D31: Protocoles Cryptographiques

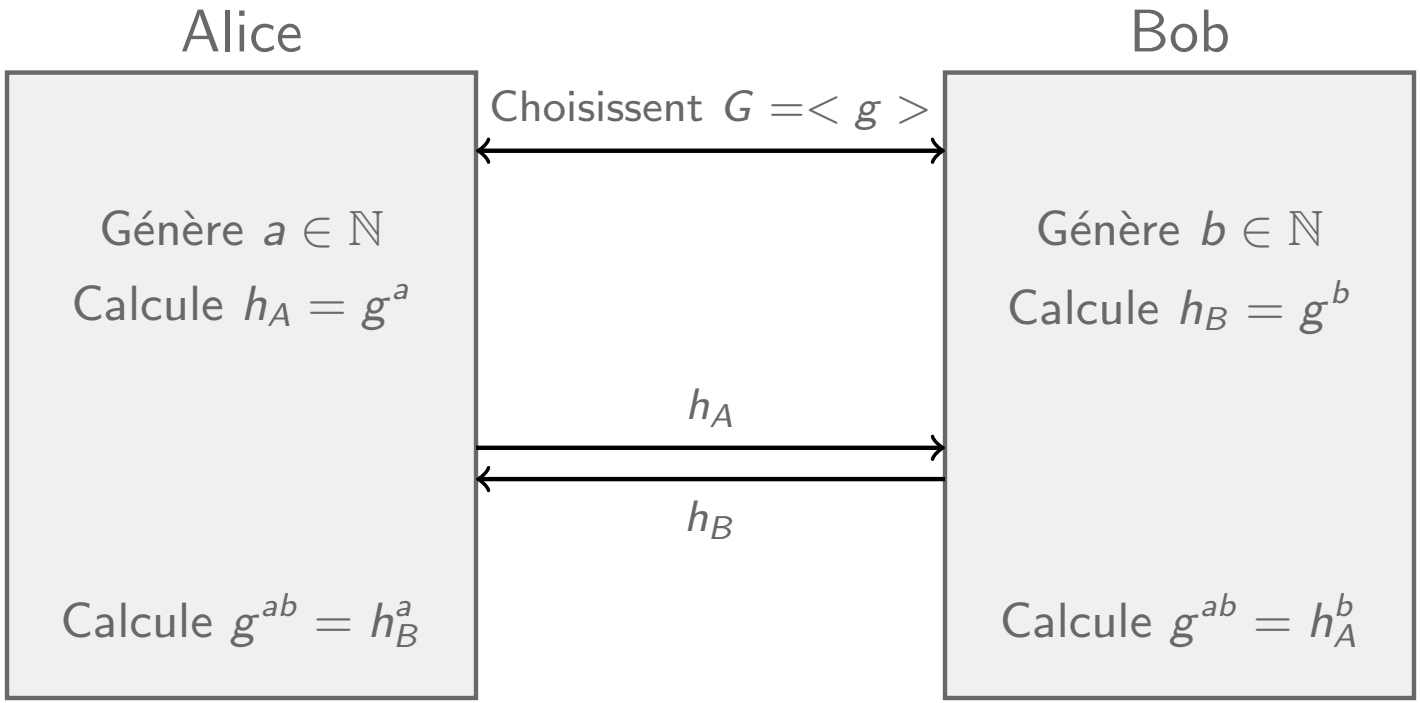
Certificats et échange de clés

Nicolas Méloni

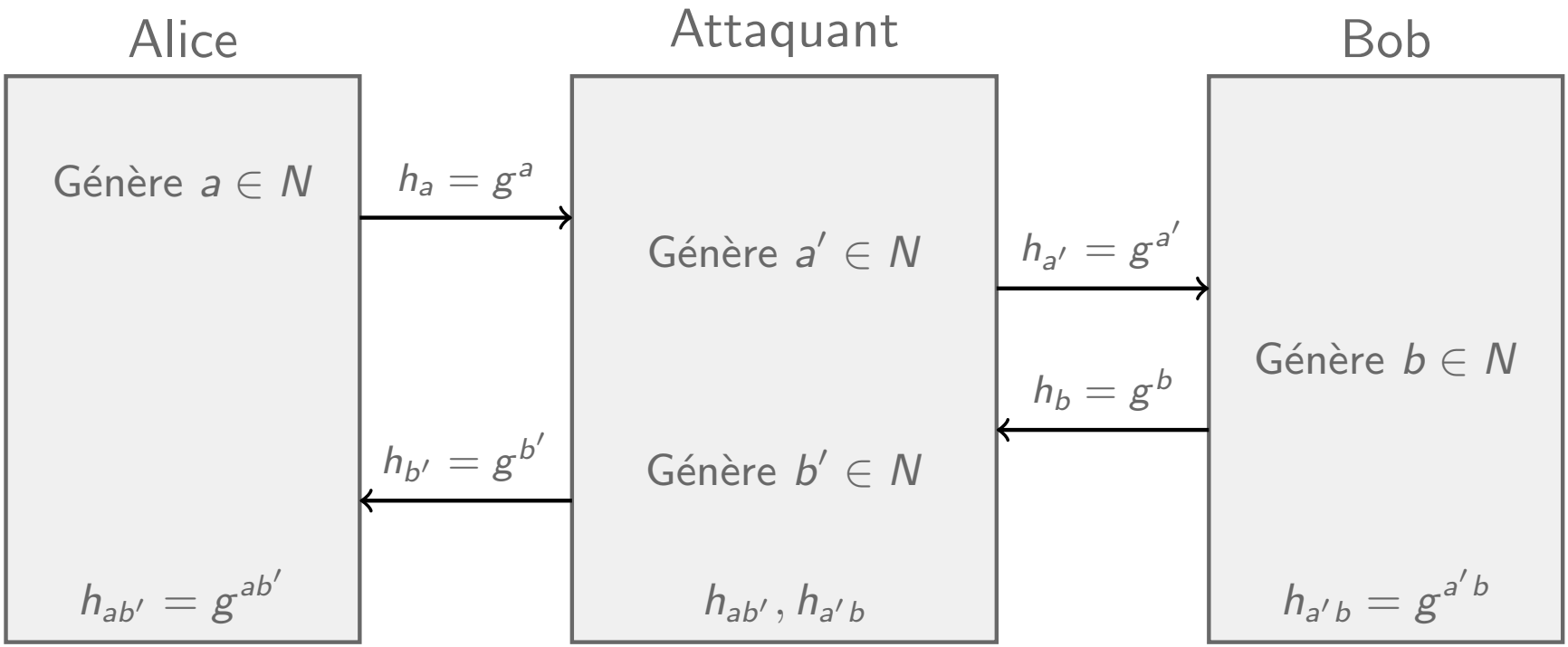
Master 2: 1er semestre
(2014/2015)

Introduction

- Protocole Diffie Hellman:



- Attaque de l'intrus du milieu:



Conclusion

- Pour Alice et Bob le protocole s'est déroulé normalement
- L'attaquant a peut chiffrer et déchiffrer tous les messages qu'Alice et Bob s'échangent

ATTENTION

- Cette attaque fonctionne face à n'importe quel cryptosystème à clé publique

Certification

Problématique

- Assurer l'authenticité des clés publiques

Idée

- Utiliser un protocole d'authentification pour vérifier la validité et l'origine des clés

Certification

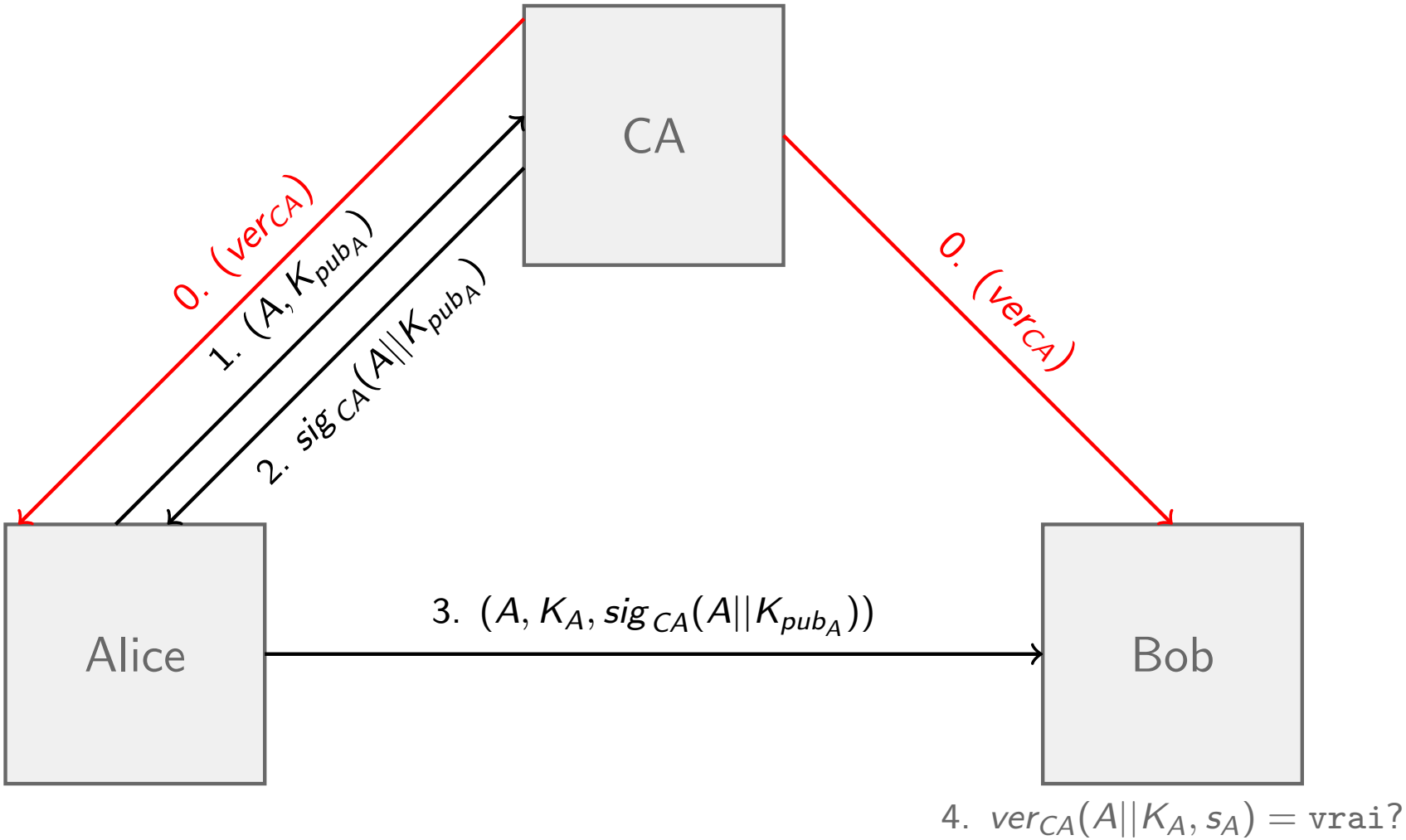
Certificat électronique

- Identité + clé publique
- numéro de version, période de validité etc
- signature de l'ensemble par une authorité de certification

Authorité de certification (CA)

- Tiers de confiance
- Certifit des clés pupliques en les signant à l'aide de sa propre clé privée (on parle souvent de *root key*)

Processus de certification



Processus de certification

Remarques

- Une seule fonction de vérification a besoin d'être distribuée
- Il est plus facile de s'assurer de l'authenticité d'une seule clé
- Les navigateurs web possèdent dès l'installation une liste de meta-certificats assurant l'authenticité des clés de nombreuses authorités

Infrastructure à clé publique (PKI)

Cycle de vie d'un certificat

- enregistrement de l'utilisateur
- génération de clé et distribution
- fabrication du certificat
- révocation ou expiration
- archivage

Infrastructure à clé publique (PKI)

- Un certificat n'est en général pas signé par une autorité de confiance
- Il faut remonter une chaîne de certificats

Modèles de confiance

- Hiérarchie stricte
- Autorités en réseau
- Navigateur web

Protocole "Station-to-Station"

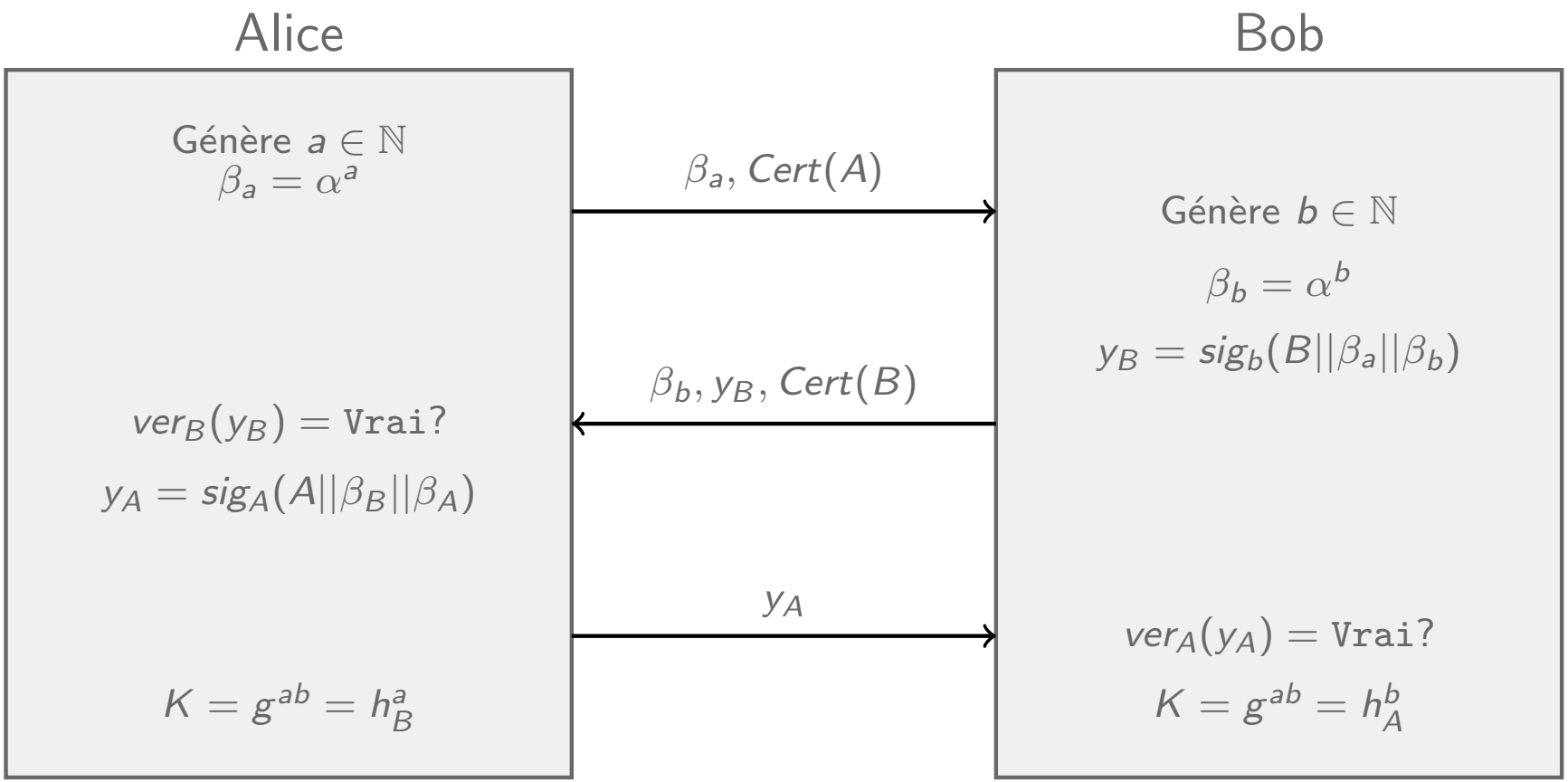
Idée Générale

- Combiner le protocole de Diffie-Hellman avec un schéma d'authentification
- Assurer la validité des fonctions de verification de signature grâce à un certificat

Paramètres

- Un groupe G, $\alpha \in G$ un élément d'ordre n
- Des fonctions de signatures (sig_U, ver_U) pour A et B
- Des certificatats $Cert(U) = (U, ver_U, sig_{CA}(U, ver_U))$

Protocole "Station-to-Station"



Protocole "Station-to-Station"

Attaque MITM

- L'attaquant voit passer α^a et le remplace par $\alpha^{a'}$
- Il reçoit en retour α^b et $sig_B(B||\alpha^{a'}||\alpha^b)$
- Pour poursuivre l'attaque, il doit remplacer α^b par $\alpha^{b'}$ dans le canal et dans la signature
- C 'est impossible sans connaître sig_B

Protocole d'échange de clé MTI

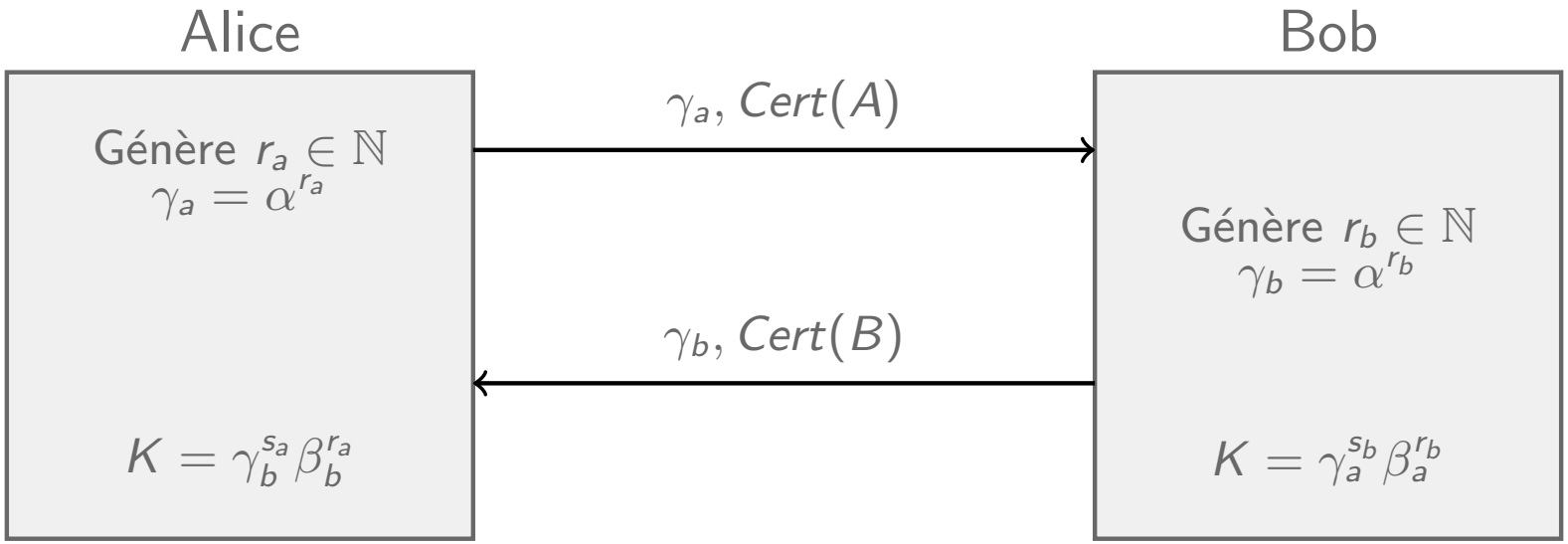
MTI (Matsumoto-Takashima-Imai)

- Version modifiée du protocole de Diffie-Hellman
- Échange de clé en 2 passes et sans signatures

Parmètres

- G un groupe, $\alpha \in G$ un élément d'ordre n
- A (resp. B) possède un exposant privé s_a (resp. s_b) et une donnée publique $\beta_a = \alpha^{s_a}$ (resp. $\beta_b = \alpha^{s_b}$) certifiée par une authorité de certification

Protocole d'échange de clé MTI



Remarques

- On a bien $\gamma_b^{s_a} \beta_b^{r_a} = \alpha^{r_a s_b + r_b s_a} = \gamma_a^{s_b} \beta_a^{r_b}$
- On peut vérifier que le protocole est sûr face à un attaquant passif ou face à l'attaque MITM

Attaque par clé de session connue

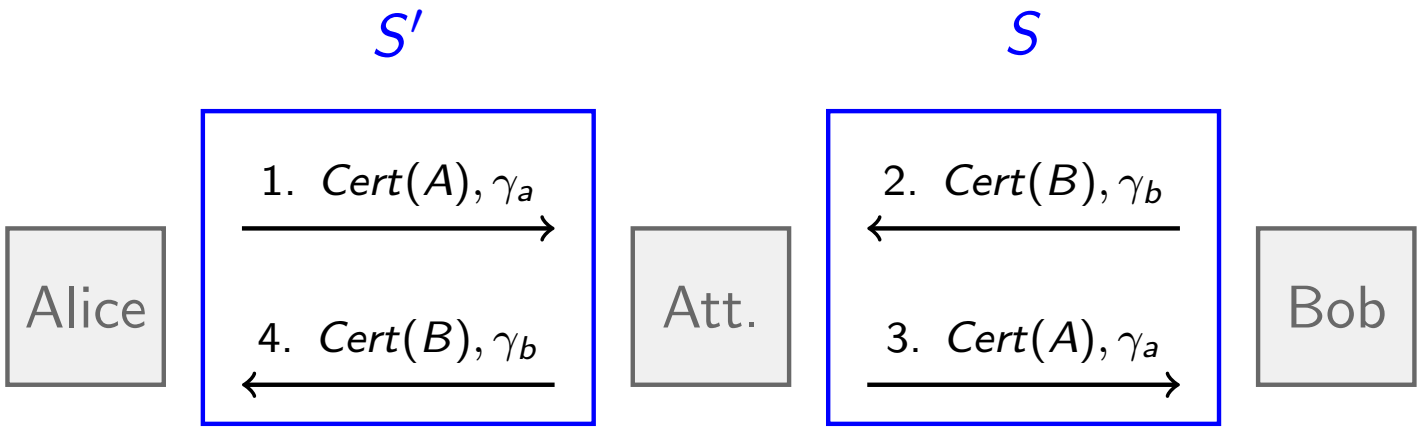
Modèle d'attaque

- L'attaquant peut demander la clé de session K_i d'un certain nombre de session S_i
- Doit être capable de calculer la clé K pour une autre session S dans laquelle il est actif

Attaque contre MTI

- Lancer deux sessions parallèles
- Obtenir la clé pour l'une et s'en servir pour calculer la clé pour l'autre

Attaque par clé de session connue



Déroulement de l'attaque

- L'Attaquant ouvre deux sessions en parallèle S et S'
- Réclame la clé de la session S'
- En déduit la clé de la session S (qui est identique)

Attaque par clé de session connue

Analyse

- L'attaque fonctionne grâce à la symétrie du calcul de K:
  - A calcule $\gamma_b^{s_a}\beta_b^{r_a}$
  - B calcule $\gamma_a^{s_b}\beta_a^{r_b}$

  Ainsi $K((s_a, r_a), s_b, r_b)) = K((s_b, r_b), (s_a, r_a))$

Contre mesure

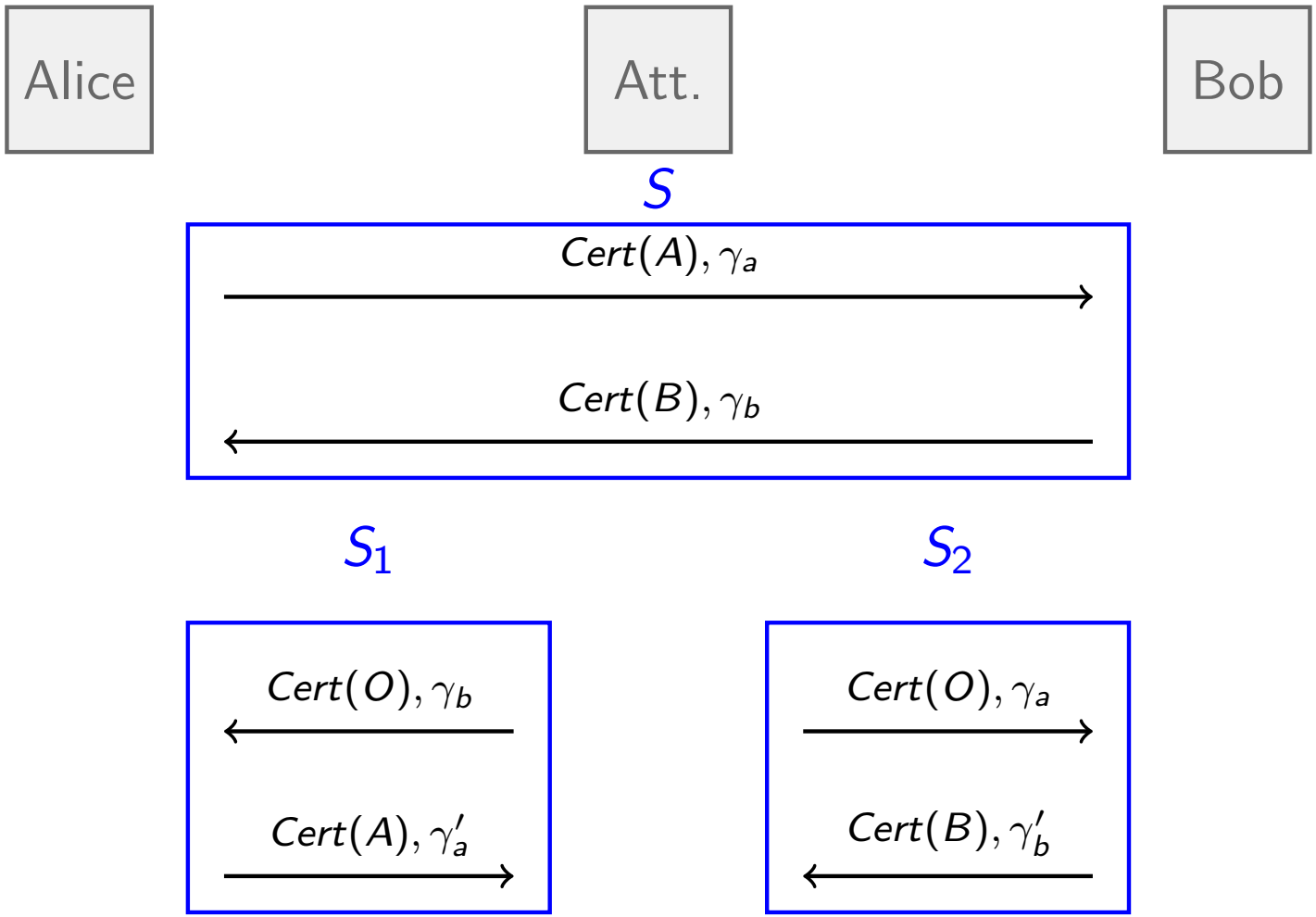
- Une contre mesure simple consiste à utiliser une fonction de hachage et ordonner le calcul de K:
  - $K = h(\alpha^{r_a s_b}||\alpha^{r_b s_a})$ où A est initiateur et B le receveur
  - Les clés des session S et S' sont alors différentes

Attaque par clé de session connue

Attaque triangulaire de Burnester

- L'attaquant observe une première session S entre Alice et Bob
- Il initie une session S_1 avec Alice en utilisant le paramètre de Bob de la session S
- Il initie une session S_2 avec Bob en utilisant le paramètre d'Alice de la session S
- Il réclame les clés de ces deux sessions

Attaque par clé de session connue



Attaque triangulaire de Burnester

- Les clés de session sont:

$$\begin{aligned} K &= \alpha^{s_A r_B + r_A s_B} \\ K_1 &= \alpha^{s_0 r'_A + r_O s_A} \\ K_2 &= \alpha^{s_0 r'_B + r_O s_B} \end{aligned}$$

- Après obtention de K_1 et K_2 on calcule:

$$K = \frac{K_1 K_2}{(\gamma'_a \gamma'_b)^{s_O}}.$$