D31: Protocoles Cryptographiques

Schémas d'authentification

Nicolas Méloni

Master 2: 1er semestre
(2013/2014)

Introduction

Objectif

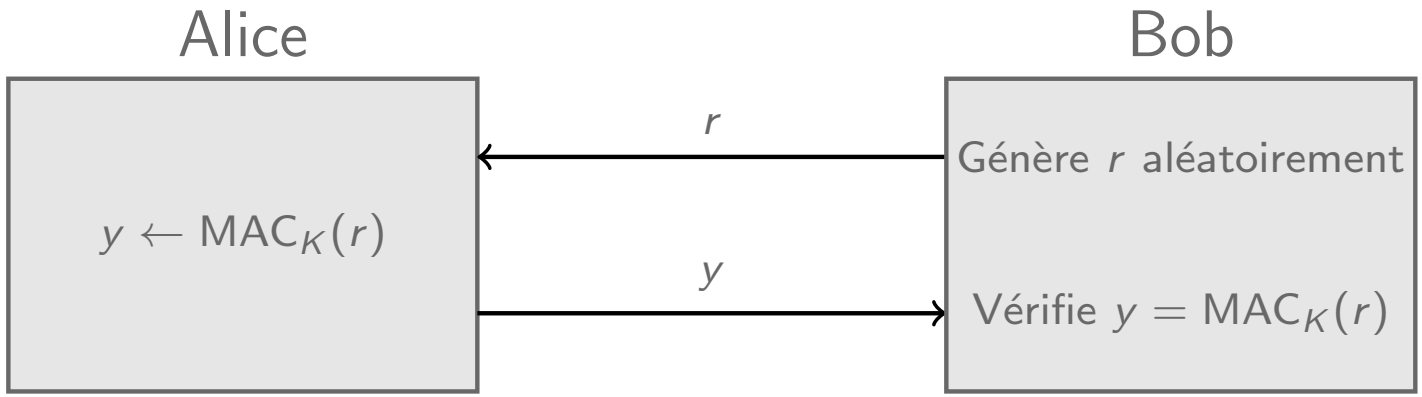
- Fournir un mécanisme d'identification auprès d'un entité extérieure (distributeur, site web, serveur informatique etc)

Dans la vie courante

- Apparence physique
- Papiers d'identité
- Codes secrets (carte bleu, PIN, salle U026 etc)

Dans le cadre de la cryptographie symétrique

- Protocole challenge et réponse:

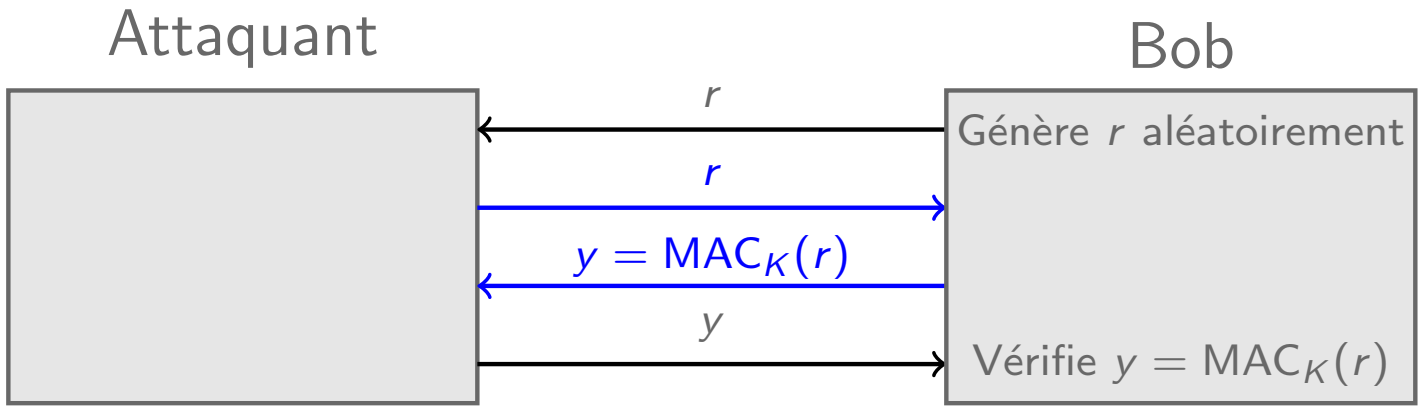


Présupposés

- Bob possède un générateur aléatoire
- La fonction MAC est sure
- Alice et Bob partagent déjà une clé K (et sont les seuls)

Dans le cadre de la cryptographie symétrique

- Attaque par session parallèle:

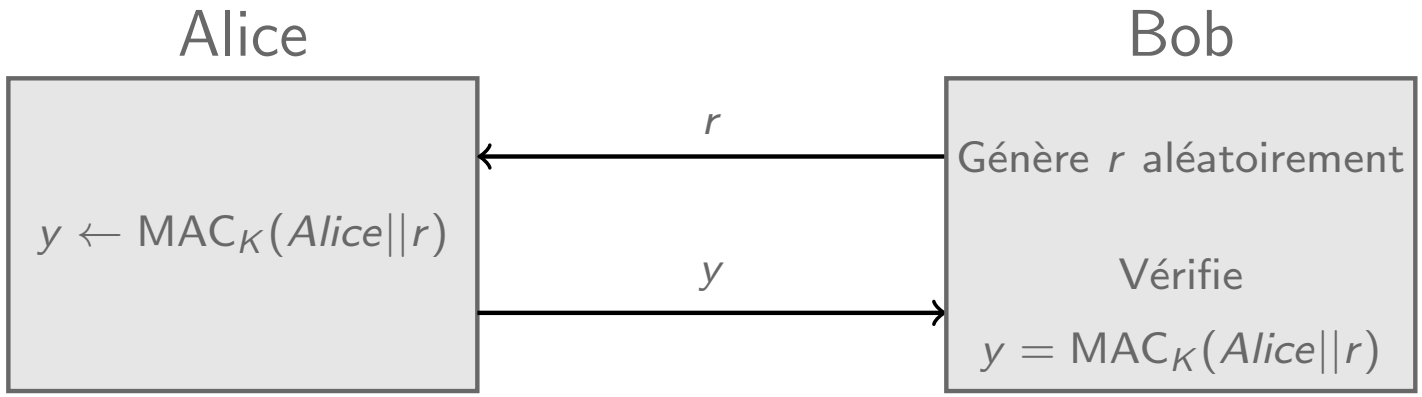


Remarques

- Alice ne joue aucune rôle
- L'attaquant n'effectue aucun calcul

Dans le cadre de la cryptographie symétrique

- Protocole challenge et réponse sécurisé:

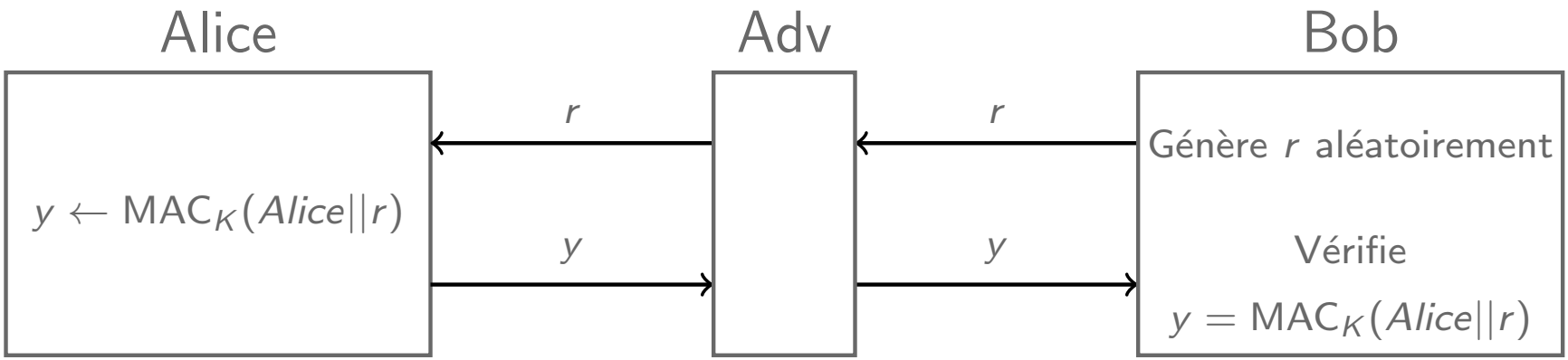


Remarques

- Un attaquant peut toujours s'immisser dans l'échange
- Il n'a aucun moyen d'obtenir de Bob la valeur $\mathrm{MAC}_K(Alice||r)$

Modèle d'attaque

- Intru du milieu



Remarque

- L'attaquant n'est pas actif, il se comporte comme un noeud du réseau.

Modèle d'attaque

Objectif

- Être accepté dans une session où l'attaquant est actif.

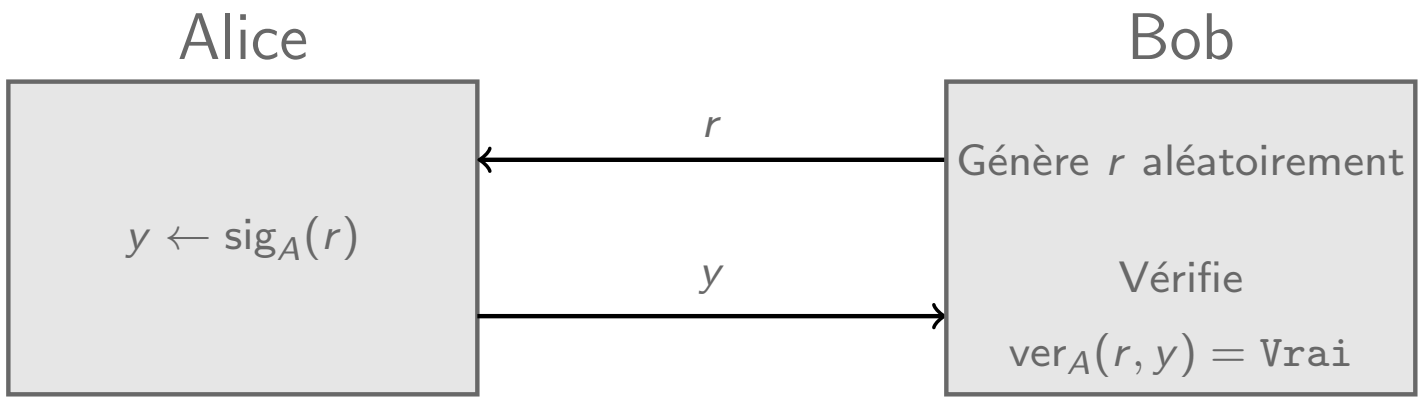
Conditions d'activité d'un attaquant

- Ajouter un nouveau message dans le canal,
- modififer un message du canal,
- détourner un message de son véritable destinataire.

Authentification et crypto à clé publique

Principe:

- Il n'est plus nécessaire de supposer que Alice et Bob partagent un secret K.
- On remplace les MAC par des primitives de signature numérique:



Authentification et crypto à clé publique

- Protocole d'authentification mutuelle:

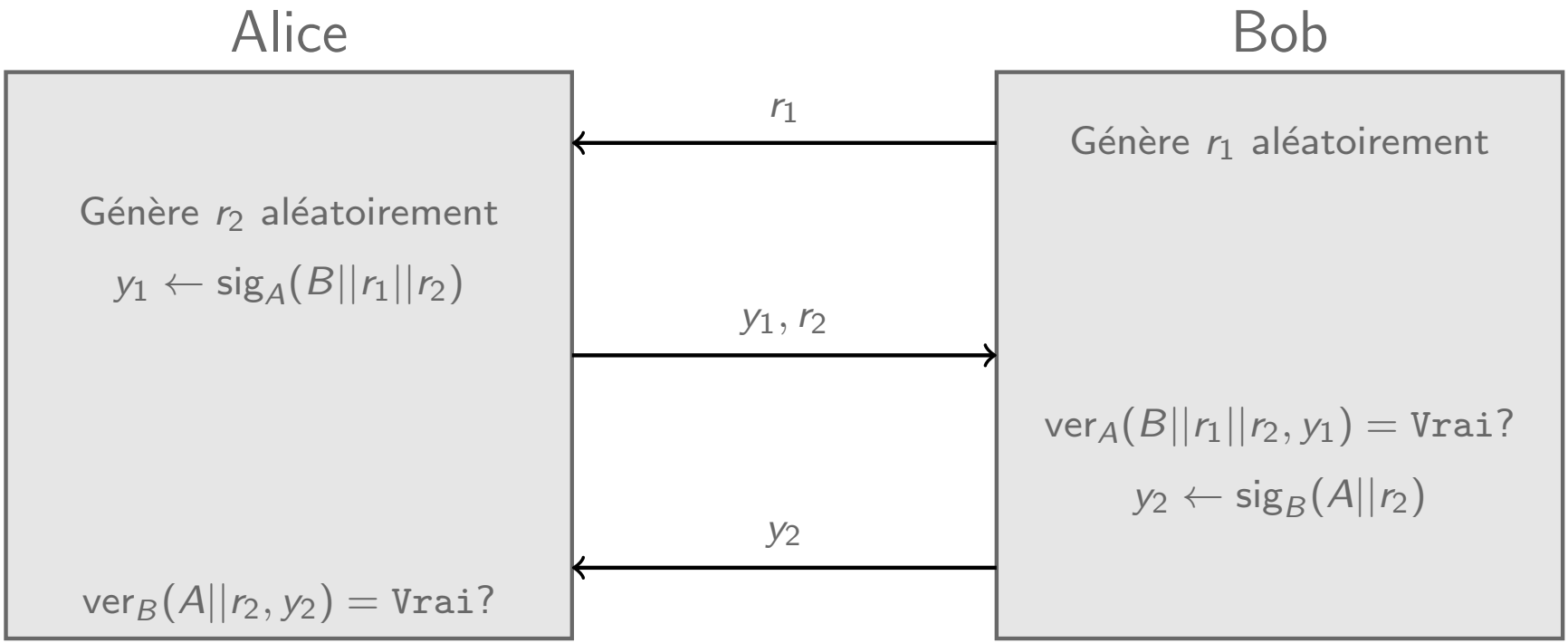


Schéma d'identification de Schnorr

Idée générale

- Concevoir un protocole d'authentification "from scratch" au lieu d'utiliser les signatures numériques

Intérêt

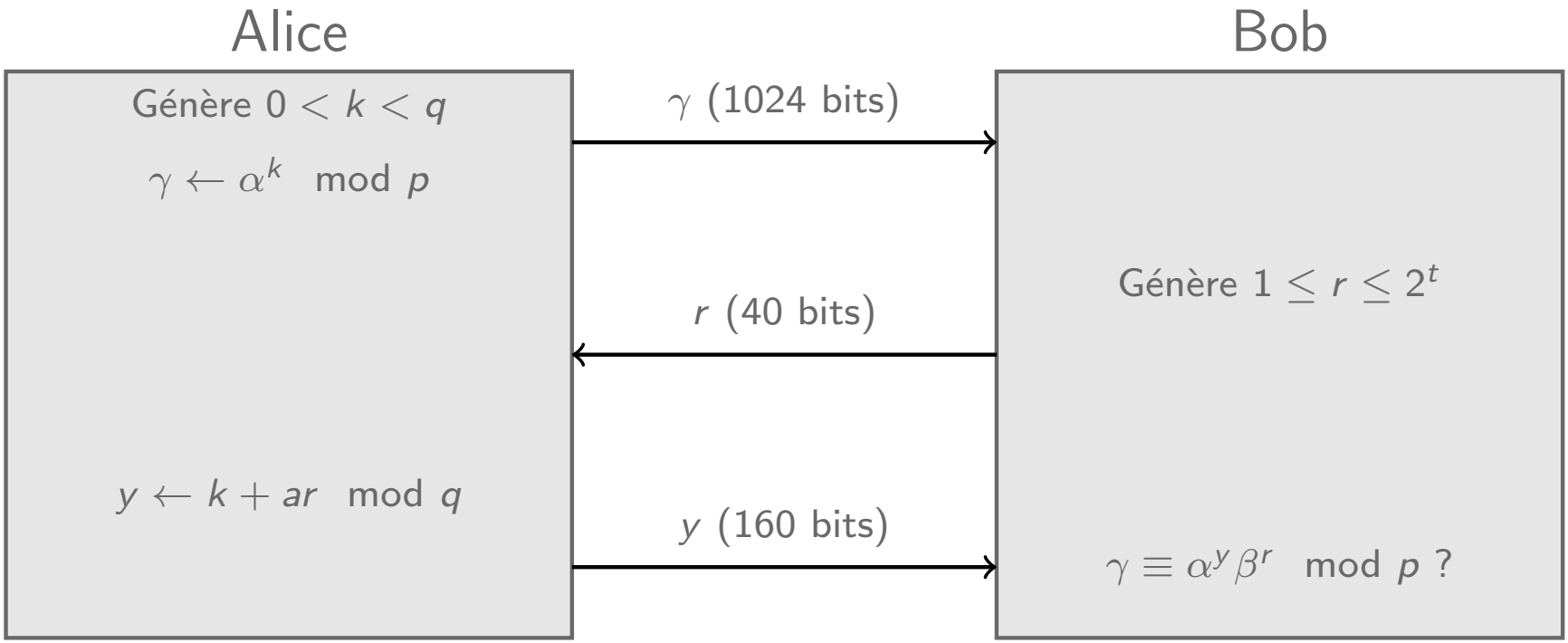
- Schémas plus efficace / moins couteux

Schéma d'identification de Schnorr

Paramètres:

- p et q deux nombres premiers tels que $q|p-1$
- $\alpha \in \mathbb{Z}_p^*$ un élément d'ordre q et $\beta = \alpha^{-a} \mod p$
- t un paramètre de sécurité tel que $q > 2^t$
- p, q, α et t sont publics et a est la clé privée de Alice

Schéma d'identification de Schnorr



Sécurité des schémas d'authentification

Pour être considéré comme sûr, un schéma d'authentification doit remplir deux critères:

- assurer au vérificateur (Bob) que le fournisseur de preuve (Alice) a bien la connaissance d'un secret (on parle de "*proof of knowledge*")
- ne divulguer aucune autre information que la preuve de la connaissance du secret (on parle de "*zero-knowledge proof*")

Sécurité des schémas d'authentification

Proof of Knowledge

- Consistence: si le fournisseur de preuve connait le secret, le vérificateur doit toujours accepter l'authentification
- Solidité: tromper le vérificateur doit être équivalent à connaître le secret

Zero-knowledge proof

- Le vérificateur n'obtient aucune information sur la valeur du secret après un nombre donné de sessions. En particulier, il doit lui être aussi difficile de calculer ce secret avant qu'après les sessions d'authentification.

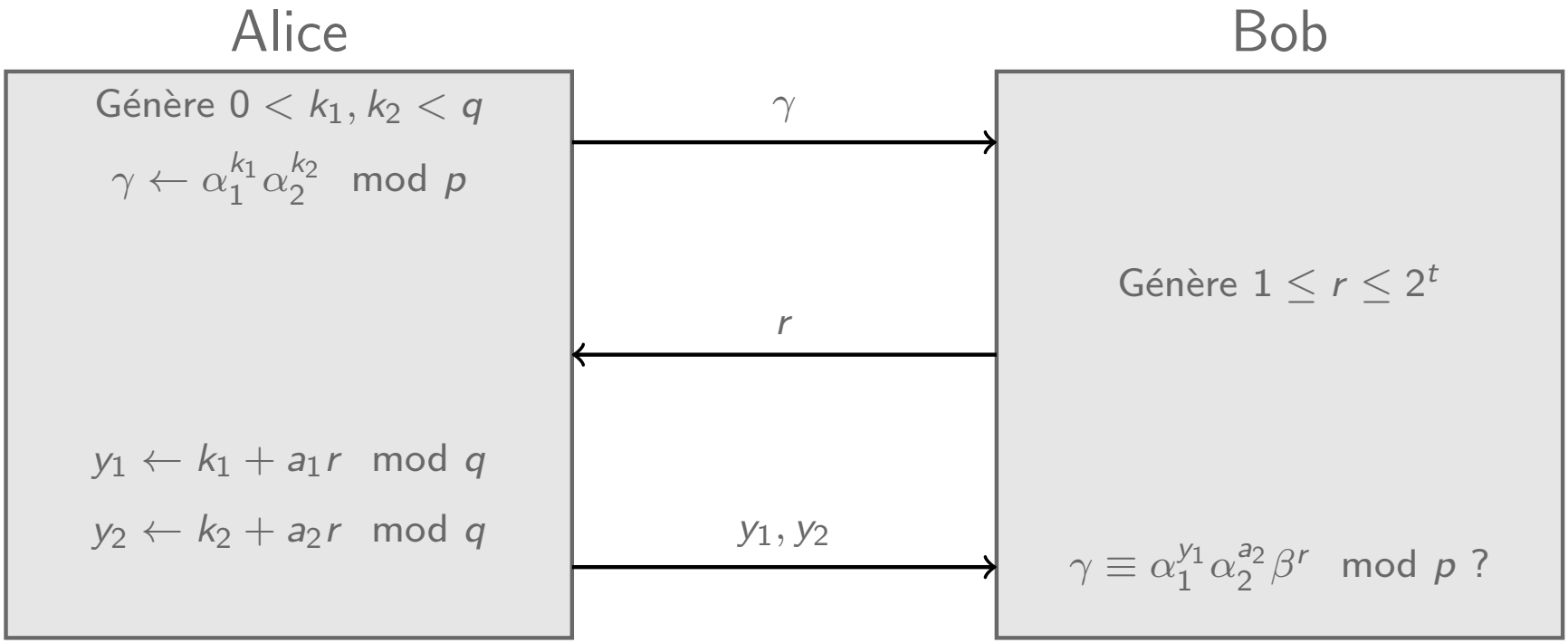
Schéma d'identification d'Okamoto

- C'est un schéma de Schnorr modifié. Son principal intêret est qu'il peut être prouvé calculatoirement sûr face à un vérificateur quelconque.

Paramètres

- p et q deux nombres premiers tels que $q|p-1$
- $\alpha_1, \alpha_2 \in \mathbb{Z}_p^*$ deux éléments d'ordre q tels $\alpha_2 = \alpha_1^c \mod p$, *choisis par une autorité de confiance*
- a_1, a_2 tels que $\beta = \alpha_1^{-a_1}\alpha_2^{-a_2} \mod p$
- t un paramètre de sécurité tel que $q > 2^t$
- $p, q, \alpha_1, \alpha_2, \beta$ et t sont publics et (a_1, a_2) est la clé privée d'Alice
- c est un paramètre secret global

Shéma d'identification d'Okamoto



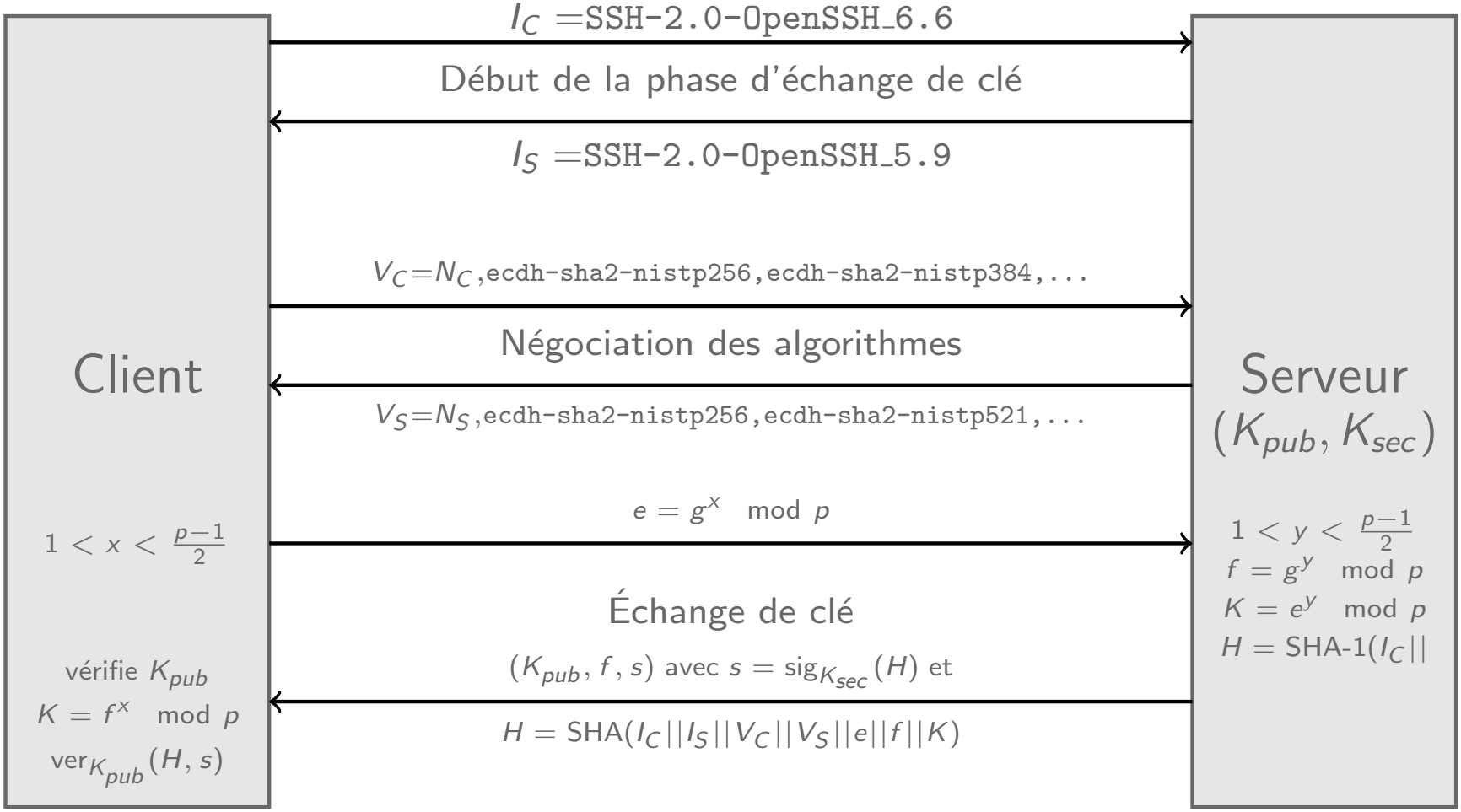
Protocole ssh

Objectifs

Garantir (entre autre)

- la confidentialité
- l'intégrité
- l'authentification

Protocole ssh



Protocole ssh

Résultat de la phase d'échange

- L'échange de clé de DH crée une clé de chiffrement commune K
- La signature du haché de la session permet d'authentifier le serveur.

Attention

- Encore faut il être sur de l'origine de la clé K_{pub}!

Protocole ssh

```
> cat .ssh/known_hosts
>
```

```
> ssh maitinfo1
The authenticity of host 'maitinfo1 (10.9.185.217)' can't be established.
ECDSA key fingerprint is f4:5c:ff:9a:14:ad:1d:33:df:da:99:65:59:81:4d:b0.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'maitinfo1,10.9.185.217' (ECDSA) to the list
of known hosts.
```

```
> cat .ssh/known_hosts
|1|aD4DsChlnysH5hWecH2rrLdMB7w=|WYpV359d0dz8EJX3DxmNsPrM6vU= ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBO3phRus8CoMAhSF87WfZ4Ao8Ply
UtpVQmM+NEmYiKol5/KFLsLFjOwAHNQNrKtbz1qrYECicoLYyvW/8g5bDzs=
|1|OQNNWKLZ07wlcu14Fc3nmp7YD9U=|+xG+iCltV31JwsJEs01zRli/nvk= ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBO3phRus8CoMAhSF87WfZ4Ao8Ply
UtpVQmM+NEmYiKol5/KFLsLFjOwAHNQNrKtbz1qrYECicoLYyvW/8g5bDzs=
```

Authentification ssh

Par mot de passe

- L'utilisateur envoie simplement le mot de passe associé à son compte utilisateur. L'ensemble des données échangées est chiffré à l'aide de la clé précédent échangée.

Par signature numérique

- L'utilisateur génère un couple clé publique/privé grâce à la commande `ssh-keygen`
- La clé publique est déposé dans le répertoire de travail de l'utilisateur
- Pour s'authentifier le poste utilisateur envoie une signature au serveur constituée,principalement, de l'identifiant de session et du nom d'utilisateur.