Signatures et Infrastructures à Clé Publique (PKI)

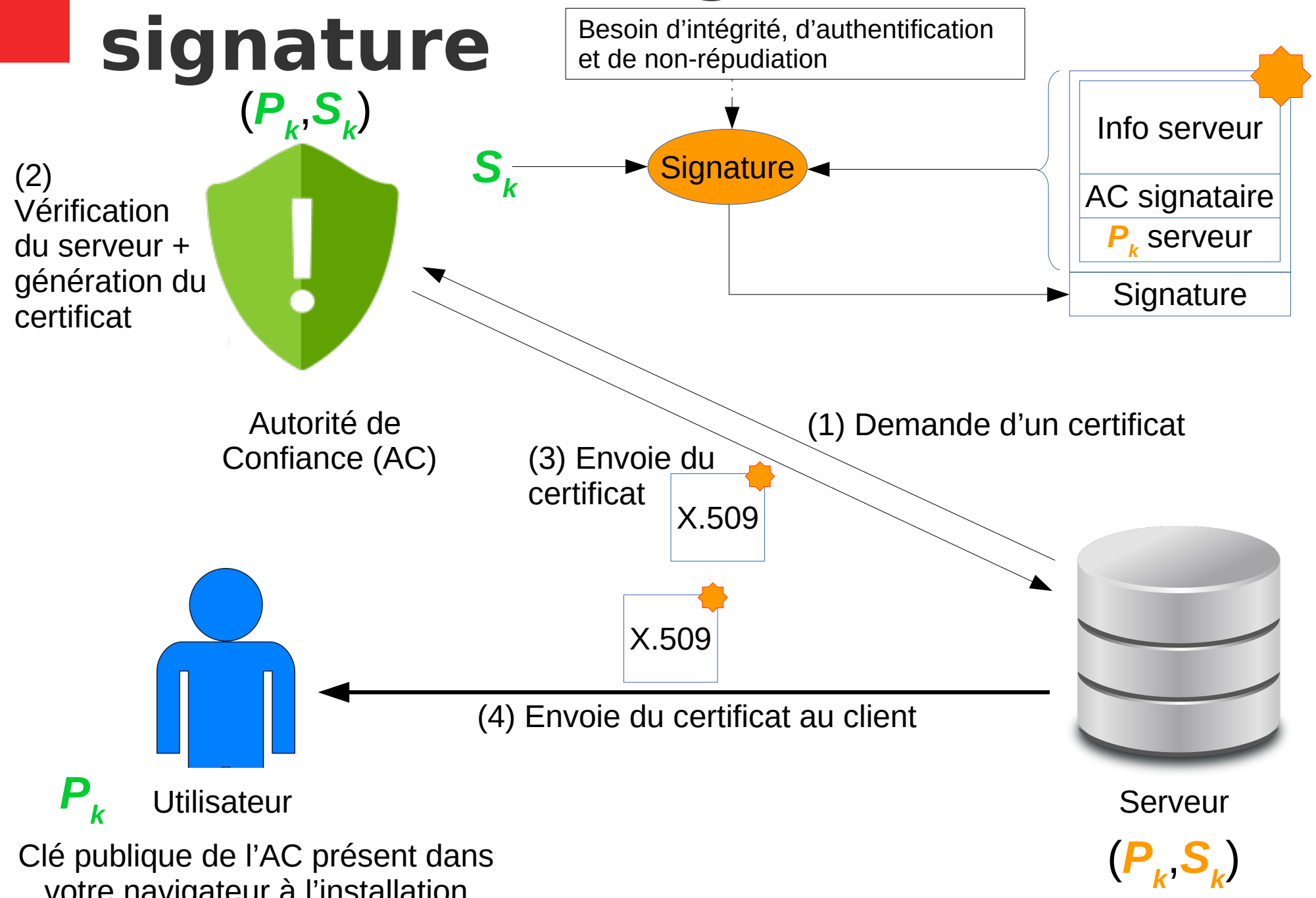
TLS-SEC
Module d’entrée Cryptographie

Vincent MIGLIORE

Agenda

- Étude des besoins cryptographiques dans une architecture à clé publique.
- Principe de fonctionnement d’une signature.
- Les standards d’infrastructure à clé publique.

Intérêt d'un algorithme de signature



Les besoins

Cible en terme de sécurité	Hash	MAC
Intégrité message non altéré		
Authentification		
Non-répudiation ne peux pas dire qu'il n'est pas l'auteur du message		
Chiffrement	Aucun	Symétrique

Les besoins

Cible en terme de sécurité	Hash	MAC
Intégrité message non altéré		
Authentification		
Non-répudiation ne peux pas dire qu'il n'est pas l'auteur du message		
Chiffrement	Aucun	Symétrique

Les deux mécanismes sont basés sur des fonctions de hachage à sens unique.
Ainsi, un attaquant ne peux pas altérer le message sans avoir à calculer un nouveau haché.

Les besoins

Cible en terme de sécurité	Hash	MAC
Intégrité message non altéré		
Authentification		
Non-répudiation ne peux pas dire qu'il n'est pas l'auteur du message		
Chiffrement	Aucun	Symétrique

La méthode basé sur le calcul du haché du message uniquement ne garanti pas l'authentification, car aucun secret n'est associé à la personne qui a généré le haché.

Les besoins

Cible en terme de sécurité	Hash	MAC
Intégrité message non altéré		
Authentification		
Non-répudiation ne peux pas dire qu'il n'est pas l'auteur du message		
Chiffrement	Aucun	Symétrique

Malheureusement, le MAC ne permet pas de définir la paternité d'un message, car comme Ce mécanisme est basé sur un chiffrement symétrique, quiconque est capable de vérifier un message est capable de générer un MAC pour un autre message.

Les besoins

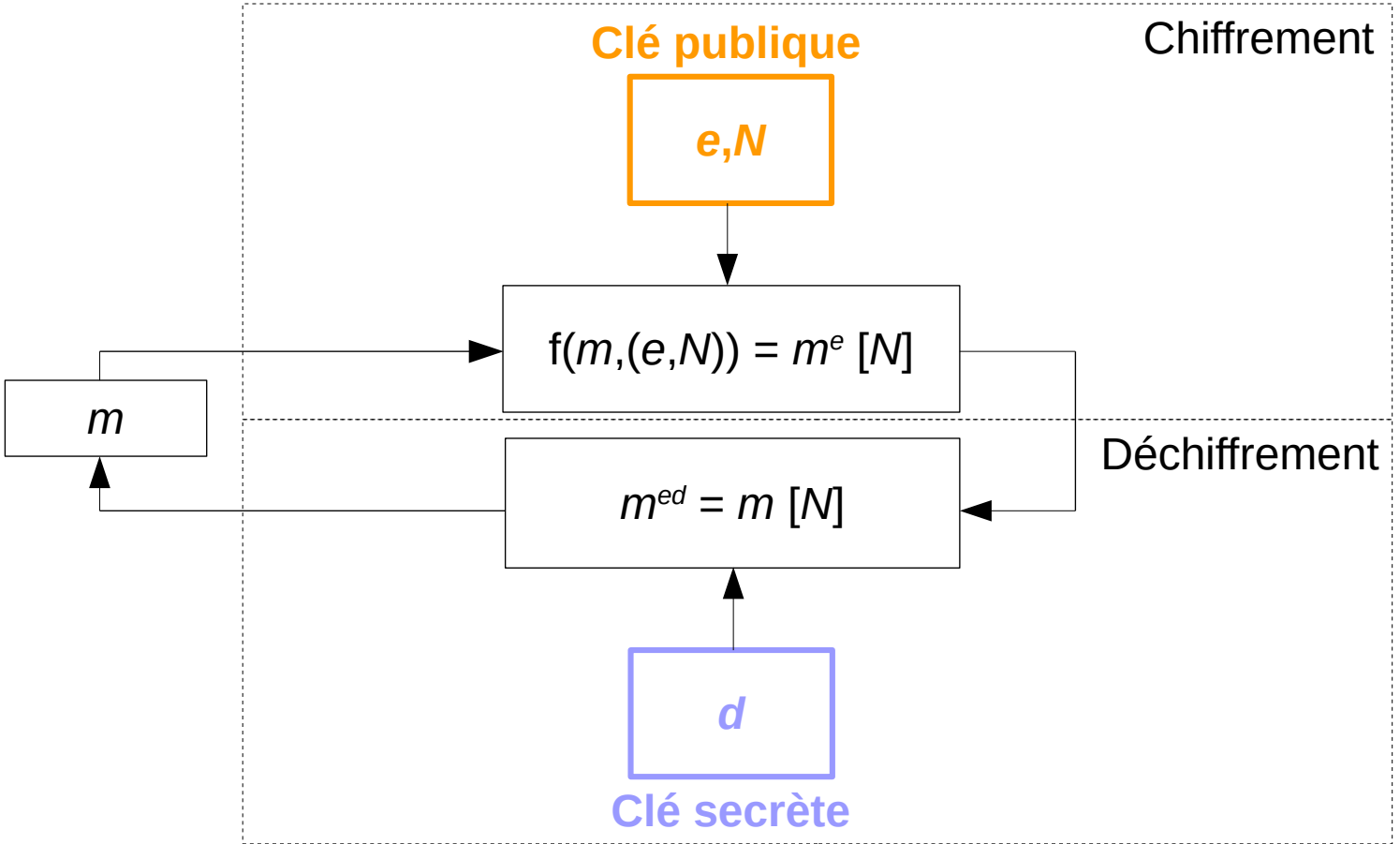
Cible en terme de sécurité	Hash	MAC	Signature
Intégrité message non altéré			
Authentification			
Non-répudiation ne peux pas dire qu'il n'est pas l'auteur du message			
Chiffrement	Aucun	Symétrique	Asymétrique

On arrive donc à l'avènement des mécanismes de signature, qui utilisent le chiffrement asymétrique pour coupler le signeur du vérifieur.

Créer un mécanisme de signature, en quelques mots

Signature à partir de *Textbook* RSA

Cas chiffrement publique, déchiffrement secret

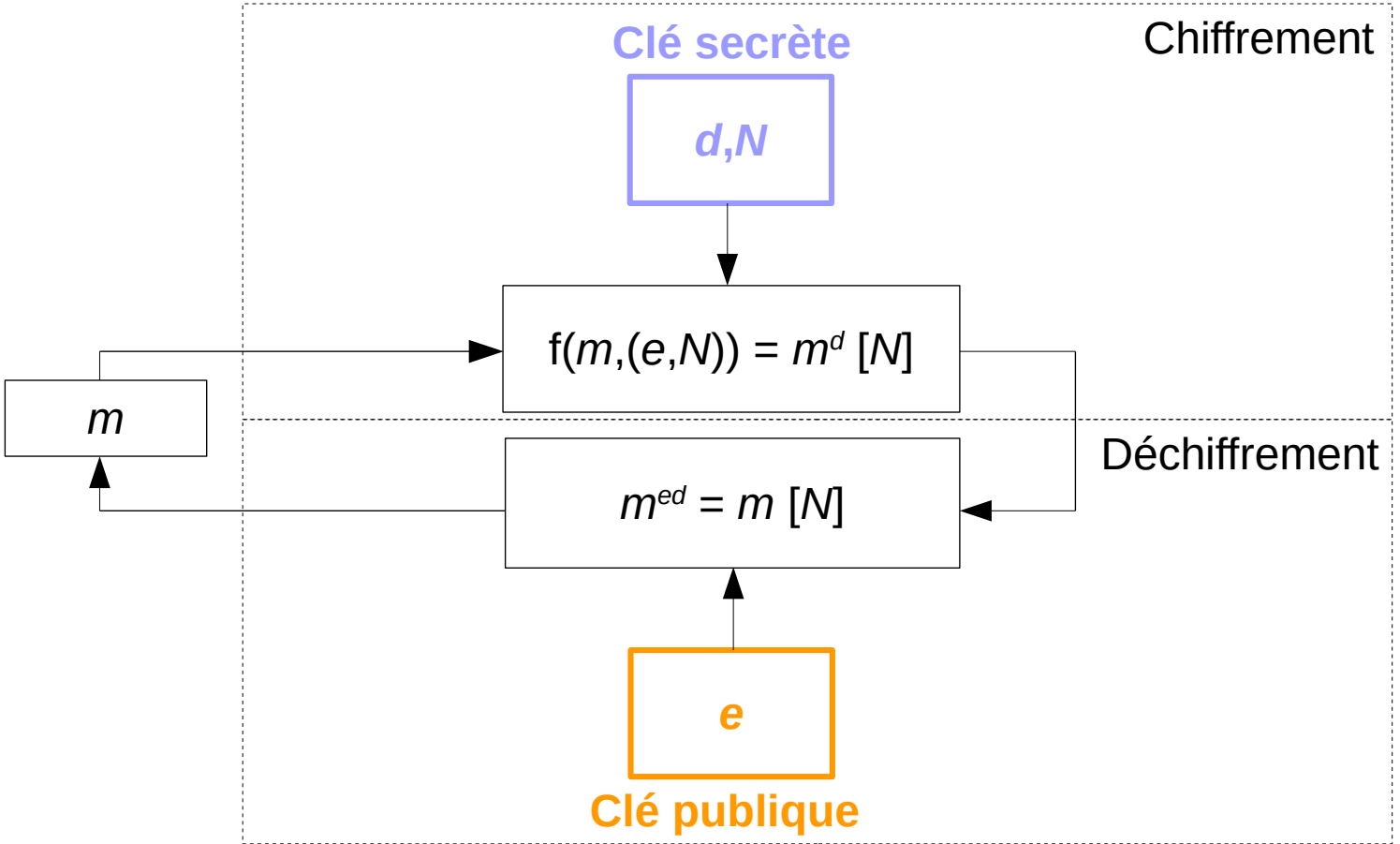


Mode de fonctionnement de RSA en mode échange de clé.

Créer un mécanisme de signature, en quelques mots

Signature à partir de *Textbook* RSA

Cas **chiffrement secret**, **déchiffrement publique**

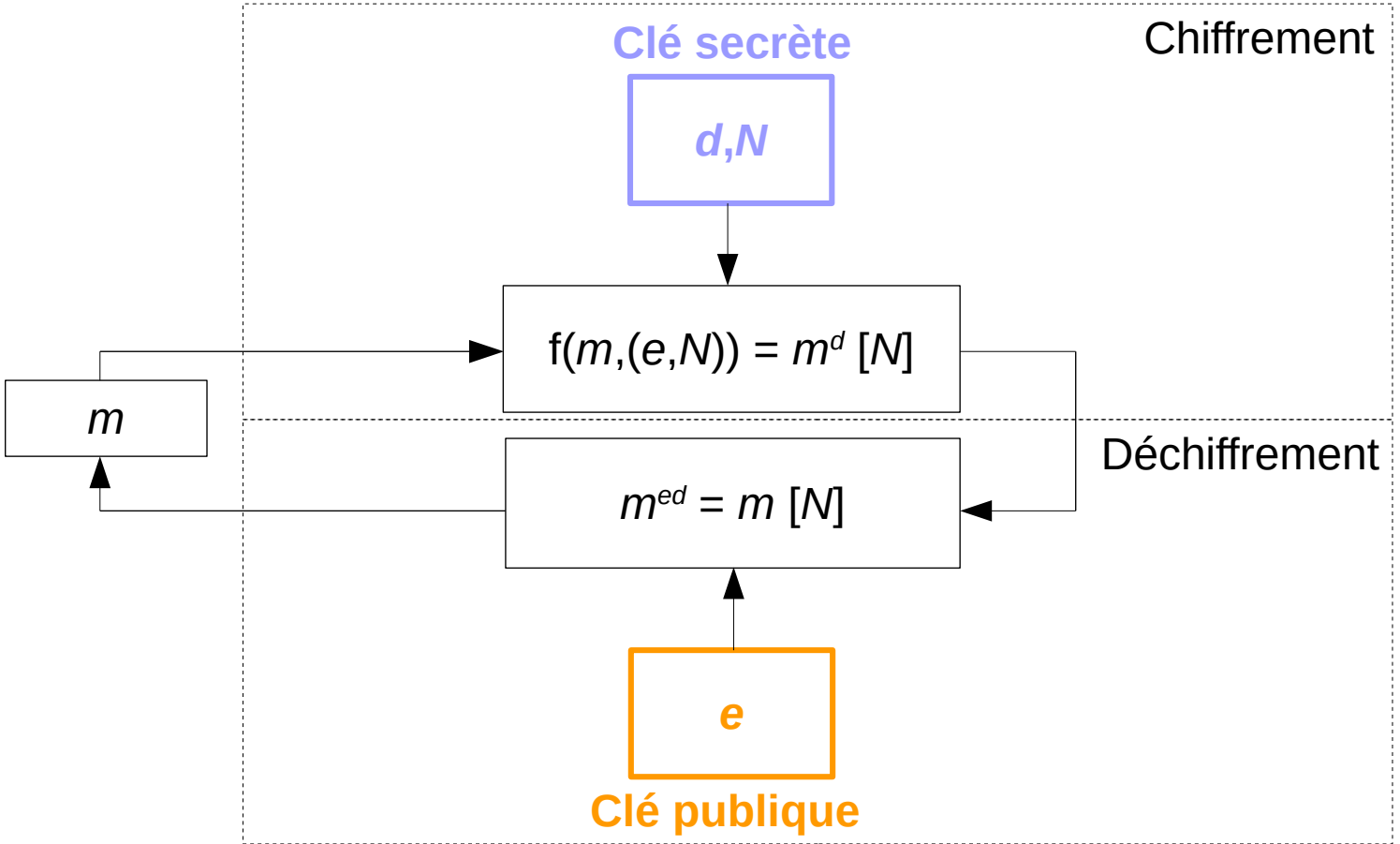


Mode de fonctionnement de RSA en mode « signature ».

Créer un mécanisme de signature, en quelques mots

Signature à partir de *Textbook* RSA

Cas **chiffrement secret**, **déchiffrement publique**

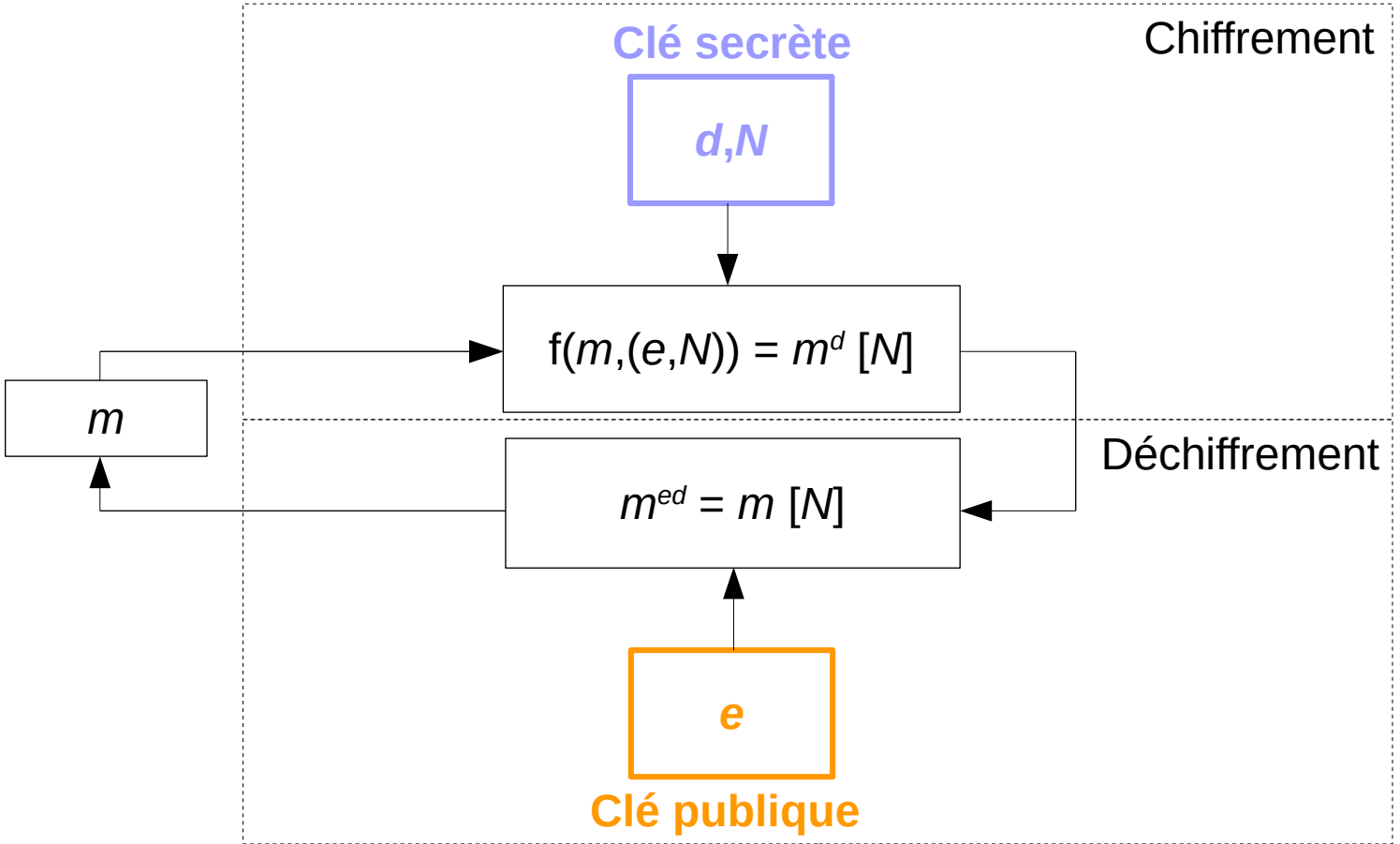


A votre avis, si l'on prends les informations publiques du serveur + la clé publique et que l'on signe le tout en utilisant cette méthode, le résultat est-il sûr ?

Créer un mécanisme de signature, en quelques mots

Signature à partir de *Textbook* RSA

Cas chiffrement secret, déchiffrement publique



A votre avis, si l'on prends les informations publiques du serveur + la clé publique et que l'on signe le tout en utilisant cette méthode, le résultat est-il sûr ?

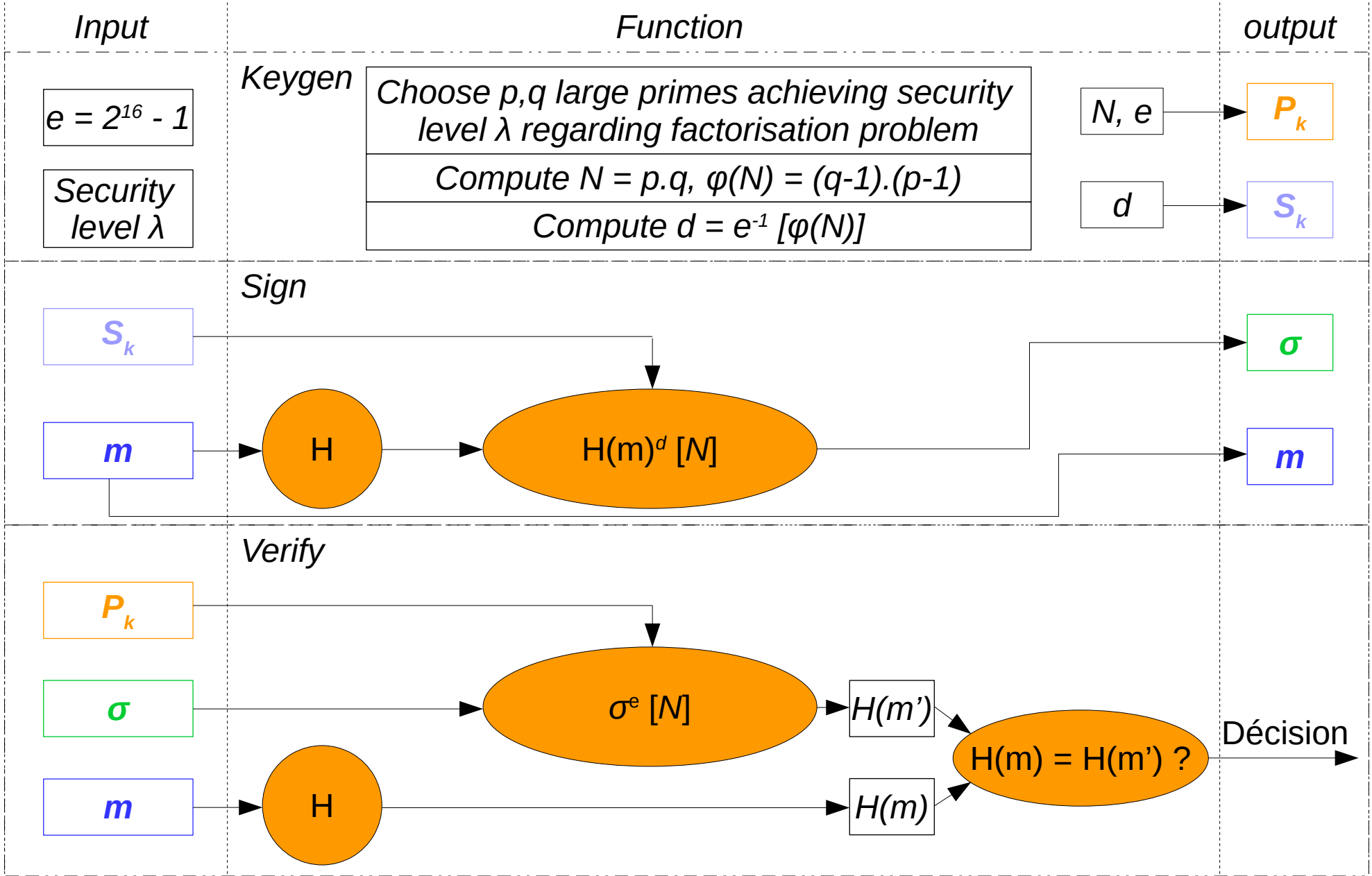
Non :

le chiffrement étant déterministe, l'attaquant peux construire à partir d'un ensemble de chiffrés une signature valide.

De plus, *Textbook* RSA est homomorphe, c'est-à-dire pour m_1, m_2 deux messages et c_1, c_2 leurs chiffrés :

$c_1 . c_2 = (m_1 . m_2)^d [N]$

RSA-FDH


Input
Function
output
Keygen
$e = 2^{16} - 1$
Security level λ
Choose p,q large primes achieving security level λ regarding factorisation problem
Compute N = p.q, φ(N) = (q-1).(p-1)
Compute $d = e^{-1}$ [φ(N)]
N, e
d
P_k
S_k
Sign
S_k
m
H
$H(m)^d$ [N]
σ
m
Verify
P_k
σ
m
H
σ^e [N]
H(m')
H(m)
H(m) = H(m') ?
Décision

RSA-FDH *in practice*

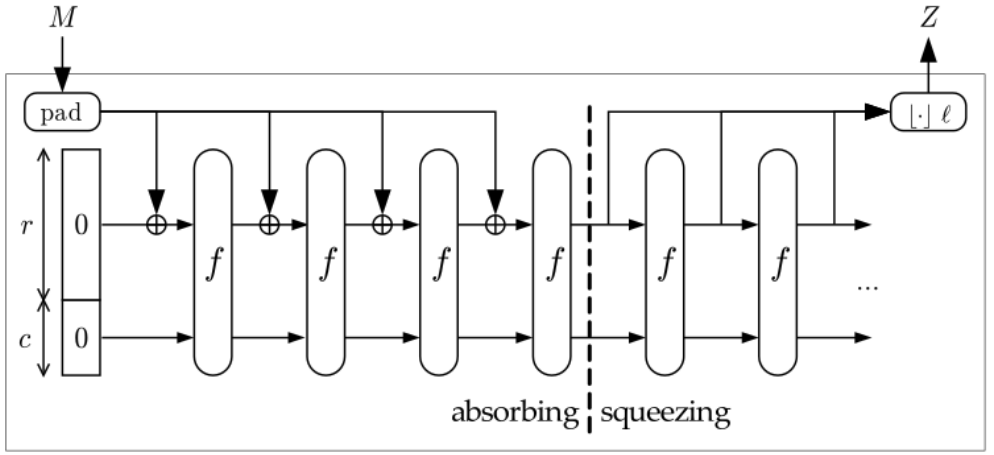
Points essentiels de RSA-FDH

- RSA-FDH a une **sécurité prouvée**.
- RSA-FDH permet d'**assurer une inforgeabilité existentielle** : Il n'existe pas de message où l'attaquant puisse forger une signature valide sans le secret.
- Cette construction demande que la fonction de hachage produise **un condensat de la taille du module *N*** de RSA.(On peut utiliser SHA3 par exemple)

Keccak

Keccak = SHA3

- Construction en « éponge »
  - Absorbing : On absorbe le message de taille arbitraire dans notre « éponge ».
  - Squeezing : On essors notre éponge pour produire un condensat de taille arbitraire.



sponge

The sponge construction

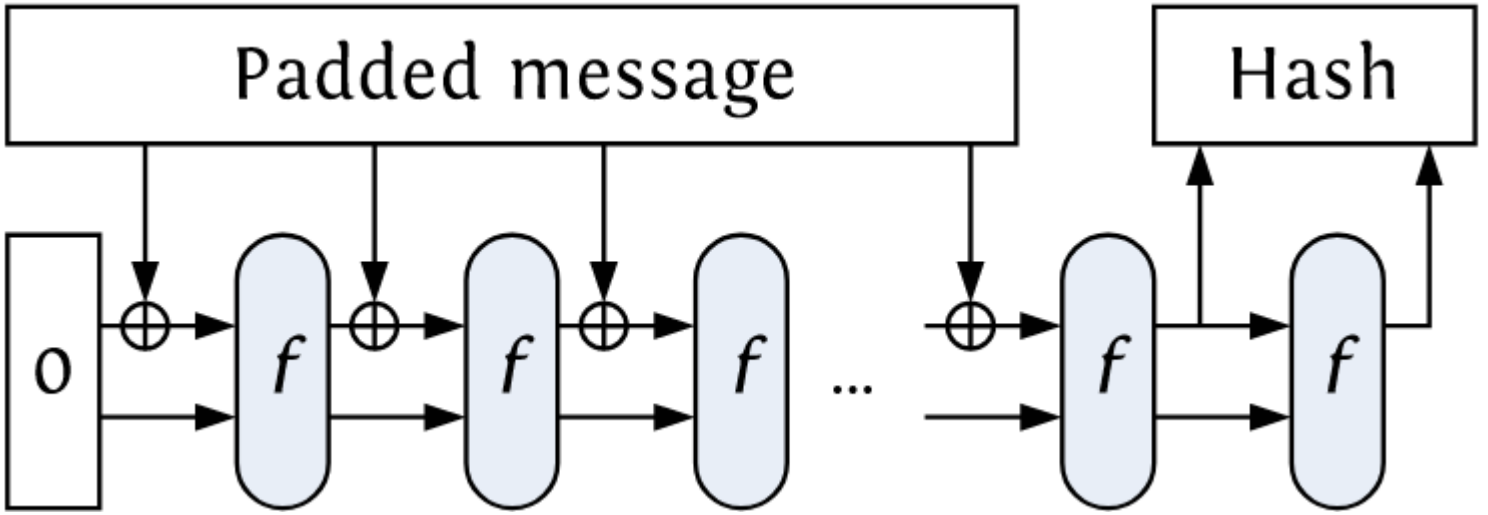
f : Permutation
c : Partie de l'état interne secret
r : Partie de l'état interné consacré à l'absorption/essorage

Images : site officiel Keccak

Keccak

Keccak = SHA3

- Utilisation en fonction de hachage

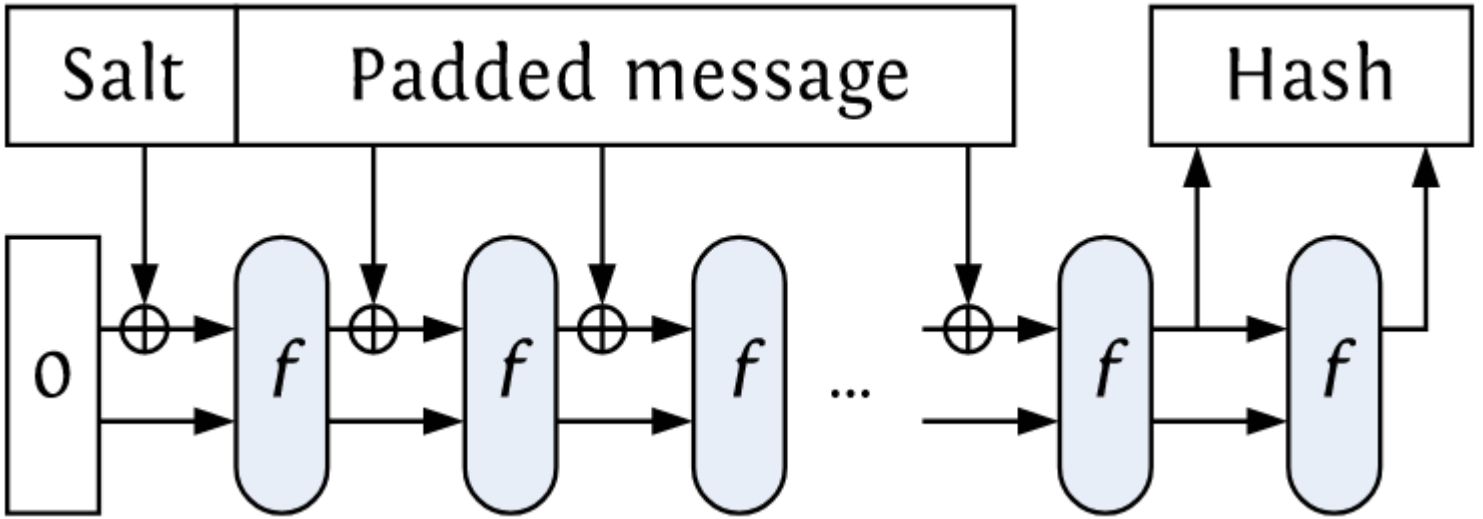


Images : http://passwords12.at.ifi.uio.no/Joan_Daemen_Passwords12.pdf

Keccak

Keccak = SHA3

- Utilisation en fonction de hachage avec sel

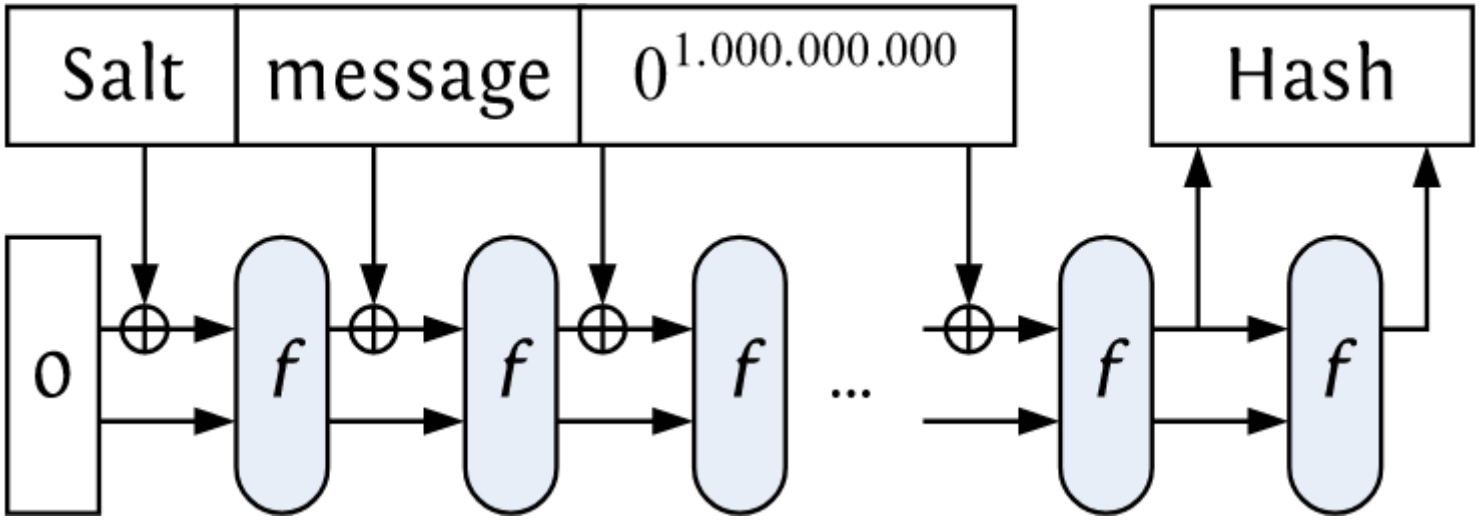


Images : http://passwords12.at.ifi.uio.no/Joan_Daemen_Passwords12.pdf

Keccak

Keccak = SHA3

- Utilisation en fonction de hachage avec sel et pénalisation

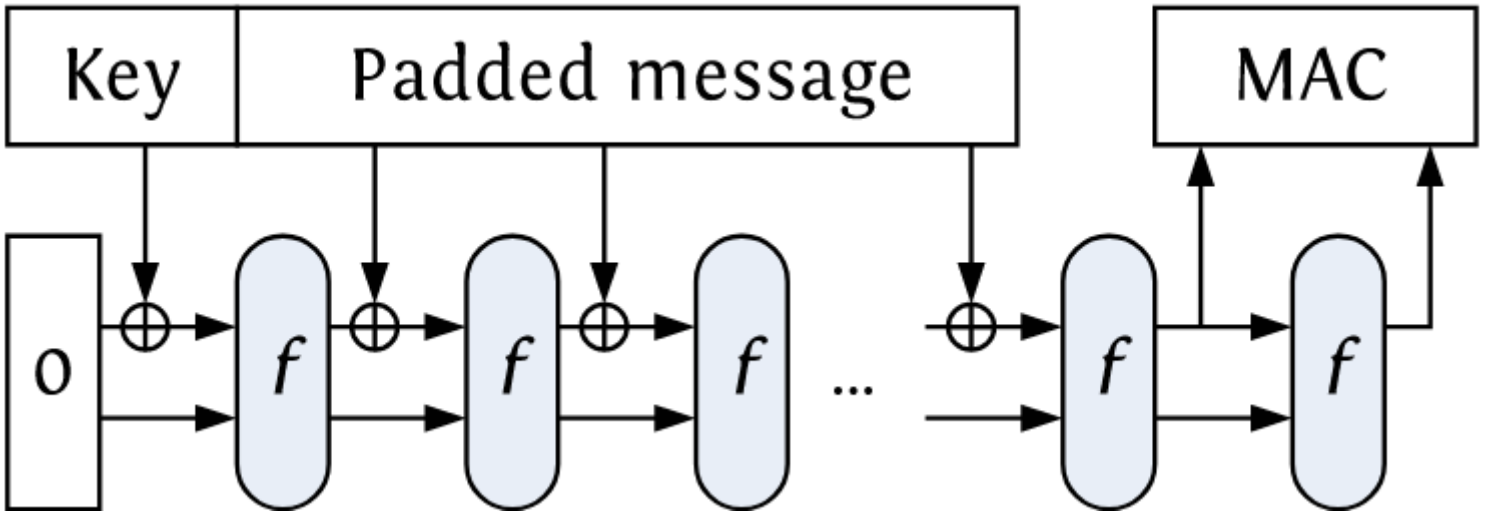


Images : http://passwords12.at.ifi.uio.no/Joan_Daemen_Passwords12.pdf

Keccak

Keccak = SHA3

- Utilisation en code d'authentification de message

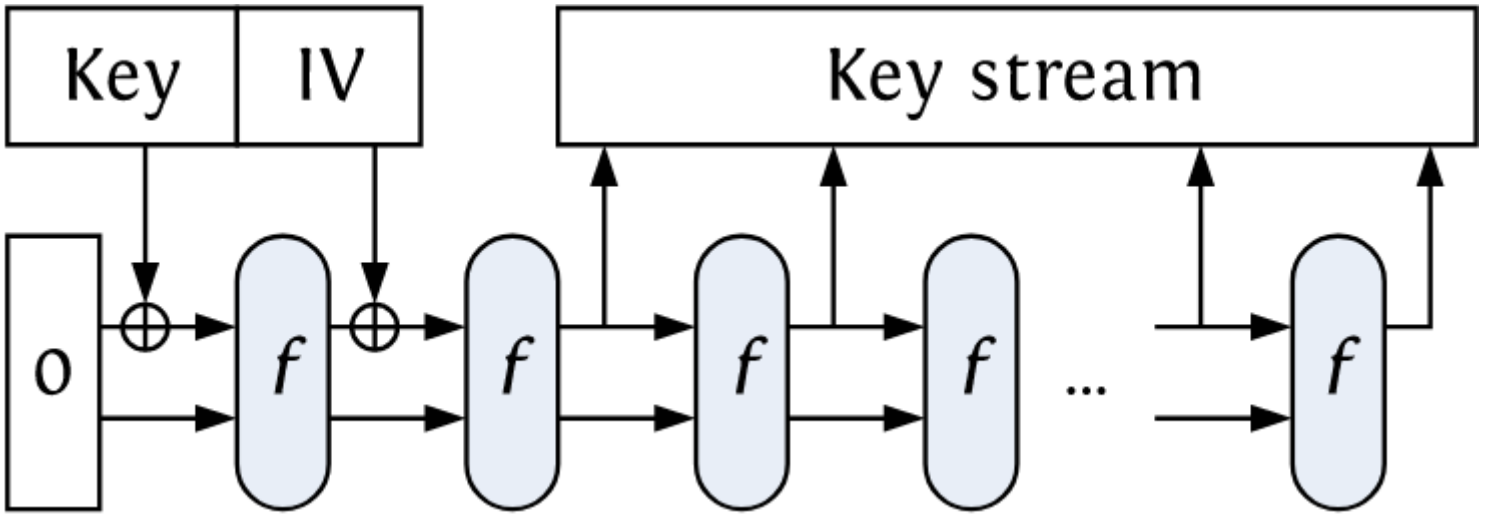


Images : http://passwords12.at.ifi.uio.no/Joan_Daemen_Passwords12.pdf

Keccak

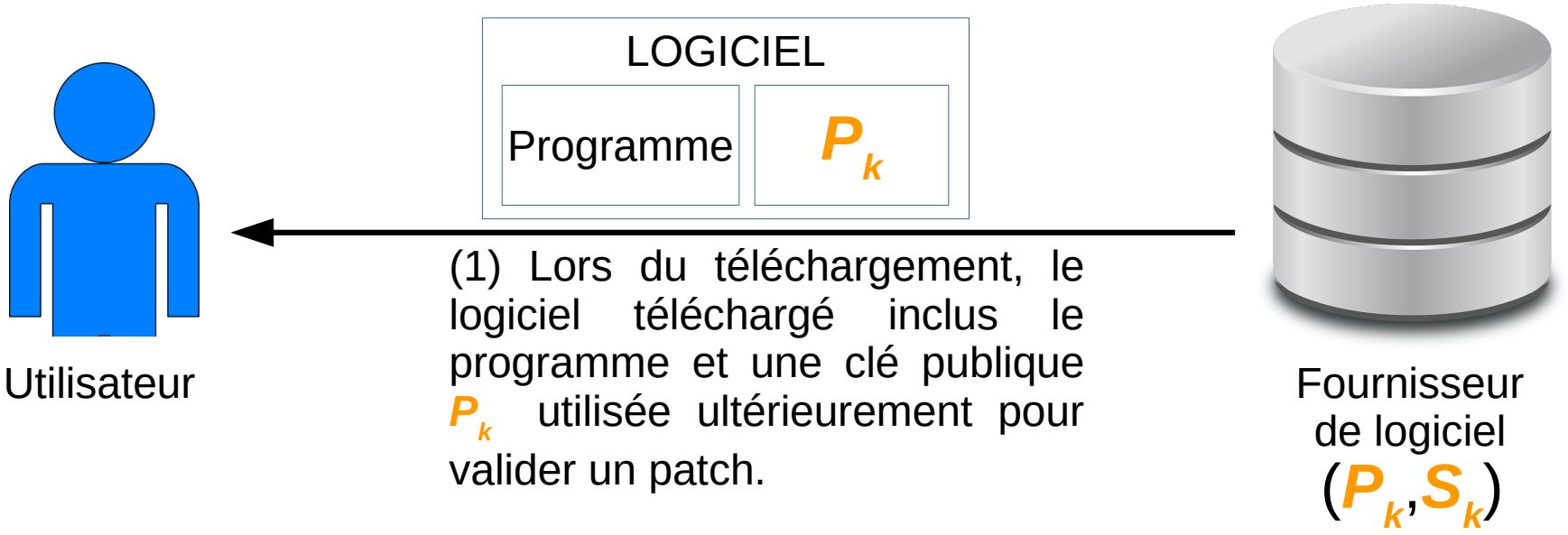
Keccak = SHA3

- Utilisation en chiffrement par flot

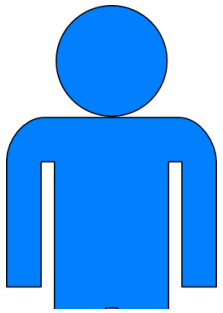


Images : http://passwords12.at.ifi.uio.no/Joan_Daemen_Passwords12.pdf

Déploiement d'un patch logiciel

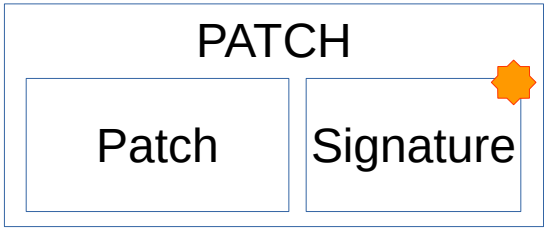


Déploiement d'un patch logiciel



P_k Utilisateur

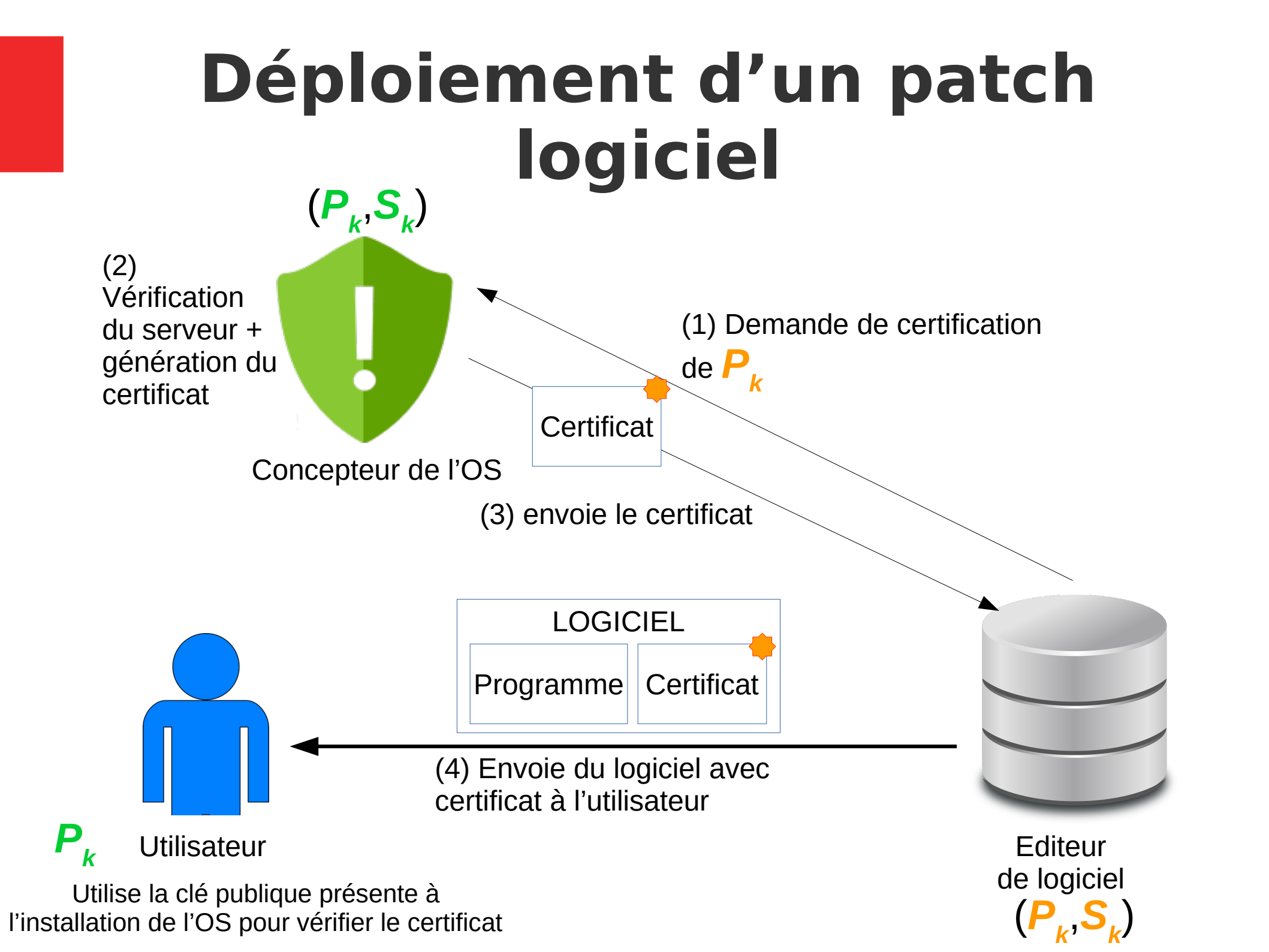
Vérification du patch avec La clé publique incluse dans le logiciel



(2) Lors du déploiement d'un patch, celui-ci est signé avec S_k du serveur pour permettre au client de vérifier l'authenticité du patch

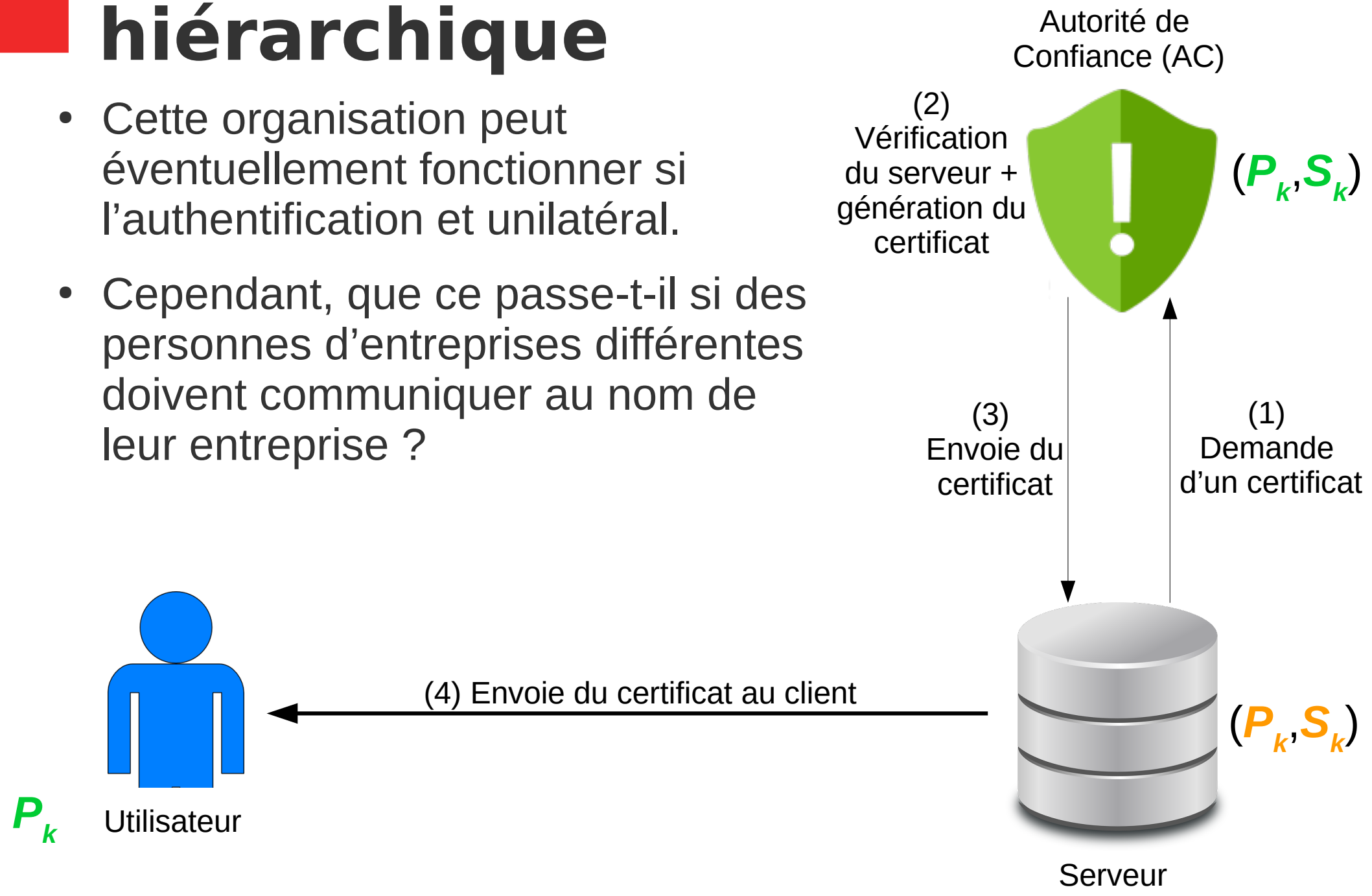
Fournisseur de logiciel (P_k,S_k)

Déploiement d'un patch logiciel



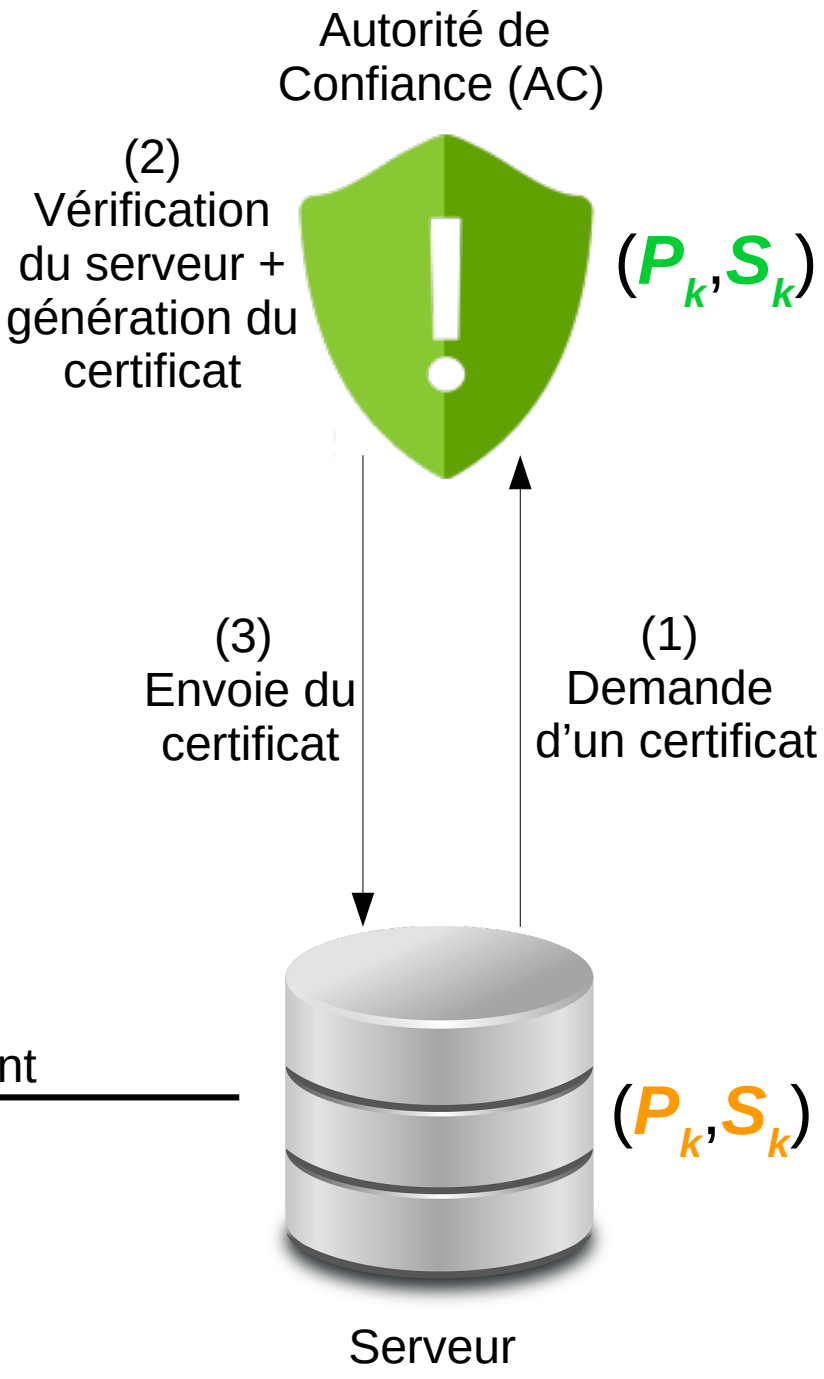
Le besoin d'un modèle hiérarchique

- Cette organisation peut éventuellement fonctionner si l'authentification et unilatéral.
- Cependant, que ce passe-t-il si des personnes d'entreprises différentes doivent communiquer au nom de leur entreprise ?

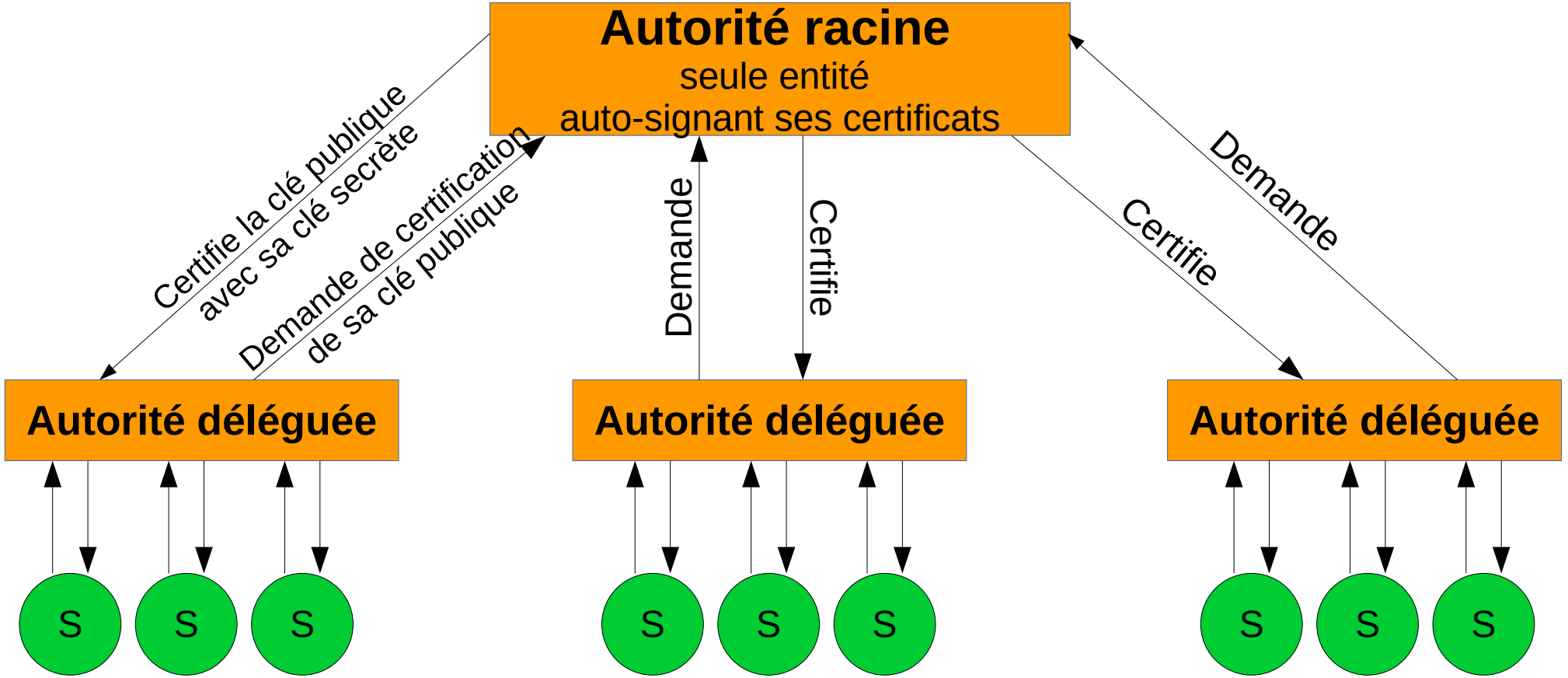


Le besoin d'un modèle hiérarchique

- Cette organisation peut éventuellement fonctionner si l'authentification et unilatéral.
- Cependant, que ce passe-t-il si des personnes d'entreprises différentes doivent communiquer au nom de leur entreprise ?
- Il faut une certification hiérarchique

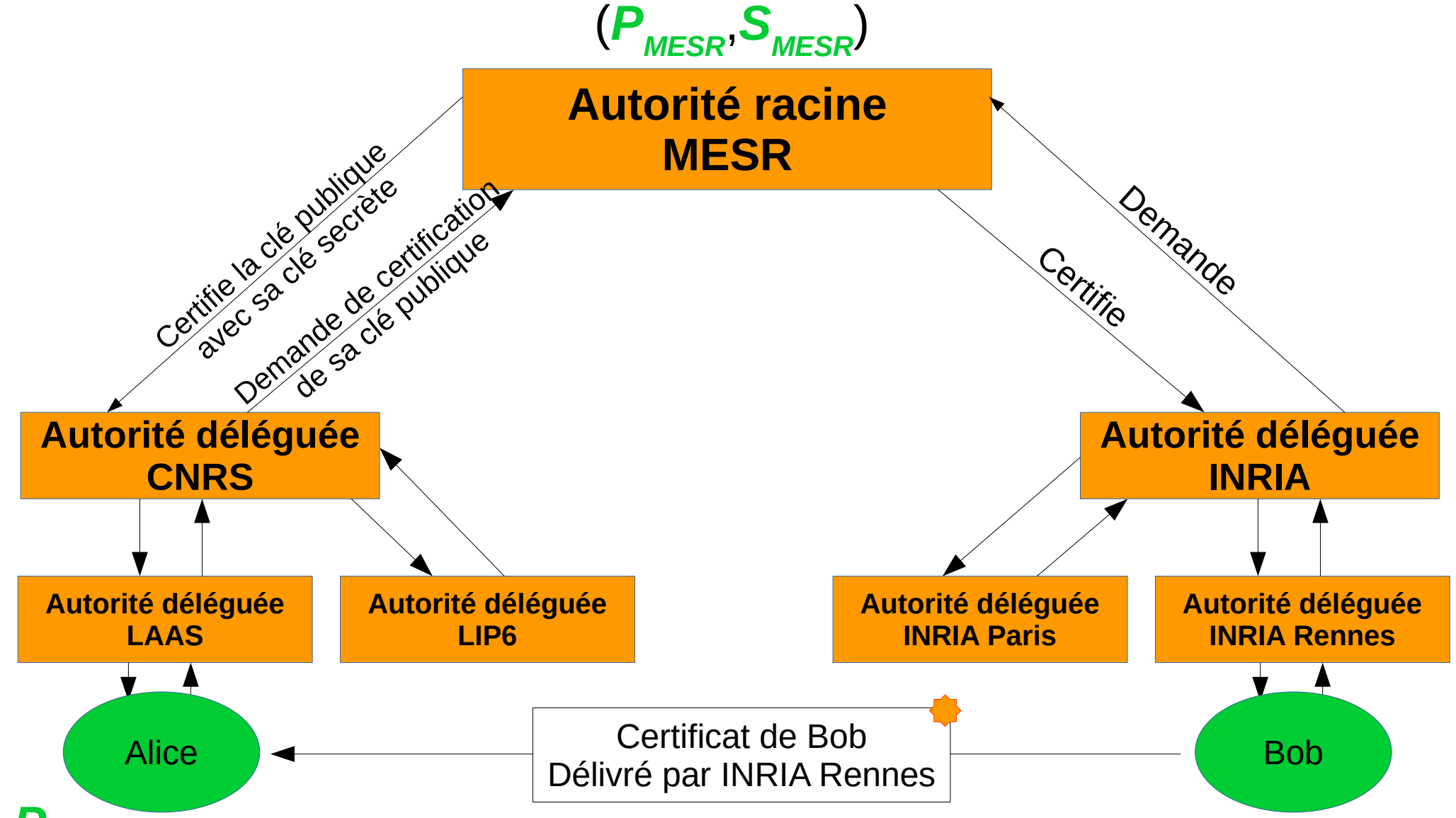


Modèle hiérarchique de la certification



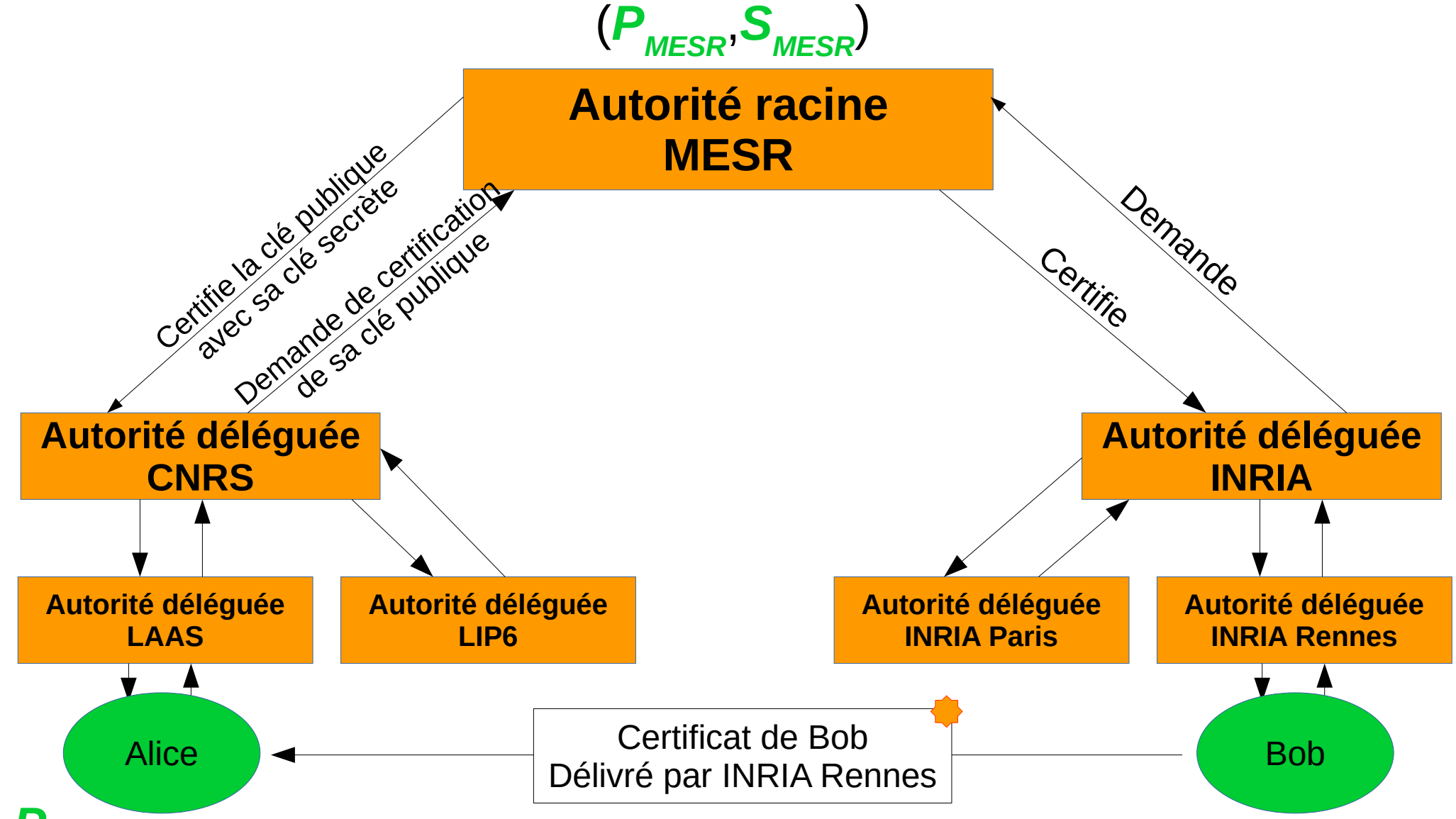
Le modèle hiérarchique permet de s'assurer que les authorités de certifications déléguées agissent au nom de l'autorité racine

Le modèle hiérarchique : Cas de la fonction publique



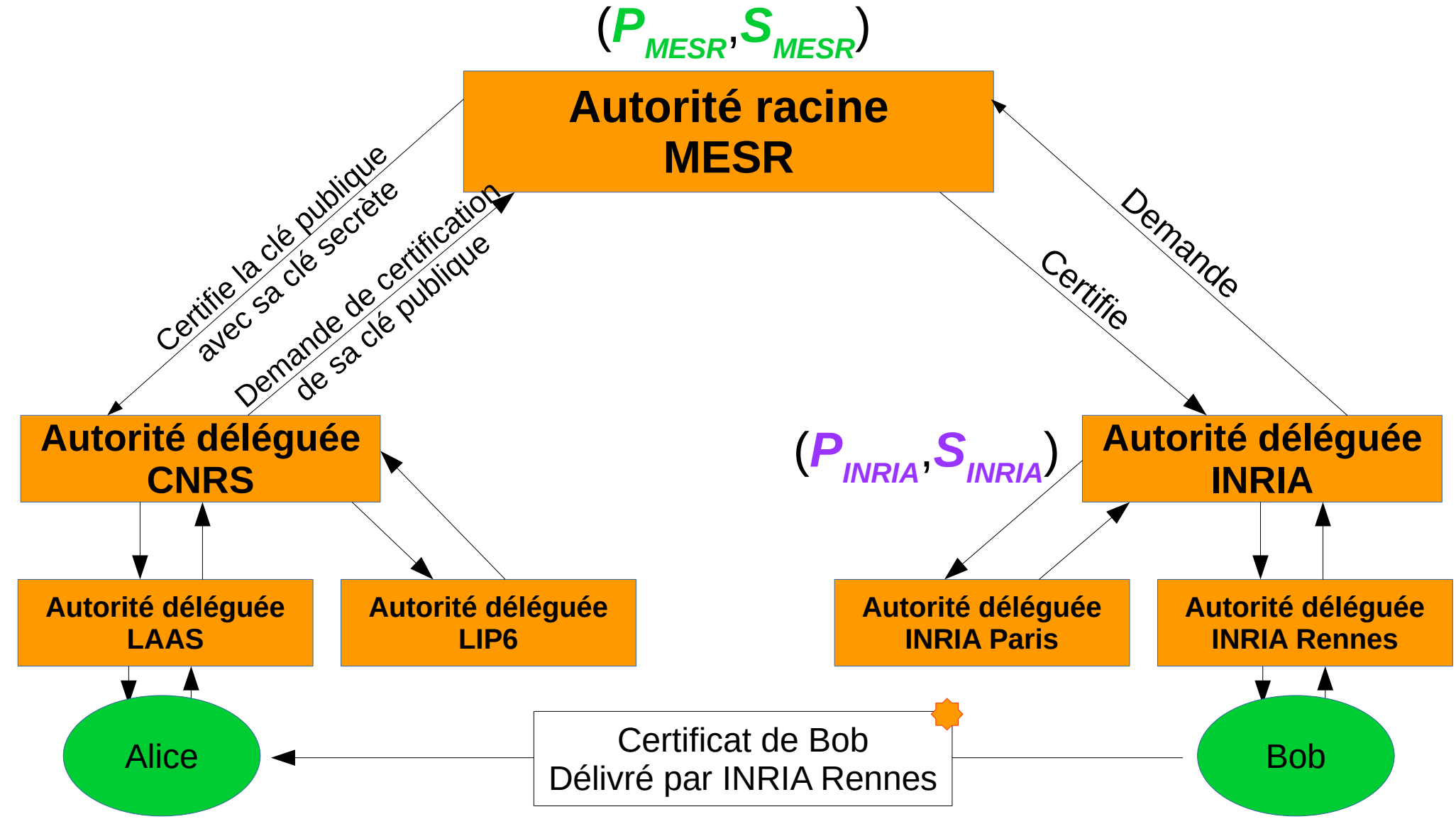
P_{MESR} Comment Alice peux vérifier le certificat de Bob à partir de P_{MESR} ?

Le modèle hiérarchique : Cas de la fonction publique



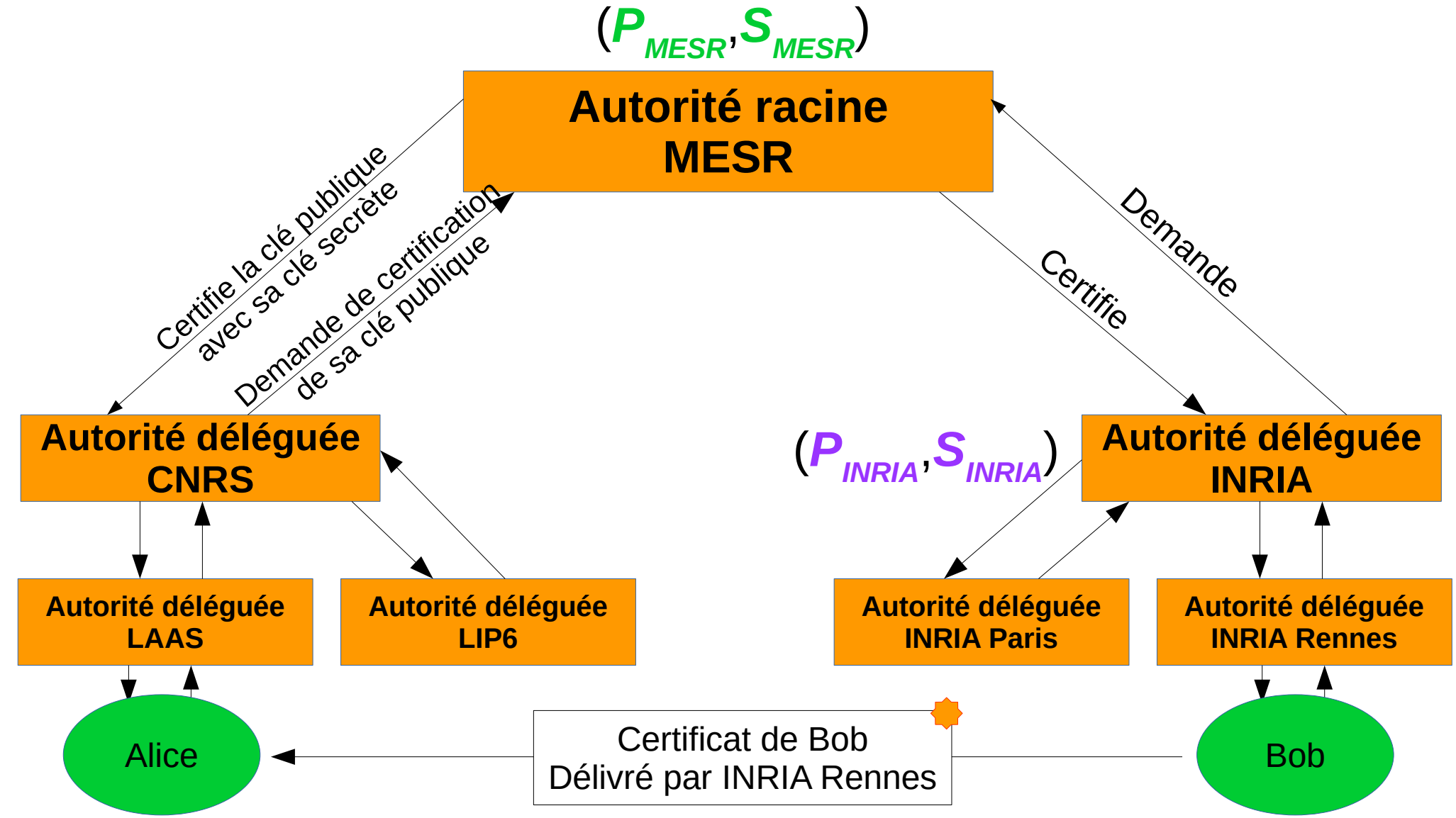
P_{MESR} Alice connaît la chaîne de confiance à vérifier : MESR → INRIA → INRIA Rennes

Le modèle hiérarchique : Cas de la fonction publique



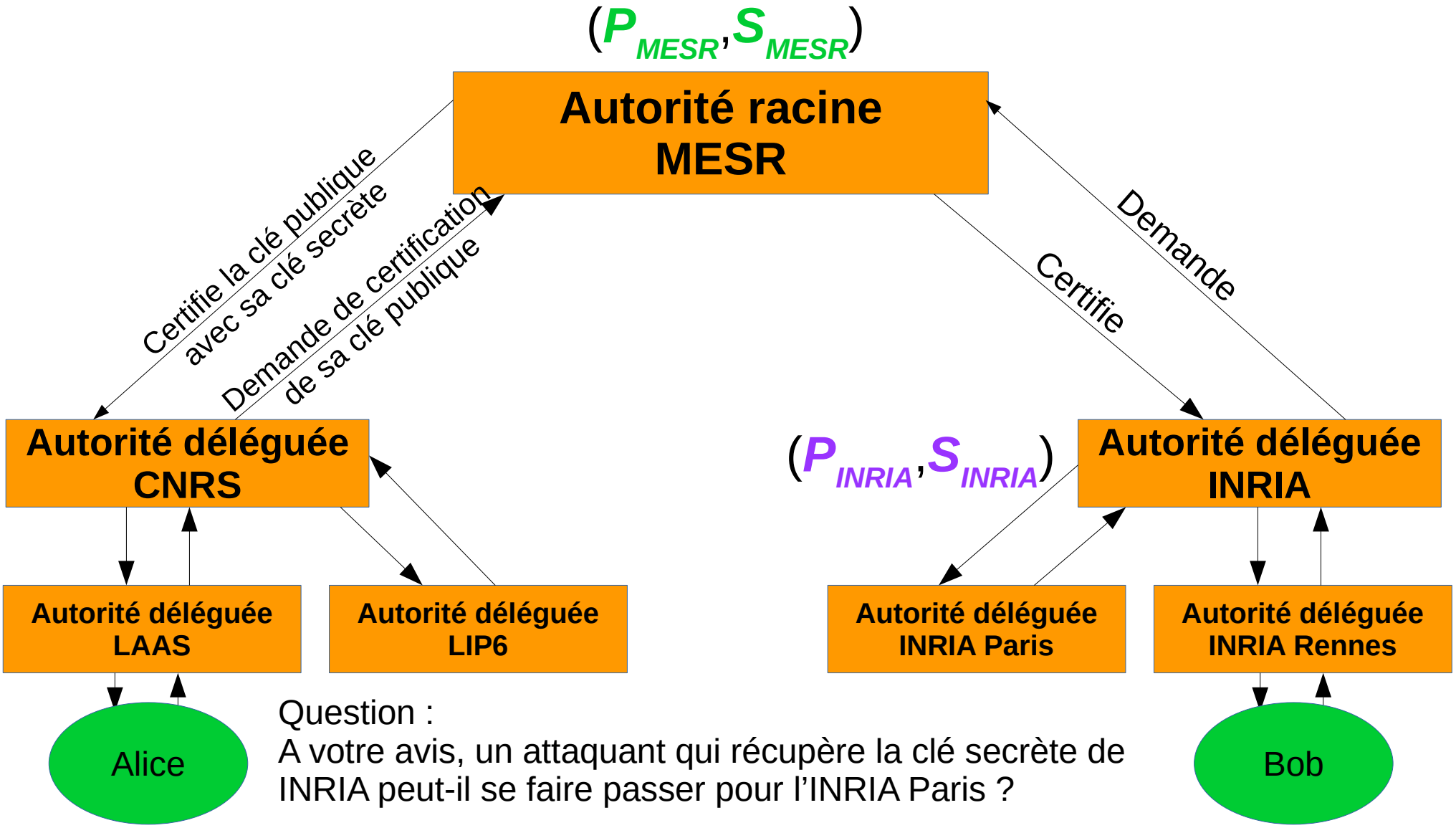
Alice va récupérer le certificat de INRIA, qui contient P_{INRIA} , et va vérifier son authenticité avec P_{MESR}

Le modèle hiérarchique : Cas de la fonction publique



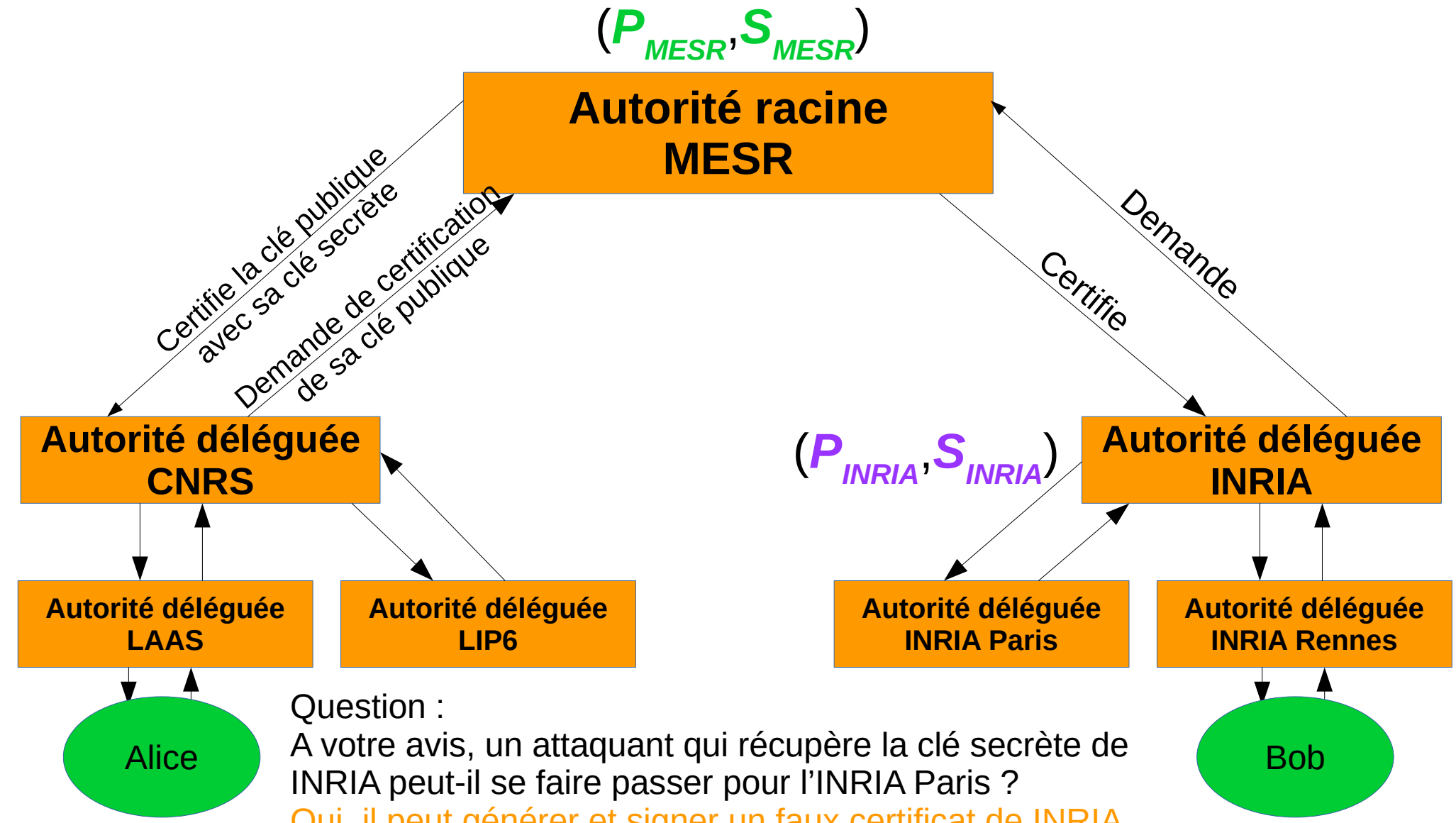
Alice fait de même pour vérifier la clé publique de INRIA Rennes à partir de P_{INRIA} , puis de Bob

Le modèle hiérarchique : Cas de la fonction publique



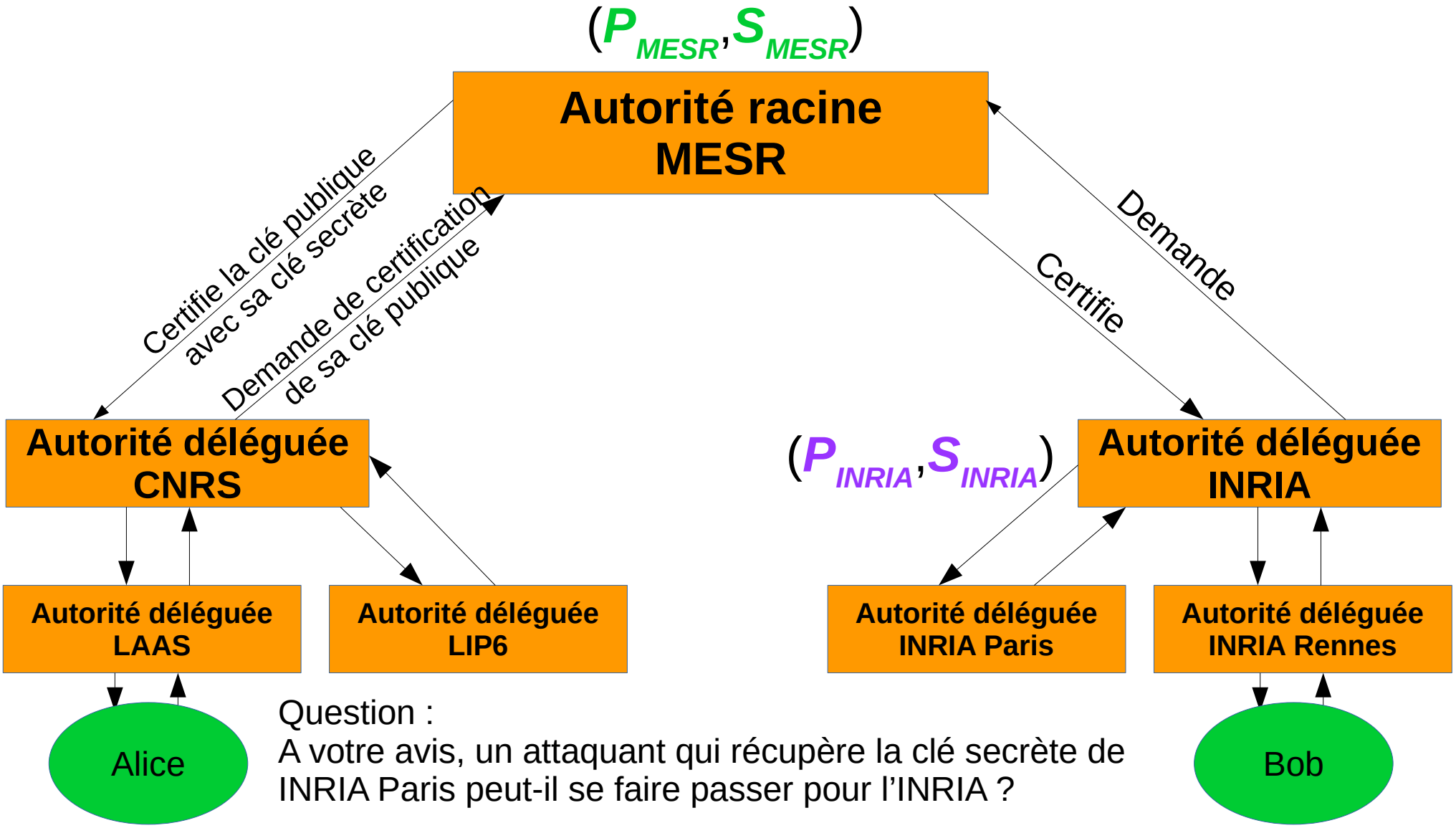
Question :
A votre avis, un attaquant qui récupère la clé secrète de INRIA peut-il se faire passer pour l'INRIA Paris ?

Le modèle hiérarchique : Cas de la fonction publique



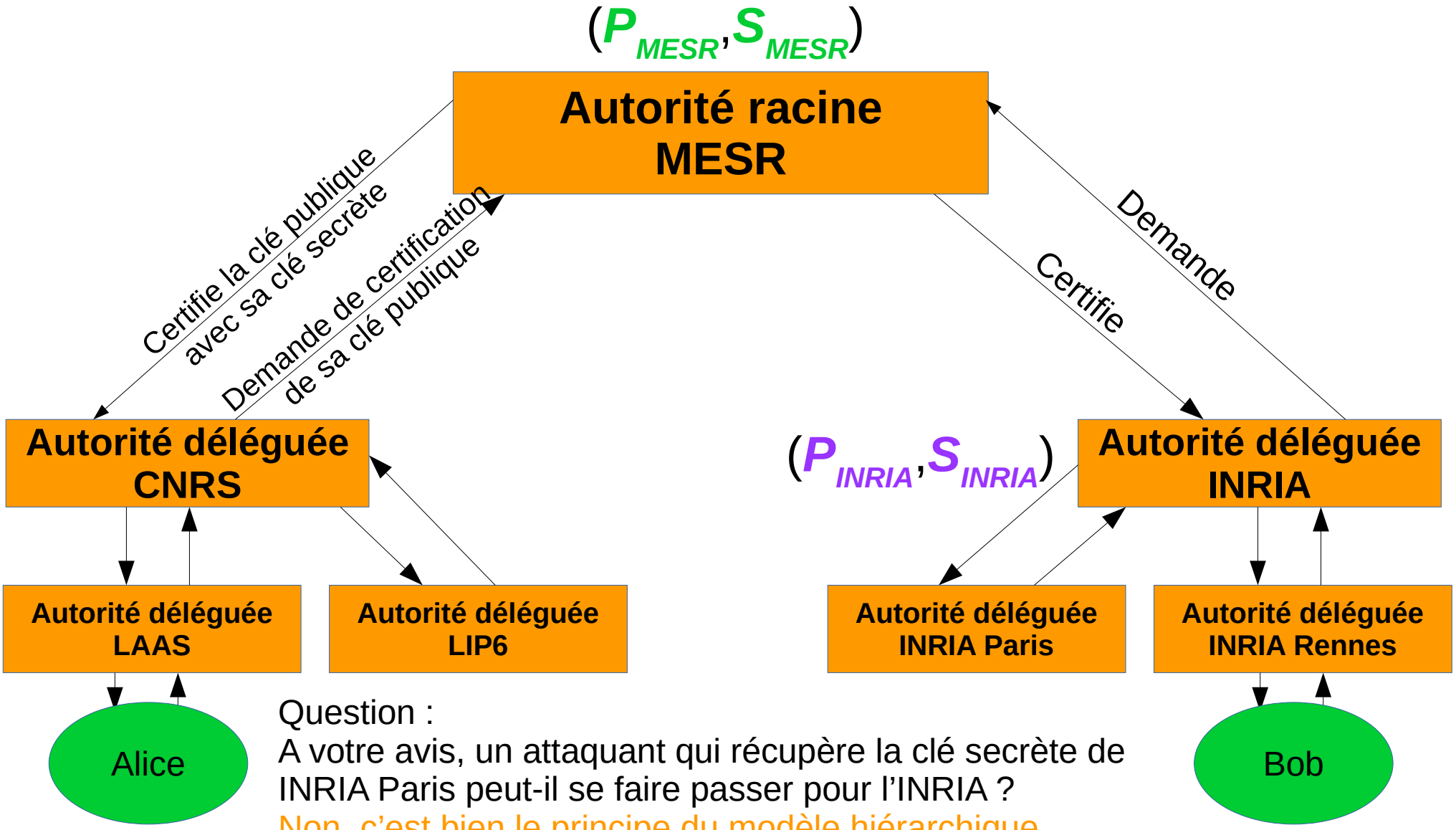
Question :
A votre avis, un attaquant qui récupère la clé secrète de INRIA peut-il se faire passer pour l'INRIA Paris ?
Oui, il peut générer et signer un faux certificat de INRIA Paris

Le modèle hiérarchique : Cas de la fonction publique



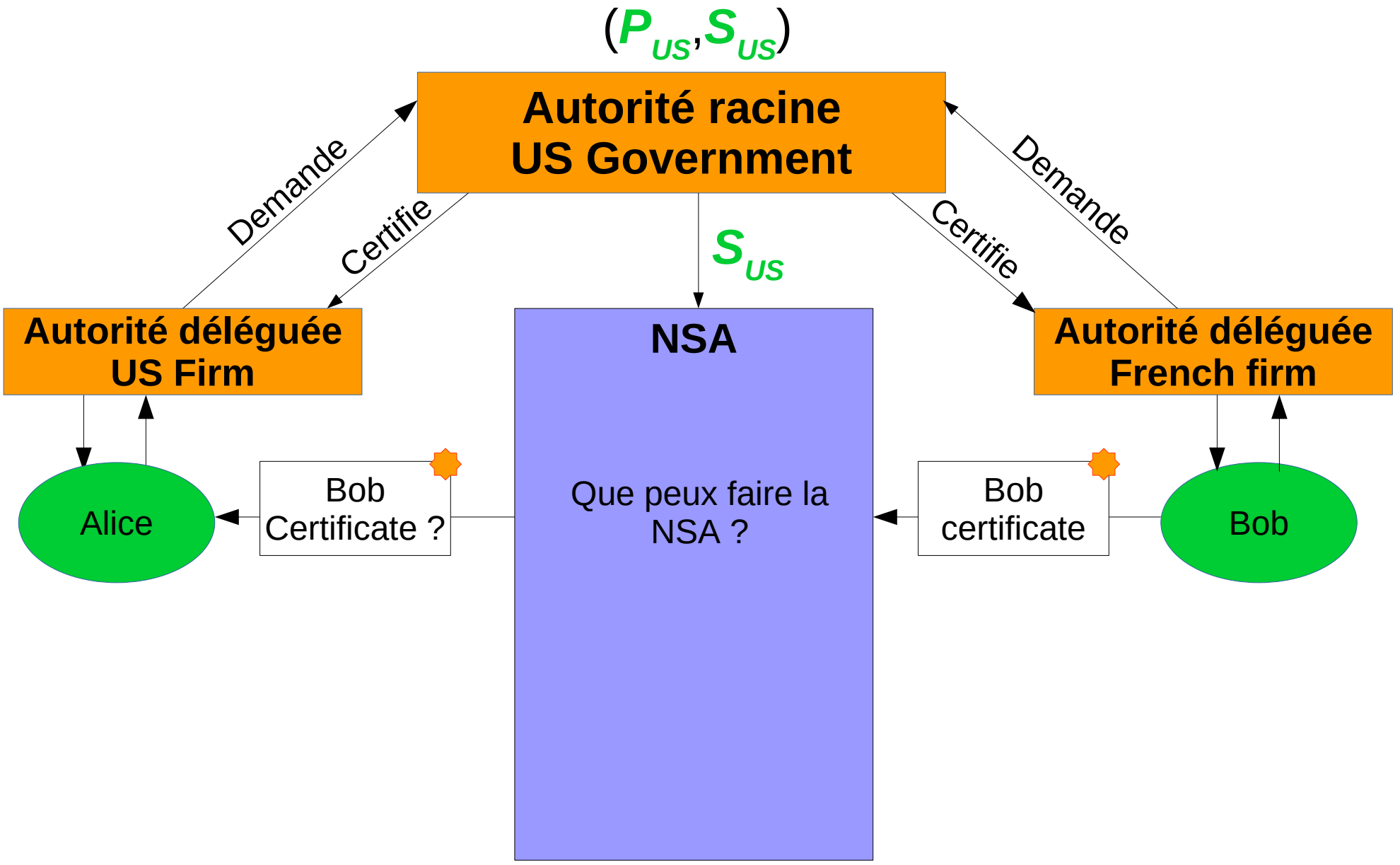
Question :
A votre avis, un attaquant qui récupère la clé secrète de INRIA Paris peut-il se faire passer pour l'INRIA ?

Le modèle hiérarchique : Cas de la fonction publique

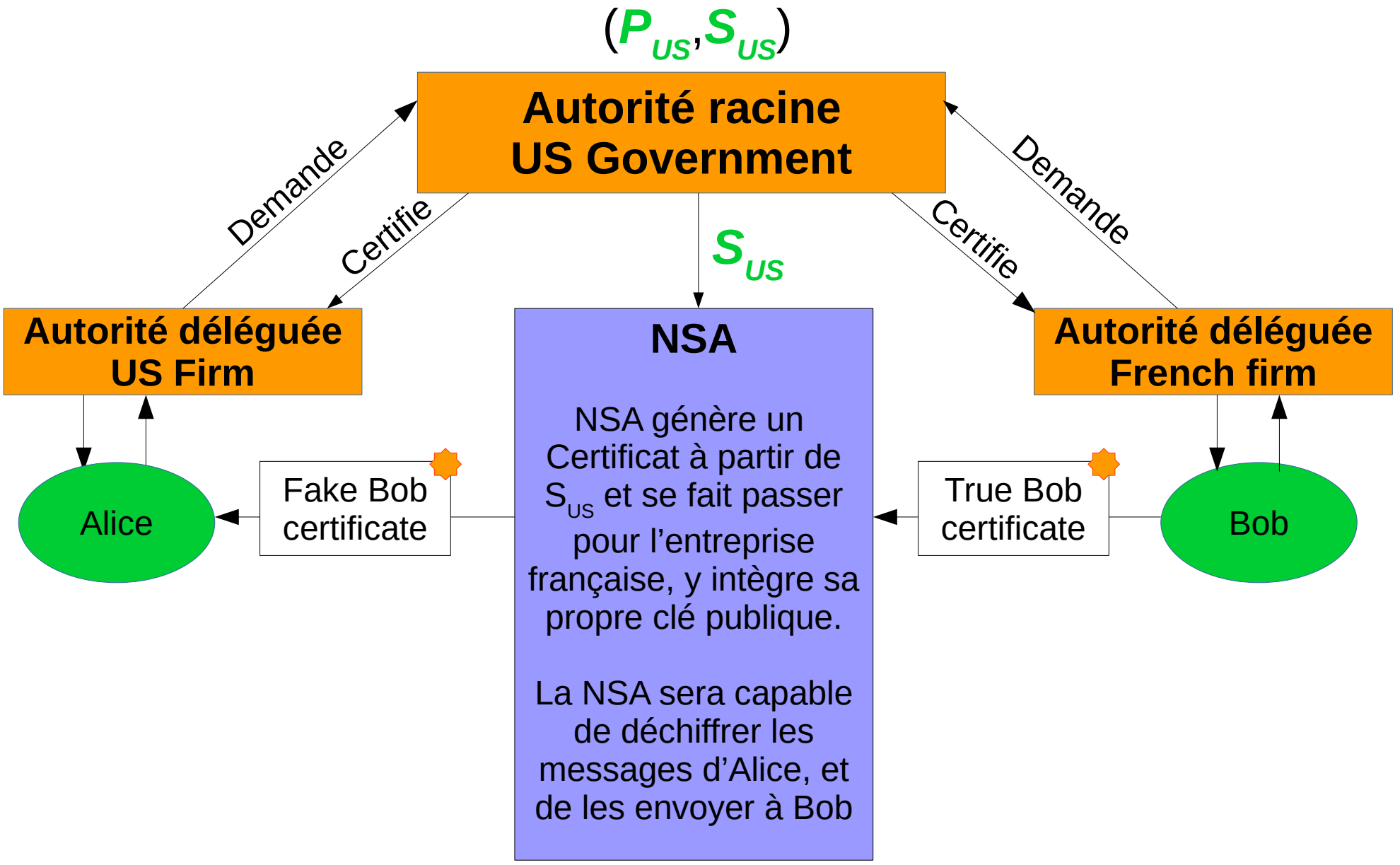


Question :
A votre avis, un attaquant qui récupère la clé secrète de INRIA Paris peut-il se faire passer pour l'INRIA ?
Non, c'est bien le principe du modèle hiérarchique

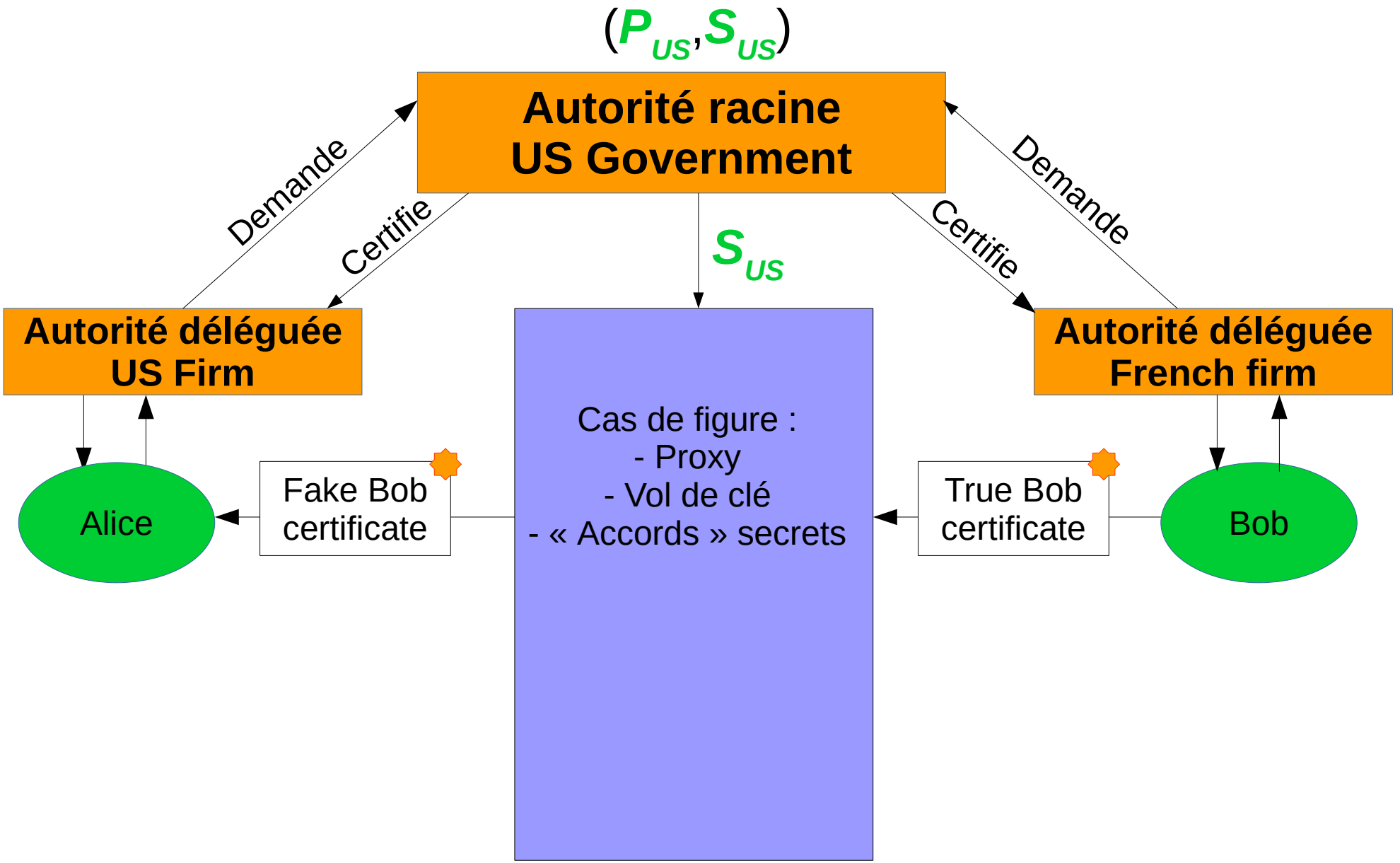
Attaque Man In The Middle



Attaque Man In The Middle



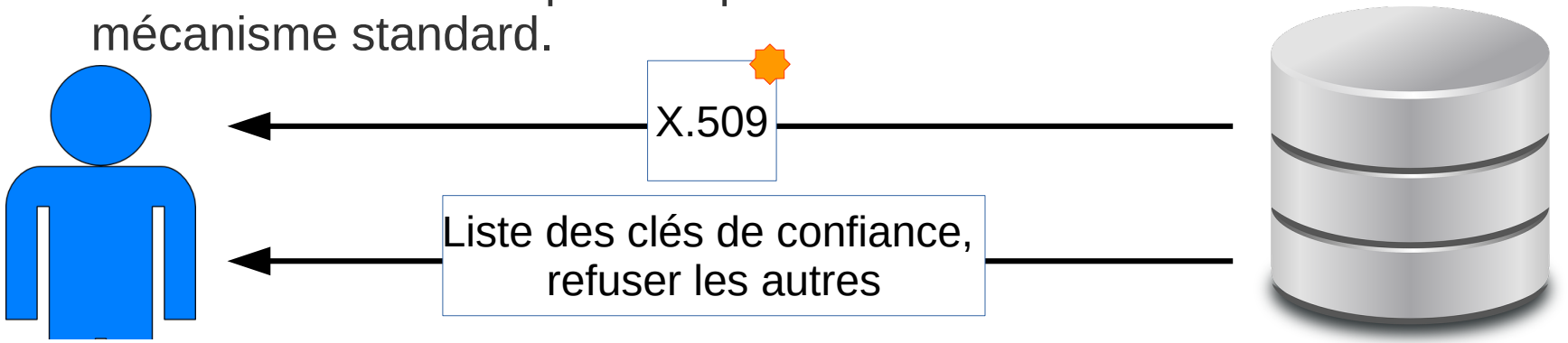
Attaque Man In The Middle



Contre-mesure face MITM attack, le Pinning

Principe :

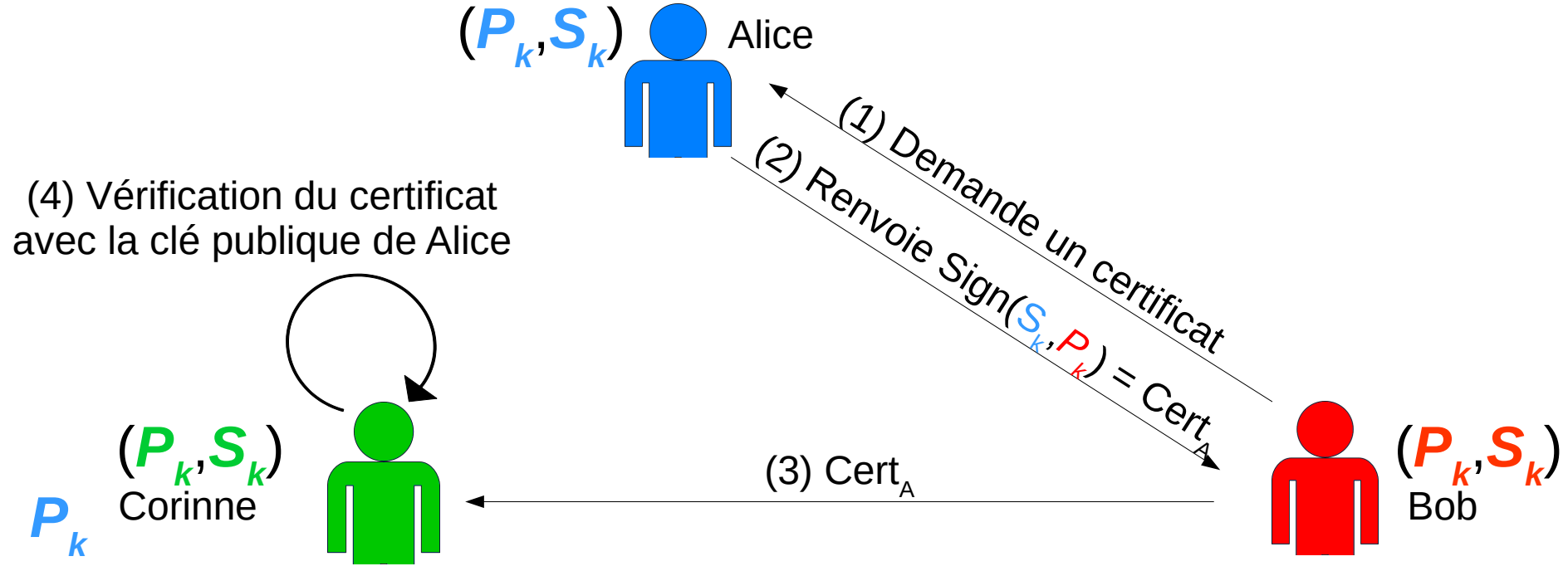
- ***Réduire*** sa confiance dans les ***Autorités de certification***.
- Pour des raisons de ***compatibilité***, de ***hiérarchisation*** ou de ***politique***, les serveurs possèdent ***plusieurs certificats provenant de plusieurs autorités***
- Le ***pinning*** est une démarche du serveur pour ***inciter les utilisateurs*** à n'accepter que ***certains certificats*** (considérés sûrs).
- Cela sous-entend que la première connexion se fait avec le mécanisme standard.



P_k

Principe de la toile de confiance (*Web of Trust*)

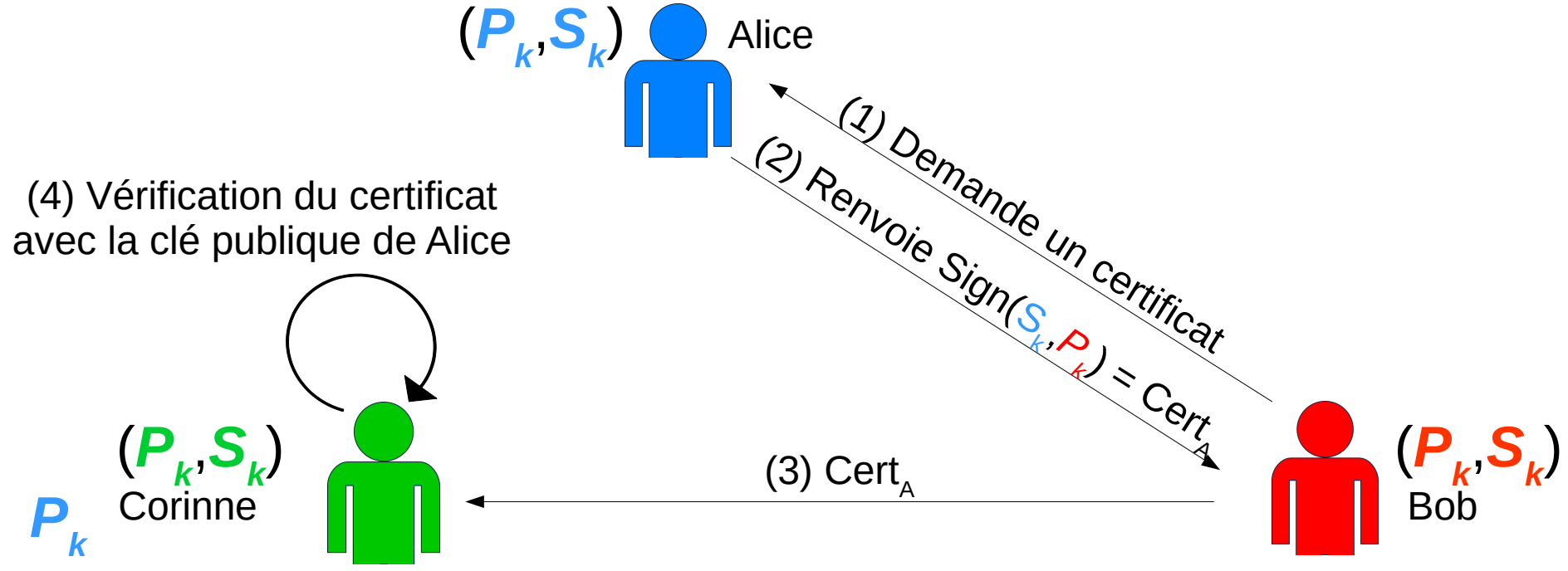
- Chaque participant agit comme un utilisateur et une autorité de confiance.
- Supposons de Corinne veuille communiquer avec Bob, sachant quelle fait confiance à Alice.
- Bob va présenter à Corinne un certificat qui a été signé par Alice. Corinne a donc confiance en ce certificat si elle peux le vérifier avec la clé publique d'Alice.



Principe de la toile de confiance (*Web of Trust*)

Quelques remarques :

- C'est le principe utilisé dans OpenPGP, utilisé pour l'échange de mail.
- Pour faire partie d'un nœud de la toile, il faut attendre qu'un nombre de personnes suffisant vous fasse confiance.



Le standard FIPS 186-4 : Digital Signature Standard

3 approches standardisées :

- **RSA :**
  - Problème de factorisation sur les corps finis.
  - Pour 112 bits de sécurité → $\log_2 N = 2048$.
  - Utilisation proche de FDH, ou padding potentiellement randomisé.
- **DSA (Digital Signature Algorithm) :**
  - Basé sur le logarithme discret sur les corps finis
  - Pour 112 bits de sécurité → $\log_2 p = 2048$
  - Construction de ype Fiat-Shamir (hors programme).
- **ECDSA (Elliptic-Curve Digital Signature Algorithm) :**
  - Basé sur le logarithme discret sur les courbes elliptiques.
  - Pour 112 bits de sécurité → $\log_2 p = 224$
  - Construction Fiat-Shamir également.

Principe de la révocation de certificat

Pour de bonnes raisons, il est nécessaire d'inclure un mécanisme de révocation de certificat. Exemples :

- Signifier qu'une ***clé a été volée*** (usurpation d'identité) ;
- Signifier qu'une ***clé n'est plus valide*** ;
- Inciter à la mise à jour des clés.

On utilise une liste de révocation de certificats (CRL) pour le vérifier :

- Soit on dispose d'une ***liste non périmée*** donc OK ;
- Soit on va ***chercher la dernière CRL*** ;
- Soit on utilise ***un OCSP*** (Online Certificate Status Protocol)