{t_1}{n}\right) = m$.

On a :

$$\left(\frac{s + 2sk_U}{n}\right) = \left(\frac{t_1 + pk_U(t_1^{-1}) + 2sk_U}{n}\right) = \left(\frac{t_1 + (sk_U)^2(t_1^{-1}) + 2sk_U}{n}\right)$$

Validité. Supposons que $(sk_U)^2 = pk_U$ (par exemple). Il faut démontrer que $\left(\frac{s+2sk_U}{n}\right) = m$ où $s = t_1 + pk_U(t_1^{-1}) \pmod n$ et $\left(\frac{t_1}{n}\right) = m$.

On a :

$$\begin{aligned} \left(\frac{s+2sk_U}{n}\right) &= \left(\frac{t_1 + pk_U(t_1^{-1}) + 2sk_U}{n}\right) = \left(\frac{t_1 + (sk_U)^2(t_1^{-1}) + 2sk_U}{n}\right) \\ &= \left(\frac{t_1(1 + 2sk_U(t_1^{-1}) + (sk_U t_1^{-1})^2)}{n}\right) = \end{aligned}$$

Validité. Supposons que $(sk_U)^2 = pk_U$ (par exemple). Il faut démontrer que $\left(\frac{s+2sk_U}{n}\right) = m$ où $s = t_1 + pk_U(t_1^{-1}) \pmod n$ et $\left(\frac{t_1}{n}\right) = m$.

On a :

$$\begin{aligned} \left(\frac{s+2sk_U}{n}\right) &= \left(\frac{t_1 + pk_U(t_1^{-1}) + 2sk_U}{n}\right) = \left(\frac{t_1 + (sk_U)^2(t_1^{-1}) + 2sk_U}{n}\right) \\ &= \left(\frac{t_1(1 + 2sk_U(t_1^{-1}) + (sk_U t_1^{-1})^2)}{n}\right) = \left(\frac{t_1(1 + sk_U t_1^{-1})^2}{n}\right) \end{aligned}$$

Validité. Supposons que $(sk_U)^2 = pk_U$ (par exemple). Il faut démontrer que $\left(\frac{s+2sk_U}{n}\right) = m$ où $s = t_1 + pk_U(t_1^{-1}) \pmod n$ et $\left(\frac{t_1}{n}\right) = m$.

On a :

$$\begin{aligned} \left(\frac{s+2sk_U}{n}\right) &= \left(\frac{t_1 + pk_U(t_1^{-1}) + 2sk_U}{n}\right) = \left(\frac{t_1 + (sk_U)^2(t_1^{-1}) + 2sk_U}{n}\right) \\ &= \left(\frac{t_1(1 + 2sk_U(t_1^{-1}) + (sk_U t_1^{-1})^2)}{n}\right) = \left(\frac{t_1(1 + sk_U t_1^{-1})^2}{n}\right) \\ &= \left(\frac{t_1}{n}\right) \left(\left(\frac{1 + sk_U t_1^{-1}}{n}\right)\right)^2 = \end{aligned}$$

Validité. Supposons que $(sk_U)^2 = pk_U$ (par exemple). Il faut démontrer que $\left(\frac{s+2sk_U}{n}\right) = m$ où $s = t_1 + pk_U(t_1^{-1}) \pmod n$ et $\left(\frac{t_1}{n}\right) = m$.

On a :

$$\begin{aligned} \left(\frac{s+2sk_U}{n}\right) &= \left(\frac{t_1 + pk_U(t_1^{-1}) + 2sk_U}{n}\right) = \left(\frac{t_1 + (sk_U)^2(t_1^{-1}) + 2sk_U}{n}\right) \\ &= \left(\frac{t_1(1 + 2sk_U(t_1^{-1}) + (sk_U t_1^{-1})^2)}{n}\right) = \left(\frac{t_1(1 + sk_U t_1^{-1})^2}{n}\right) \\ &= \left(\frac{t_1}{n}\right) \left(\left(\frac{1 + sk_U t_1^{-1}}{n}\right)\right)^2 = \left(\frac{t_1}{n}\right) = m \end{aligned}$$

Validité. Supposons que $(sk_U)^2 = pk_U$ (par exemple). Il faut démontrer que $\left(\frac{s+2sk_U}{n}\right) = m$ où $s = t_1 + pk_U(t_1^{-1}) \pmod n$ et $\left(\frac{t_1}{n}\right) = m$.

On a :

$$\begin{aligned} \left(\frac{s+2sk_U}{n}\right) &= \left(\frac{t_1 + pk_U(t_1^{-1}) + 2sk_U}{n}\right) = \left(\frac{t_1 + (sk_U)^2(t_1^{-1}) + 2sk_U}{n}\right) \\ &= \left(\frac{t_1(1 + 2sk_U(t_1^{-1}) + (sk_U t_1^{-1})^2)}{n}\right) = \left(\frac{t_1(1 + sk_U t_1^{-1})^2}{n}\right) \\ &= \left(\frac{t_1}{n}\right) \left(\left(\frac{1 + sk_U t_1^{-1}}{n}\right)\right)^2 = \left(\frac{t_1}{n}\right) = m \end{aligned}$$

Exercices.

- ▶ Pourquoi $\left(\frac{1+sk_U t_1^{-1}}{n}\right) \neq 0$?
- ▶ Démontrer la validité pour le cas $(sk_U)^2 = -pk_U$.

Sécurité. La sécurité de l'IBE de Cocks se réduit au **problème de la résiduosit  quadratique**.

Sécurité. La sécurité de l'IBE de Cocks se réduit au **problème de la résiduosit  quadratique**.

 l ments de preuve. Supposons que l'on ait un algorithme `ALGO` qui d crypte les chiffr s de Cocks. Concevons un algorithme qui r sout le probl me de la r siduosit  quadratique avec une bonne probabilit .

Sécurité. La sécurité de l'IBE de Cocks se réduit au **problème de la résiduosit  quadratique**.

El ments de preuve. Supposons que l'on ait un algorithme `ALGO` qui d crypte les chiffr s de Cocks. Concevons un algorithme qui r sout le probl me de la r siduosit  quadratique avec une bonne probabilit .

Entr e : a et n tels que $\left(\frac{a}{n}\right) = 1$

Sortie : $a \in \text{QR}_n^\times$?

1. Choisir $x \in \{-1, 1\}$ al atoirement
2. Choisir un entier t tel que $\left(\frac{t}{n}\right) = x$
3. Calculer $y_1 = t + at^{-1} \pmod n$
4. Choisir y_2 al atoirement
5. D finir $y = (y_1, y_2)$ et appeler $x' \leftarrow \text{ALGO}((n, a), y)$
6. **Si** $x' = x$, alors **retourner** $a \in \text{QR}_n^\times$
7. **Sinon, retourner** $a \notin \text{QR}_n^\times$

Sécurité. La sécurité de l'IBE de Cocks se réduit au **problème de la résiduosit  quadratique**.

El ments de preuve. Supposons que l'on ait un algorithme `ALGO` qui d crypte les chiffr s de Cocks. Concevons un algorithme qui r sout le probl me de la r siduosit  quadratique avec une bonne probabilit .

Entr e : a et n tels que $\left(\frac{a}{n}\right) = 1$

Sortie : $a \in \text{QR}_n^\times$?

1. Choisir $x \in \{-1, 1\}$ al atoirement
2. Choisir un entier t tel que $\left(\frac{t}{n}\right) = x$
3. Calculer $y_1 = t + at^{-1} \pmod n$
4. Choisir y_2 al atoirement
5. D finir $y = (y_1, y_2)$ et appeler $x' \leftarrow \text{ALGO}((n, a), y)$
6. **Si** $x' = x$, alors **retourner** $a \in \text{QR}_n^\times$
7. **Sinon, retourner** $a \notin \text{QR}_n^\times$

L'algorithme est correct avec

- ▶ probabilit  1 si $a \in \text{QR}_n^\times$
- ▶ probabilit  1/2 si $a \notin \text{QR}_n^\times$

1. Identification et signature

Schémas d'identification

Construction à partir de signature

Construction sans signature

De l'identification à la signature

2. Chiffrement basé sur l'identité

Définitions

IBE de Cocks

Pour aller plus loin : IBE de Boneh-Franklin

Pour construire l'IBE de Boneh-Franklin, on a besoin de la notion de **couplage** (*pairing*).

Pour construire l'IBE de Boneh-Franklin, on a besoin de la notion de **couplage** (*pairing*).

Définition. Soient $(\mathbb{G}_1, +)$, $(\mathbb{G}_2, +)$, (\mathbb{G}_3, \cdot) trois groupes commutatifs. Un **couplage** est une application

$$e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_3 .$$

Un couplage est **bilinéaire** si, pour tout $P_1, P_2 \in \mathbb{G}_1$ et tout $Q_1, Q_2 \in \mathbb{G}_2$ on a :

$$e(P_1 + P_2, Q_1) = e(P_1, Q_1) \cdot e(P_2, Q_1) \quad \text{et} \quad e(P_1, Q_1 + Q_2) = e(P_1, Q_1) \cdot e(P_1, Q_2)$$

Pour construire l'IBE de Boneh-Franklin, on a besoin de la notion de **couplage** (*pairing*).

Définition. Soient $(\mathbb{G}_1, +)$, $(\mathbb{G}_2, +)$, (\mathbb{G}_3, \cdot) trois groupes commutatifs. Un **couplage** est une application

$$e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_3 .$$

Un couplage est **bilinéaire** si, pour tout $P_1, P_2 \in \mathbb{G}_1$ et tout $Q_1, Q_2 \in \mathbb{G}_2$ on a :

$$e(P_1 + P_2, Q_1) = e(P_1, Q_1) \cdot e(P_2, Q_1) \quad \text{et} \quad e(P_1, Q_1 + Q_2) = e(P_1, Q_1) \cdot e(P_1, Q_2)$$

Remarque. Si $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_3$ est un couplage bilinéaire, alors, $e(aP, bQ) = e(P, Q)^{ab}$.

Pour construire l'IBE de Boneh-Franklin, on a besoin de la notion de **couplage** (*pairing*).

Définition. Soient $(\mathbb{G}_1, +)$, $(\mathbb{G}_2, +)$, (\mathbb{G}_3, \cdot) trois groupes commutatifs. Un **couplage** est une application

$$e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_3 .$$

Un couplage est **bilinéaire** si, pour tout $P_1, P_2 \in \mathbb{G}_1$ et tout $Q_1, Q_2 \in \mathbb{G}_2$ on a :

$$e(P_1 + P_2, Q_1) = e(P_1, Q_1) \cdot e(P_2, Q_1) \quad \text{et} \quad e(P_1, Q_1 + Q_2) = e(P_1, Q_1) \cdot e(P_1, Q_2)$$

Remarque. Si $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_3$ est un couplage bilinéaire, alors, $e(aP, bQ) = e(P, Q)^{ab}$.

Exemples (hors contexte cryptographique)

- ▶ Le produit scalaire $\langle \cdot, \cdot \rangle : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$, où $\mathbb{G}_3 = \mathbb{R}$ est noté additivement
- ▶ Le déterminant $\det : \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}$, où $\mathbb{G}_3 = \mathbb{R}$ est noté additivement.

Pour construire l'IBE de Boneh-Franklin, on a besoin de la notion de **couplage** (*pairing*).

Définition. Soient $(\mathbb{G}_1, +)$, $(\mathbb{G}_2, +)$, (\mathbb{G}_3, \cdot) trois groupes commutatifs. Un **couplage** est une application

$$e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_3 .$$

Un couplage est **bilinéaire** si, pour tout $P_1, P_2 \in \mathbb{G}_1$ et tout $Q_1, Q_2 \in \mathbb{G}_2$ on a :

$$e(P_1 + P_2, Q_1) = e(P_1, Q_1) \cdot e(P_2, Q_1) \quad \text{et} \quad e(P_1, Q_1 + Q_2) = e(P_1, Q_1) \cdot e(P_1, Q_2)$$

Remarque. Si $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_3$ est un couplage bilinéaire, alors, $e(aP, bQ) = e(P, Q)^{ab}$.

Exemples (hors contexte cryptographique)

- ▶ Le produit scalaire $\langle \cdot, \cdot \rangle : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$, où $\mathbb{G}_3 = \mathbb{R}$ est noté additivement
- ▶ Le déterminant $\det : \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}$, où $\mathbb{G}_3 = \mathbb{R}$ est noté additivement.

En cryptographie, on utilise des couplages provenant de **courbes elliptiques**. Par exemple, si $E(\mathbb{F}_p)$ est le groupe de points d'une courbe E sur \mathbb{F}_p , on note $E(\mathbb{F}_p)[\ell]$ le sous-groupe des points d'ordre divisant ℓ .

Pour construire l'IBE de Boneh-Franklin, on a besoin de la notion de **couplage** (*pairing*).

Définition. Soient $(\mathbb{G}_1, +)$, $(\mathbb{G}_2, +)$, (\mathbb{G}_3, \cdot) trois groupes commutatifs. Un **couplage** est une application

$$e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_3 .$$

Un couplage est **bilinéaire** si, pour tout $P_1, P_2 \in \mathbb{G}_1$ et tout $Q_1, Q_2 \in \mathbb{G}_2$ on a :

$$e(P_1 + P_2, Q_1) = e(P_1, Q_1) \cdot e(P_2, Q_1) \quad \text{et} \quad e(P_1, Q_1 + Q_2) = e(P_1, Q_1) \cdot e(P_1, Q_2)$$

Remarque. Si $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_3$ est un couplage bilinéaire, alors, $e(aP, bQ) = e(P, Q)^{ab}$.

Exemples (hors contexte cryptographique)

- ▶ Le produit scalaire $\langle \cdot, \cdot \rangle : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$, où $\mathbb{G}_3 = \mathbb{R}$ est noté additivement
- ▶ Le déterminant $\det : \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}$, où $\mathbb{G}_3 = \mathbb{R}$ est noté additivement.

En cryptographie, on utilise des couplages provenant de **courbes elliptiques**. Par exemple, si $E(\mathbb{F}_p)$ est le groupe de points d'une courbe E sur \mathbb{F}_p , on note $E(\mathbb{F}_p)[\ell]$ le sous-groupe des points d'ordre divisant ℓ . Alors, il existe un **couplage de Weil** (explicite)

$$e : E(\mathbb{F}_p)[\ell] \times E(\mathbb{F}_p)[\ell] \rightarrow \mathbb{F}_{p^k}^\times$$

Remarque : la description de ce couplage demande des notions avancées en courbes elliptiques.

Soit $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_3$ un couplage bilinéaire, et P un générateur de \mathbb{G}_2 , que l'on suppose d'ordre q . On note \mathbb{G}_1 et \mathbb{G}_2 additivement, et \mathbb{G}_3 multiplicativement. On se donne également deux fonctions de hachage

$$H_1 : \{0, 1\}^* \rightarrow \mathbb{G}_1 \setminus \{1\}$$

$$H_2 : \mathbb{G}_2 \rightarrow \{0, 1\}^n$$

où n est un paramètre (longueur du clair).

Soit $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_3$ un couplage bilinéaire, et P un générateur de \mathbb{G}_2 , que l'on suppose d'ordre q . On note \mathbb{G}_1 et \mathbb{G}_2 additivement, et \mathbb{G}_3 multiplicativement. On se donne également deux fonctions de hachage

$$H_1 : \{0, 1\}^* \rightarrow \mathbb{G}_1 \setminus \{1\}$$

$$H_2 : \mathbb{G}_2 \rightarrow \{0, 1\}^n$$

où n est un paramètre (longueur du clair).

Boneh et Franklin (2001) ont proposé l'IBE suivant.

Soit $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_3$ un couplage bilinéaire, et P un générateur de \mathbb{G}_2 , que l'on suppose d'ordre q . On note \mathbb{G}_1 et \mathbb{G}_2 additivement, et \mathbb{G}_3 multiplicativement. On se donne également deux fonctions de hachage

$$H_1 : \{0, 1\}^* \rightarrow \mathbb{G}_1 \setminus \{1\}$$

$$H_2 : \mathbb{G}_2 \rightarrow \{0, 1\}^n$$

où n est un paramètre (longueur du clair).

Boneh et Franklin (2001) ont proposé l'IBE suivant.

Clés maîtresses. $\text{msk} = s \in \{1, \dots, q - 1\}$ et $\text{mpk} = sP \in \mathbb{G}_2$

Soit $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_3$ un couplage bilinéaire, et P un générateur de \mathbb{G}_2 , que l'on suppose d'ordre q . On note \mathbb{G}_1 et \mathbb{G}_2 additivement, et \mathbb{G}_3 multiplicativement. On se donne également deux fonctions de hachage

$$H_1 : \{0, 1\}^* \rightarrow \mathbb{G}_1 \setminus \{1\}$$

$$H_2 : \mathbb{G}_2 \rightarrow \{0, 1\}^n$$

où n est un paramètre (longueur du clair).

Boneh et Franklin (2001) ont proposé l'IBE suivant.

Clés maîtresses. $\text{msk} = s \in \{1, \dots, q-1\}$ et $\text{mpk} = sP \in \mathbb{G}_2$

Génération de clés. Pour l'utilisateur U , la paire de clef $(\text{pk}_U, \text{sk}_U)$ est créée, où

$$\text{pk}_U = H_1(\text{id}_U) \in \mathbb{G}_1$$

et

$$\text{sk}_U = s \cdot \text{pk}_U \in \mathbb{G}_1$$

L'espace des clés est $\mathcal{M} = \{0, 1\}^n$ et l'espace des chiffrés est $\mathcal{C} = \mathbb{G}_2 \times \{0, 1\}^n$.

L'espace des clairs est $\mathcal{M} = \{0, 1\}^n$ et l'espace des chiffrés est $\mathcal{C} = \mathbb{G}_2 \times \{0, 1\}^n$.

Chiffrement. Pour chiffrer un message $\mathbf{m} \in \{0, 1\}^n$:

1. Choisir $r \in \mathbb{Z}/q\mathbb{Z}$ inversible.
2. Calculer $\mathbf{y}_1 = \mathbf{m} \oplus H_2(e(\mathbf{pk}_U, \mathbf{mpk})^r) \in \{0, 1\}^n$
3. Calculer $y_2 = rP \in \mathbb{G}_2$.
4. Le chiffré est $y = (\mathbf{y}_1, y_2)$

L'espace des clairs est $\mathcal{M} = \{0, 1\}^n$ et l'espace des chiffrés est $\mathcal{C} = \mathbb{G}_2 \times \{0, 1\}^n$.

Chiffrement. Pour chiffrer un message $\mathbf{m} \in \{0, 1\}^n$:

1. Choisir $r \in \mathbb{Z}/q\mathbb{Z}$ inversible.
2. Calculer $\mathbf{y}_1 = \mathbf{m} \oplus H_2(e(\mathbf{pk}_U, \mathbf{mpk})^r) \in \{0, 1\}^n$
3. Calculer $y_2 = rP \in \mathbb{G}_2$.
4. Le chiffré est $y = (\mathbf{y}_1, y_2)$

Déchiffrement. Pour déchiffrer $y = (\mathbf{y}_1, y_2)$:

1. Calculer $\mathbf{m}' = \mathbf{y}_1 \oplus H_2(e(\mathbf{sk}_U, y_2)) \in \{0, 1\}^n$.
2. Retourner \mathbf{m}' .

L'espace des clés est $\mathcal{M} = \{0, 1\}^n$ et l'espace des chiffrés est $\mathcal{C} = \mathbb{G}_2 \times \{0, 1\}^n$.

Chiffrement. Pour chiffrer un message $m \in \{0, 1\}^n$:

1. Choisir $r \in \mathbb{Z}/q\mathbb{Z}$ inversible.
2. Calculer $y_1 = m \oplus H_2(e(\text{pk}_U, \text{mpk})^r) \in \{0, 1\}^n$
3. Calculer $y_2 = rP \in \mathbb{G}_2$.
4. Le chiffré est $y = (y_1, y_2)$

Déchiffrement. Pour déchiffrer $y = (y_1, y_2)$:

1. Calculer $m' = y_1 \oplus H_2(e(\text{sk}_U, y_2)) \in \{0, 1\}^n$.
2. Retourner m' .

Validité. Il suffit de vérifier que $e(\text{sk}_U, y_2) = e(\text{pk}_U, \text{mpk})^r$. C'est vrai car :

$$e(\text{sk}_U, y_2) = e(s\text{pk}_U, rP) = e(\text{pk}_U, P)^{rs} = e(\text{pk}_U, sP)^r = e(\text{pk}_U, \text{mpk})^r.$$

Sécurité. Dans le modèle de l'oracle aléatoire (rappel informel : \simeq fonctions de hachage idéales), la **sécurité sémantique** (CPA) de l'IBE de Boneh-Franklin se réduit au problème de Diffie-Hellman bilinéaire.

Sécurité. Dans le modèle de l'oracle aléatoire (rappel informel : \simeq fonctions de hachage idéales), la **sécurité sémantique** (CPA) de l'IBE de Boneh-Franklin se réduit au problème de Diffie-Hellman bilinéaire.

Problème de Diffie-Hellman bilinéaire (BDH). Soient \mathbb{G}_1 et \mathbb{G}_2 deux groupes finis d'ordre q , notés additivement, et soit $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_3$ un couplage bilinéaire où \mathbb{G}_3 est noté multiplicativement.

Instance. un quadruplet (P, Q, aQ, bQ) , où $P \in \mathbb{G}_1$, $Q \in \mathbb{G}_2$ d'ordre q , et $a, b \in (\mathbb{Z}/q\mathbb{Z})^\times$.

Question. Trouver Z tel que $Z = e(P, Q)^{ab}$.

Sécurité. Dans le modèle de l'oracle aléatoire (rappel informel : \simeq fonctions de hachage idéales), la **sécurité sémantique** (CPA) de l'IBE de Boneh-Franklin se réduit au problème de Diffie-Hellman bilinéaire.

Problème de Diffie-Hellman bilinéaire (BDH). Soient \mathbb{G}_1 et \mathbb{G}_2 deux groupes finis d'ordre q , notés additivement, et soit $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_3$ un couplage bilinéaire où \mathbb{G}_3 est noté multiplicativement.

Instance. un quadruplet (P, Q, aQ, bQ) , où $P \in \mathbb{G}_1$, $Q \in \mathbb{G}_2$ d'ordre q , et $a, b \in (\mathbb{Z}/q\mathbb{Z})^\times$.

Question. Trouver Z tel que $Z = e(P, Q)^{ab}$.

Ce problème est supposé **difficile** pour des groupes génériques \mathbb{G}_i .

Sécurité. Dans le modèle de l'oracle aléatoire (rappel informel : \simeq fonctions de hachage idéales), la **sécurité sémantique** (CPA) de l'IBE de Boneh-Franklin se réduit au problème de Diffie-Hellman bilinéaire.

Problème de Diffie-Hellman bilinéaire (BDH). Soient \mathbb{G}_1 et \mathbb{G}_2 deux groupes finis d'ordre q , notés additivement, et soit $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_3$ un couplage bilinéaire où \mathbb{G}_3 est noté multiplicativement.

Instance. un quadruplet (P, Q, aQ, bQ) , où $P \in \mathbb{G}_1$, $Q \in \mathbb{G}_2$ d'ordre q , et $a, b \in (\mathbb{Z}/q\mathbb{Z})^\times$.

Question. Trouver Z tel que $Z = e(P, Q)^{ab}$.

Ce problème est supposé **difficile** pour des groupes génériques \mathbb{G}_i .

Exercice : Montrer que si BDH est difficile, alors DL est également difficile dans \mathbb{G}_1 et \mathbb{G}_2 .

Questions ?