Introduction à la sécurité
Cours + – Chiffrement authentifié et identification

Julien Lavauzelle

Université Paris 8

Licence 3 Informatique et Vidéoludisme

20/09/2024

Plan

Motivations

Contexte. Alice et Bob font partie d'un grand réseau de communication, dans lequel chaque participant engendre une paire de clefs publique/privée (pour du chiffrement ou de la signature).

Motivations

Contexte. Alice et Bob font partie d'un grand réseau de communication, dans lequel chaque participant engendre une paire de clefs publique/privée (pour du chiffrement ou de la signature).

Les clefs publiques sont émises publiquement. Pour valider ces clefs, on a besoin d'une **infrastructure à clef publique** (*public-key infrastructure*, PKI).

Motivations

Contexte. Alice et Bob font partie d'un grand réseau de communication, dans lequel chaque participant engendre une paire de clefs publique/privée (pour du chiffrement ou de la signature).

Les clefs publiques sont émises publiquement. Pour valider ces clefs, on a besoin d'une **infrastructure à clef publique** (*public-key infrastructure*, PKI).

Une solution est d'adoption d'une **autorité de certification** (*certification authority*, CA).

- C'est un organisme externe auquel tout participant fait confiance.
- Cette autorité va produire des **certificats** d'authenticité, par exemple pour les clefs publiques des utilisateurs.

Motivations

Contexte. Alice et Bob font partie d'un grand réseau de communication, dans lequel chaque participant engendre une paire de clefs publique/privée (pour du chiffrement ou de la signature).

Les clefs publiques sont émises publiquement. Pour valider ces clefs, on a besoin d'une **infrastructure à clef publique** (*public-key infrastructure*, PKI).

Une solution est d'adoption d'une **autorité de certification** (*certification authority*, CA).

- C'est un organisme externe auquel tout participant fait confiance.
- Cette autorité va produire des **certificats** d'authenticité, par exemple pour les clefs publiques des utilisateurs.

Autres types de PKI :

Motivations

Contexte. Alice et Bob font partie d'un grand réseau de communication, dans lequel chaque participant engendre une paire de clefs publique/privée (pour du chiffrement ou de la signature).

Les clefs publiques sont émises publiquement. Pour valider ces clefs, on a besoin d'une **infrastructure à clef publique** (*public-key infrastructure*, PKI).

Une solution est d'adoption d'une **autorité de certification** (*certification authority*, CA).

- C'est un organisme externe auquel tout participant fait confiance.
- Cette autorité va produire des **certificats** d'authenticité, par exemple pour les clefs publiques des utilisateurs.

Autres types de PKI :

- **Toile de confiance** (*web of trust*), utilisé dans PGP (*pretty good privacy*) : modèle complètement décentralisé où chaque entité peut certifier les clefs d'autres entités.

Motivations

Contexte. Alice et Bob font partie d'un grand réseau de communication, dans lequel chaque participant engendre une paire de clefs publique/privée (pour du chiffrement ou de la signature).

Les clefs publiques sont émises publiquement. Pour valider ces clefs, on a besoin d'une **infrastructure à clef publique** (*public-key infrastructure*, PKI).

Une solution est d'adoption d'une **autorité de certification** (*certification authority*, CA).

- C'est un organisme externe auquel tout participant fait confiance.
- Cette autorité va produire des **certificats** d'authenticité, par exemple pour les clefs publiques des utilisateurs.

Autres types de PKI :

- **Toile de confiance** (*web of trust*), utilisé dans PGP (*pretty good privacy*) : modèle complètement décentralisé où chaque entité peut certifier les clefs d'autres entités.
- Plus récemment : ***blockchains*** (ex : CertCoin).

Certificats

Un **certificat** est une structure de données reliant

- l'identité d'une personne (nom, adresse email, etc.),
- une information à certifier (clef publique)
- et la signature d'une autorité de confiance.

Certificats

Un **certificat** est une structure de données reliant
- l'identité d'une personne (nom, adresse email, etc.),
- une information à certifier (clef publique)
- et la signature d'une autorité de confiance.

On peut y ajouter des informations additionnelles.

Certificats

Un **certificat** est une structure de données reliant
- l'identité d'une personne (nom, adresse email, etc.),
- une information à certifier (clef publique)
- et la signature d'une autorité de confiance.

On peut y ajouter des informations additionnelles.

Exemple. La norme X.509 est utilisée pour les certificats dans TLS/SSL (protocoles de sécurisation).

Certificats

Un **certificat** est une structure de données reliant
- l'identité d'une personne (nom, adresse email, etc.),
- une information à certifier (clef publique)
- et la signature d'une autorité de confiance.

On peut y ajouter des informations additionnelles.

Exemple. La norme X.509 est utilisée pour les certificats dans TLS/SSL (protocoles de sécurisation). La structure de données contient (dans le désordre) :

- L'identifiant du signataire et/ou du détenteur du certificat
- Algorithme de chiffrement
- Clé publique
- La signature de l'émetteur du certificat
- La version du certificat
- Le numéro de série
- Le nom de l'autorité de certification
- La date de début et de fin de validité
- L'objet de l'utilisation du certificat
- Les extensions au certificat
- L'algorithme de signature

Protocole de création/vérification d'un certificat

L'**autorité de certification** (CA) émet publiquement un algorithme Verif_{CA} de vérification de sa signature. Elle garde secrètement l'algorithme de signature associé Sign_{CA}.

Protocole de création/vérification d'un certificat

L'**autorité de certification** (CA) émet publiquement un algorithme Verif_{CA} de vérification de sa signature. Elle garde secrètement l'algorithme de signature associé Sign_{CA}.

Supposons qu'Alice veuille **faire certifier une clé publique**. Deux cas possibles :

Protocole de création/vérification d'un certificat

L'**autorité de certification** (CA) émet publiquement un algorithme Verif_{CA} de vérification de sa signature. Elle garde secrètement l'algorithme de signature associé Sign_{CA}.

Supposons qu'Alice veuille **faire certifier une clé publique**. Deux cas possibles :

1. C'est l'autorité de certification (CA) qui crée la paire de clefs. Puis la CA transmet la clé privée $\mathtt{sk}_A$ à Alice par un moyen sécurisé, authentifié et privé.

Protocole de création/vérification d'un certificat

L'**autorité de certification** (CA) émet publiquement un algorithme Verif_{CA} de vérification de sa signature. Elle garde secrètement l'algorithme de signature associé Sign_{CA}.

Supposons qu'Alice veuille **faire certifier une clé publique**. Deux cas possibles :

1. C'est l'autorité de certification (CA) qui crée la paire de clefs. Puis la CA transmet la clé privée $\mathtt{sk}_A$ à Alice par un moyen sécurisé, authentifié et privé.
2. C'est Alice qui crée la paire de clefs. Puis elle transmet la clef publique $\mathtt{pk}_A$ à la CA par un moyen sécurisé. Alice fournit également une **preuve de connaissance de la clef privée**.

Protocole de création/vérification d'un certificat

L'**autorité de certification** (CA) émet publiquement un algorithme Verif_{CA} de vérification de sa signature. Elle garde secrètement l'algorithme de signature associé Sign_{CA}.

Supposons qu'Alice veuille **faire certifier une clé publique**. Deux cas possibles :

1. C'est l'autorité de certification (CA) qui crée la paire de clefs. Puis la CA transmet la clé privée $\mathtt{sk}_A$ à Alice par un moyen sécurisé, authentifié et privé.
2. C'est Alice qui crée la paire de clefs. Puis elle transmet la clef publique $\mathtt{pk}_A$ à la CA par un moyen sécurisé. Alice fournit également une **preuve de connaissance de la clef privée**.

Dans tous les cas, la clef publique $\mathtt{pk}_A$ est ensuite **certifiée** par la CA. Si $\mathtt{ID}(\text{Alice})$ représente l'identité d'Alice :

$$\begin{cases} s = \text{Sign}_{\text{CA}}(\mathtt{ID}(\text{Alice}) \,\|\, \mathtt{pk}_A) \\ \mathtt{CERT}(\text{Alice}, \mathtt{pk}_A) = \left[\, \mathtt{ID}(\text{Alice}) \,\|\, \mathtt{pk}_A \,\|\, s \,\right] \end{cases}$$

Protocole de création/vérification d'un certificat

L'**autorité de certification** (CA) émet publiquement un algorithme Verif_{CA} de vérification de sa signature. Elle garde secrètement l'algorithme de signature associé Sign_{CA}.

Supposons qu'Alice veuille **faire certifier une clé publique**. Deux cas possibles :

1. C'est l'autorité de certification (CA) qui crée la paire de clefs. Puis la CA transmet la clé privée $\texttt{sk}_A$ à Alice par un moyen sécurisé, authentifié et privé.
2. C'est Alice qui crée la paire de clefs. Puis elle transmet la clef publique $\texttt{pk}_A$ à la CA par un moyen sécurisé. Alice fournit également une **preuve de connaissance de la clef privée**.

Dans tous les cas, la clef publique $\texttt{pk}_A$ est ensuite **certifiée** par la CA. Si $\texttt{ID}(\text{Alice})$ représente l'identité d'Alice :

$$\begin{cases} s = \text{Sign}_{\text{CA}}(\texttt{ID}(\text{Alice}) \,\|\, \texttt{pk}_A) \\ \texttt{CERT}(\text{Alice}, \texttt{pk}_A) = \left[\, \texttt{ID}(\text{Alice}) \,\|\, \texttt{pk}_A \,\|\, s \,\right] \end{cases}$$

Puis, lorsque Bob souhaite **vérifier la certification**, il vérifie simplement que

$$\text{Verif}_{\text{CA}}(s,\ \texttt{ID}(\text{Alice}) \,\|\, \texttt{pk}_A) = \texttt{true}$$

Démo !

Exemple en ligne

Plan

Une première proposition de chiffrement authentifié

Objectif. Alice veut envoyer un message ***m*** chiffré et **authentifié** à Bob (= Bob est convaincu que c'est bien Alice qui a produit le message).

Une première proposition de chiffrement authentifié

Objectif. Alice veut envoyer un message ***m*** chiffré et **authentifié** à Bob (= Bob est convaincu que c'est bien Alice qui a produit le message).

Notation.

- $\mathrm{Enc}_B/\mathrm{Dec}_B$ sont les algorithmes de chiffrement public / déchiffrement privé de Bob.
- $\mathrm{Sign}_A/\mathrm{Verif}_A$ sont les algorithmes de signature privé / vérification publique d'Alice.

Une première proposition de chiffrement authentifié

Objectif. Alice veut envoyer un message $\boldsymbol{m}$ chiffré et **authentifié** à Bob (= Bob est convaincu que c'est bien Alice qui a produit le message).

Notation.

- Enc_B/Dec_B sont les algorithmes de chiffrement public / déchiffrement privé de Bob.
- Sign_A/Verif_A sont les algorithmes de signature privé / vérification publique d'Alice.

Première idée (*sign-and-encrypt*) :

1. Alice signe $s = \mathsf{Sign}_A(\boldsymbol{m})$.
2. Alice calcule $c = \mathsf{Enc}_B(\boldsymbol{m} \,\|\, s)$ et envoie c à Bob.
3. Bob déchiffre $\mathsf{Dec}_B(c) = \boldsymbol{m} \,\|\, s$ puis vérifie que $\mathsf{Verif}_A(\boldsymbol{m}, s) = \texttt{true}$.

Une première proposition de chiffrement authentifié

Objectif. Alice veut envoyer un message $\boldsymbol{m}$ chiffré et **authentifié** à Bob (= Bob est convaincu que c'est bien Alice qui a produit le message).

Notation.

- Enc_B/Dec_B sont les algorithmes de chiffrement public / déchiffrement privé de Bob.
- Sign_A/Verif_A sont les algorithmes de signature privé / vérification publique d'Alice.

Première idée (*sign-and-encrypt*) :

1. Alice signe $s = \mathsf{Sign}_A(\boldsymbol{m})$.
2. Alice calcule $c = \mathsf{Enc}_B(\boldsymbol{m} \,\|\, s)$ et envoie c à Bob.
3. Bob déchiffre $\mathsf{Dec}_B(c) = \boldsymbol{m} \,\|\, s$ puis vérifie que $\mathsf{Verif}_A(\boldsymbol{m}, s) = \texttt{true}$.

Problème. Bob lui-même peut briser l'authentification du message, en **réutilisant la signature** :

Une première proposition de chiffrement authentifié

Objectif. Alice veut envoyer un message $\boldsymbol{m}$ chiffré et **authentifié** à Bob (= Bob est convaincu que c'est bien Alice qui a produit le message).

Notation.

- $\mathsf{Enc}_B/\mathsf{Dec}_B$ sont les algorithmes de chiffrement public / déchiffrement privé de Bob.
- $\mathsf{Sign}_A/\mathsf{Verif}_A$ sont les algorithmes de signature privé / vérification publique d'Alice.

Première idée (*sign-and-encrypt*) :

1. Alice signe $s = \mathsf{Sign}_A(\boldsymbol{m})$.
2. Alice calcule $c = \mathsf{Enc}_B(\boldsymbol{m} \,\|\, s)$ et envoie c à Bob.
3. Bob déchiffre $\mathsf{Dec}_B(c) = \boldsymbol{m} \,\|\, s$ puis vérifie que $\mathsf{Verif}_A(\boldsymbol{m}, s) = \texttt{true}$.

Problème. Bob lui-même peut briser l'authentification du message, en **réutilisant la signature** :

1. Bob déchiffre $\mathsf{Dec}_B(c) = \boldsymbol{m} \,\|\, s$, puis chiffre $\boldsymbol{m} \,\|\, s$ avec l'algorithme de chiffrement Enc_C public associé à Charlie, une autre personne.
2. Charlie croit que Alice lui a envoyé un message $\mathsf{Enc}_C(\boldsymbol{m} \,\|\, s)$, car après déchiffrement la signature s est celle d'Alice

Une seconde proposition

Seconde idée (*encrypt-and-sign*) :

1. Alice calcule $c = \mathrm{Enc}_B(\boldsymbol{m})$.
2. Alice signe $s = \mathrm{Sign}_A(c)$ et envoie (c, s) à Bob.
3. Bob vérifie la signature $\mathrm{Verif}_B(c, s) = \texttt{true}$ puis déchiffre $\mathrm{Dec}_B(c) = \boldsymbol{m}$.

Une seconde proposition

Seconde idée (*encrypt-and-sign*) :

1. Alice calcule $c = \mathrm{Enc}_B(\boldsymbol{m})$.
2. Alice signe $s = \mathrm{Sign}_A(c)$ et envoie (c, s) à Bob.
3. Bob vérifie la signature $\mathrm{Verif}_B(c, s) = \texttt{true}$ puis déchiffre $\mathrm{Dec}_B(c) = \boldsymbol{m}$.

Problème. Oscar peut procéder à une attaque par le milieu :

1. Oscar observe (c, s) et remplace s par sa signature $s' = \mathrm{Sign}_O(c)$.
2. Oscar envoie (c, s') à Bob et lui fait croire que le message $\boldsymbol{m}$ est le sien...

Une seconde proposition

Seconde idée (*encrypt-and-sign*) :

1. Alice calcule $c = \mathrm{Enc}_B(\boldsymbol{m})$.
2. Alice signe $s = \mathrm{Sign}_A(c)$ et envoie (c, s) à Bob.
3. Bob vérifie la signature $\mathrm{Verif}_B(c, s) = \texttt{true}$ puis déchiffre $\mathrm{Dec}_B(c) = \boldsymbol{m}$.

Problème. Oscar peut procéder à une attaque par le milieu :

1. Oscar observe (c, s) et remplace s par sa signature $s' = \mathrm{Sign}_O(c)$.
2. Oscar envoie (c, s') à Bob et lui fait croire que le message $\boldsymbol{m}$ est le sien...

Remarque : dans ce cas, Oscar ne connaît pas le message $\boldsymbol{m}$.

Solution 1

À Alice et Bob sont associés des identifiants publics $\texttt{ID}_A$ et $\texttt{ID}_B$.

PROTOCOLE « SIGN-THEN-ENCRYPT »

Solution 1

À Alice et Bob sont associés des identifiants publics $\texttt{ID}_A$ et $\texttt{ID}_B$.

PROTOCOLE « SIGN-THEN-ENCRYPT »

Pour envoyer un message authentifié $\boldsymbol{m}$ à Bob, Alice :

1. calcule une signature $s = \text{Sign}_A(\boldsymbol{m} \| \texttt{ID}_B)$
2. calcule un chiffré $c = \text{Enc}_B(\boldsymbol{m} \| s \| \texttt{ID}_A)$

Solution 1

À Alice et Bob sont associés des identifiants publics $\texttt{ID}_A$ et $\texttt{ID}_B$.

PROTOCOLE « SIGN-THEN-ENCRYPT »

Pour envoyer un message authentifié $\boldsymbol{m}$ à Bob, Alice :

1. calcule une signature $s = \text{Sign}_A(\boldsymbol{m} \,\|\, \texttt{ID}_B)$
2. calcule un chiffré $c = \text{Enc}_B(\boldsymbol{m} \,\|\, s \,\|\, \texttt{ID}_A)$

Pour déchiffrer et authentifier le message reçu c, Bob :

1. déchiffre $\text{Dec}_B(c) = \boldsymbol{m} \,\|\, s \,\|\, \texttt{ID}_A$
2. reconnaît l'identité d'Alice dans le message déchiffré
3. vérifie la signature associée est correcte $\text{Verif}_A(\boldsymbol{m} \,\|\, \texttt{ID}_B,\ s) = \texttt{true}$.

Solution 1

À Alice et Bob sont associés des identifiants publics $\texttt{ID}_A$ et $\texttt{ID}_B$.

Protocole « sign-then-encrypt »

Pour envoyer un message authentifié $\boldsymbol{m}$ à Bob, Alice :

1. calcule une signature $s = \text{Sign}_A(\boldsymbol{m} \,\|\, \texttt{ID}_B)$
2. calcule un chiffré $c = \text{Enc}_B(\boldsymbol{m} \,\|\, s \,\|\, \texttt{ID}_A)$

Pour déchiffrer et authentifier le message reçu c, Bob :

1. déchiffre $\text{Dec}_B(c) = \boldsymbol{m} \,\|\, s \,\|\, \texttt{ID}_A$
2. reconnaît l'identité d'Alice dans le message déchiffré
3. vérifie la signature associée est correcte $\text{Verif}_A(\boldsymbol{m} \,\|\, \texttt{ID}_B,\ s) = \texttt{true}$.

La première attaque ne fonctionne plus, car Alice a lié sa signature s à l'identité du destinataire, Bob.

Solution 2

À Alice et Bob sont associés des identifiants publics $\texttt{ID}_A$ et $\texttt{ID}_B$.

PROTOCOLE « ENCRYPT-THEN-SIGN »

Solution 2

À Alice et Bob sont associés des identifiants publics $\texttt{ID}_A$ et $\texttt{ID}_B$.

PROTOCOLE « ENCRYPT-THEN-SIGN »

Pour envoyer un message authentifié $\boldsymbol{m}$ à Bob, Alice :

1. calcule un chiffré $c = \text{Enc}_B(\boldsymbol{m} \,\|\, \texttt{ID}_A)$,
2. signe $s = \text{Sign}_A(c \,\|\, \texttt{ID}_B)$ et envoie $(c, s, \texttt{ID}_A)$ à Bob.

Solution 2

À Alice et Bob sont associés des identifiants publics $\texttt{ID}_A$ et $\texttt{ID}_B$.

PROTOCOLE « ENCRYPT-THEN-SIGN »

Pour envoyer un message authentifié $\boldsymbol{m}$ à Bob, Alice :

1. calcule un chiffré $c = \mathrm{Enc}_B(\boldsymbol{m} \,\|\, \texttt{ID}_A)$,
2. signe $s = \mathrm{Sign}_A(c \,\|\, \texttt{ID}_B)$ et envoie $(c, s, \texttt{ID}_A)$ à Bob.

Pour déchiffrer et authentifier le message reçu, Bob :

1. vérifie que $\mathrm{Verif}_A(c \,\|\, \texttt{ID}_B,\ s)$,
2. déchiffre c et obtient $\boldsymbol{m}$ et l'identité $\texttt{ID}_A$,
3. vérifie que l'identité correspond à ce qu'il a reçu précédemment.

Solution 2

À Alice et Bob sont associés des identifiants publics $\texttt{ID}_A$ et $\texttt{ID}_B$.

PROTOCOLE « ENCRYPT-THEN-SIGN »

Pour envoyer un message authentifié $\boldsymbol{m}$ à Bob, Alice :

1. calcule un chiffré $c = \text{Enc}_B(\boldsymbol{m} \,\|\, \texttt{ID}_A)$,
2. signe $s = \text{Sign}_A(c \,\|\, \texttt{ID}_B)$ et envoie $(c, s, \texttt{ID}_A)$ à Bob.

Pour déchiffrer et authentifier le message reçu, Bob :

1. vérifie que $\text{Verif}_A(c \,\|\, \texttt{ID}_B,\ s)$,
2. déchiffre c et obtient $\boldsymbol{m}$ et l'identité $\texttt{ID}_A$,
3. vérifie que l'identité correspond à ce qu'il a reçu précédemment.

La seconde attaque ne fonctionne plus, car Oscar ne peut pas intégrer son identité personnelle ID_O au chiffré c afin de prétendre avoir chiffré le message $\boldsymbol{m}$.

Plan

Plan

Schémas d'identification : motivation

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

Schémas d'identification : motivation

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)

Schémas d'identification : motivation

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)

Schémas d'identification : motivation

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**.

Schémas d'identification : motivation

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.)

Schémas d'identification : motivation

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.

Schémas d'identification : motivation

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.

Schémas d'identification : définitions

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

Schémas d'identification : définitions

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.

Schémas d'identification : définitions

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.

Schémas d'identification : définitions

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

Schémas d'identification : définitions

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**.

Schémas d'identification : définitions

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

Schémas d'identification : définitions

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. 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 est **valide** si tout prouveur honnête convainc le vérifieur de son identité avec probabilité 1.

Schémas d'identification : définitions

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. 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 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.

Schémas d'identification : modèle de sécurité

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.

Schémas d'identification : modèle de sécurité

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.

Schémas d'identification : modèle de sécurité

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 $[\mathcal{P} \longleftrightarrow \mathcal{V}]$.

Schémas d'identification : modèle de sécurité

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 $\widetilde{\mathcal{P}}$. L'attaque est réussie si, lors d'une exécution de $[\widetilde{\mathcal{P}} \longleftrightarrow \mathcal{V}']$ avec un vérifieur $\mathcal{V}'$, la probabilité que le vérifieur $\mathcal{V}'$ soit convaincu d'interagir avec $\mathcal{P}$ est non-négligeable.

Schémas d'identification : modèle de sécurité

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 $\widetilde{\mathcal{P}}$. L'attaque est réussie si, lors d'une exécution de $[\widetilde{\mathcal{P}} \longleftrightarrow \mathcal{V}']$ 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 :

Schémas d'identification : modèle de sécurité

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 $\widetilde{\mathcal{P}}$. L'attaque est réussie si, lors d'une exécution de $[\widetilde{\mathcal{P}} \longleftrightarrow \mathcal{V}']$ 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.

Schémas d'identification : modèle de sécurité

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 $\widetilde{\mathcal{P}}$. L'attaque est réussie si, lors d'une exécution de $[\widetilde{\mathcal{P}} \longleftrightarrow \mathcal{V}']$ 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).

Plan

Une tentative de construction

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

Une tentative de construction

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 !

Une tentative de construction

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 $(\mathtt{pk}_A, \mathtt{sk}_A)$ certifiée, pour un schéma de signature numérique quelconque.

Une tentative de construction

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 $(\mathtt{pk}_A, \mathtt{sk}_A)$ certifiée, pour un schéma de signature numérique quelconque.

On associe également à Alice son identité $\mathtt{id}_A$.

Une tentative de construction

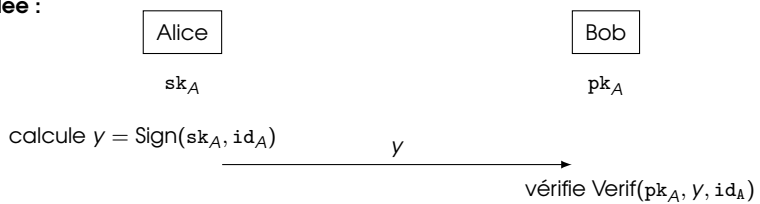
Théorème (informel). Il est possible de construire un schéma d'identification sûr contre des adversaires actifs, **à partir** d'une signature numérique EUF-CMA.

Essayons !

Mise en place. On suppose que Alice est munie d'une paire de clefs $(\mathtt{pk}_A, \mathtt{sk}_A)$ certifiée, pour un schéma de signature numérique quelconque.

On associe également à Alice son identité $\mathtt{id}_A$.

Une première idée :



Une tentative de construction

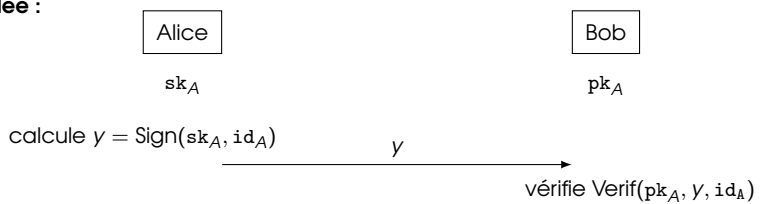
Théorème (informel). Il est possible de construire un schéma d'identification sûr contre des adversaires actifs, **à partir** d'une signature numérique EUF-CMA.

Essayons !

Mise en place. On suppose que Alice est munie d'une paire de clefs $(\mathtt{pk}_A, \mathtt{sk}_A)$ certifiée, pour un schéma de signature numérique quelconque.

On associe également à Alice son identité $\mathtt{id}_A$.

Une première idée :



Problème. Attaque par rejeu.

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

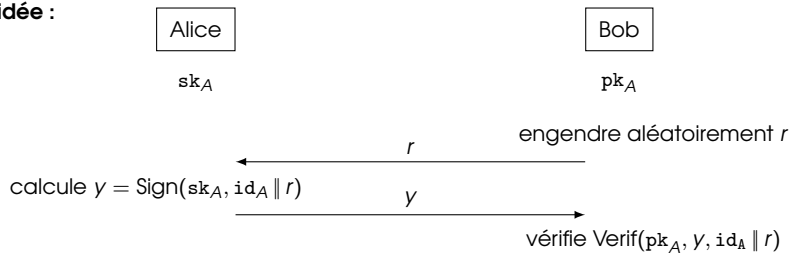
Identification à partir d'une signature

Mise en place. On suppose que Alice est munie d'une paire de clefs $(\mathtt{pk}_A, \mathtt{sk}_A)$ certifiée, pour un schéma de signature numérique quelconque. On associe également à Alice son identité $\mathtt{id}_A$.

Identification à partir d'une signature

Mise en place. On suppose que Alice est munie d'une paire de clefs $(\mathtt{pk}_A, \mathtt{sk}_A)$ certifiée, pour un schéma de signature numérique quelconque. On associe également à Alice son identité $\mathtt{id}_A$.

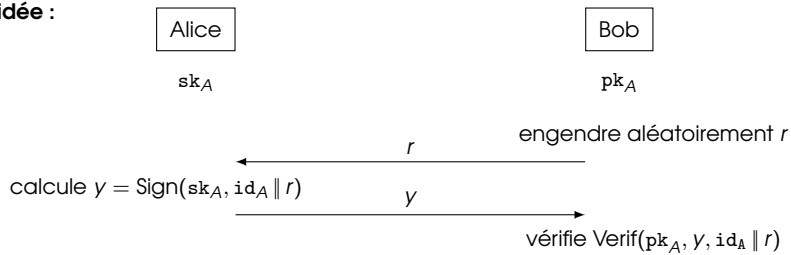
Une seconde idée :



Identification à partir d'une signature

Mise en place. On suppose que Alice est munie d'une paire de clefs $(\mathtt{pk}_A, \mathtt{sk}_A)$ certifiée, pour un schéma de signature numérique quelconque. On associe également à Alice son identité $\mathtt{id}_A$.

Une seconde idée :

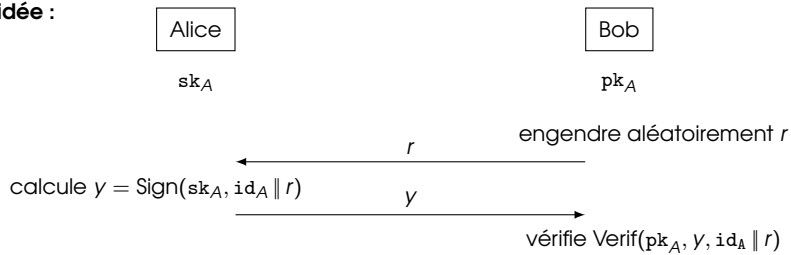


La valeur de y ne peut pas être réutilisée, car elle dépend du "challenge" r engendré par Bob.

Identification à partir d'une signature

Mise en place. On suppose que Alice est munie d'une paire de clefs $(\mathtt{pk}_A, \mathtt{sk}_A)$ certifiée, pour un schéma de signature numérique quelconque. On associe également à Alice son identité $\mathtt{id}_A$.

Une seconde idée :



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

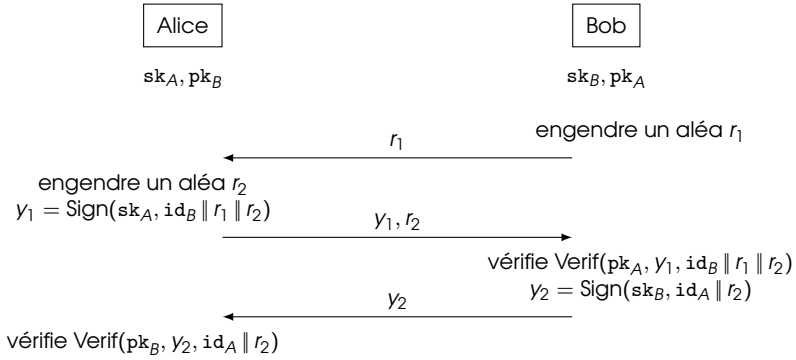
Théorème (informel). Il est possible de construire un schéma d'identification sûr contre des adversaires actifs, **à partir** d'une signature numérique EUF-CMA.

Identification mutuelle

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

Identification mutuelle

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



Plan

Schéma d'identification de Schnorr

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

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 $\mathbb{F}_p^\times$
- $a \in \{0, \ldots, q-1\}$ est gardé secrètement par Alice.

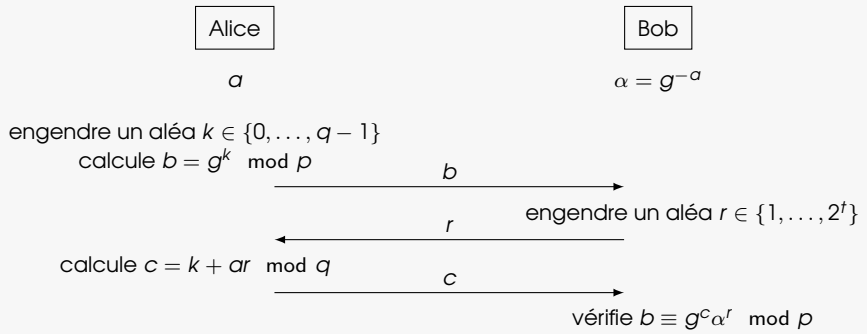
Schéma d'identification de Schnorr

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

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

Sécurité du schéma d'identification de Schnorr

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

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

Sécurité. On peut démontrer le résultat suivant.

Sécurité du schéma d'identification de Schnorr

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

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

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 $O(1/\varepsilon)$.

Sécurité du schéma d'identification de Schnorr

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

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

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** ».

Sécurité du schéma d'identification de Schnorr

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

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

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 ($g^k \mod p$) à Bob.

Sécurité du schéma d'identification de Schnorr

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

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

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 ($g^k \mod p$) à Bob.
2. Phase de **défi**/**challenge** : Bob construit un défi aléatoire (c) auquel Alice doit répondre. La résolution de ce défi dépend de l'engagement d'Alice.

Sécurité du schéma d'identification de Schnorr

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

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

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 ($g^k \mod p$) à Bob.
2. Phase de **défi**/**challenge** : Bob construit un défi aléatoire (c) 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.

Sécurité du schéma d'identification de Schnorr

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

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

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 ($g^k \mod p$) à Bob.
2. Phase de **défi**/**challenge** : Bob construit un défi aléatoire (c) 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.

Exemple-jouet pour Schnorr

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

Exemple-jouet pour 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.

Exemple-jouet pour 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.

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

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

Exemple-jouet pour 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.

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

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

Engagement. Alice choisit $k = 543$ et envoie

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

Exemple-jouet pour 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.

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

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

Engagement. Alice choisit $k = 543$ et envoie

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

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

Exemple-jouet pour 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.

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

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

Engagement. Alice choisit $k = 543$ et envoie

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

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

Réponse. Alice calcule

$$c = k + ar = 543 + 755 \times 1000 \equiv 851 \mod 1031$$

Exemple-jouet pour 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.

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

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

Engagement. Alice choisit $k = 543$ et envoie

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

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

Réponse. Alice calcule

$$c = k + ar = 543 + 755 \times 1000 \equiv 851 \mod 1031$$

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

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

Plan

Transformation de Fiat–Shamir

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**.

Transformation de Fiat–Shamir

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**.

Transformation de Fiat–Shamir

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 :

Transformation de Fiat–Shamir

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

Transformation de Fiat–Shamir

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 $\boldsymbol{m}$

Transformation de Fiat–Shamir

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 $\boldsymbol{m}$
- le prouveur calcule ensuite une réponse c au défi $r = H(x, \boldsymbol{m})$, qui constitue alors la signature de $\boldsymbol{m}$.

Transformation de Fiat–Shamir

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 $\boldsymbol{m}$
- le prouveur calcule ensuite une réponse c au défi $r = H(x, \boldsymbol{m})$, qui constitue alors la signature de $\boldsymbol{m}$.

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

Transformation de Fiat–Shamir

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 $\boldsymbol{m}$
- le prouveur calcule ensuite une réponse c au défi $r = H(x, \boldsymbol{m})$, qui constitue alors la signature de $\boldsymbol{m}$.

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-divulgation** du protocole interactif.

Signature de Schnorr

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 $\mathtt{pk} = \alpha$, la clé privée est $\mathtt{sk} = a$.

Signature de Schnorr

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 $\mathtt{pk} = \alpha$, la clé privée est $\mathtt{sk} = a$.

SIGNATURE DE SCHNORR : Sign($\boldsymbol{m}$, $\mathtt{sk}$)

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

Signature de Schnorr

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 $\mathtt{pk} = \alpha$, la clé privée est $\mathtt{sk} = a$.

Signature de Schnorr : Sign($\boldsymbol{m}$, $\mathtt{sk}$)

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

Signature de Schnorr : Verif($\boldsymbol{m}$, s, $\mathtt{pk}$)

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

Signature de Schnorr

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 $\mathtt{pk} = \alpha$, la clé privée est $\mathtt{sk} = a$.

SIGNATURE DE SCHNORR : Sign($\boldsymbol{m}$, $\mathtt{sk}$)

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

SIGNATURE DE SCHNORR : Verif($\boldsymbol{m}$, s, $\mathtt{pk}$)

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

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