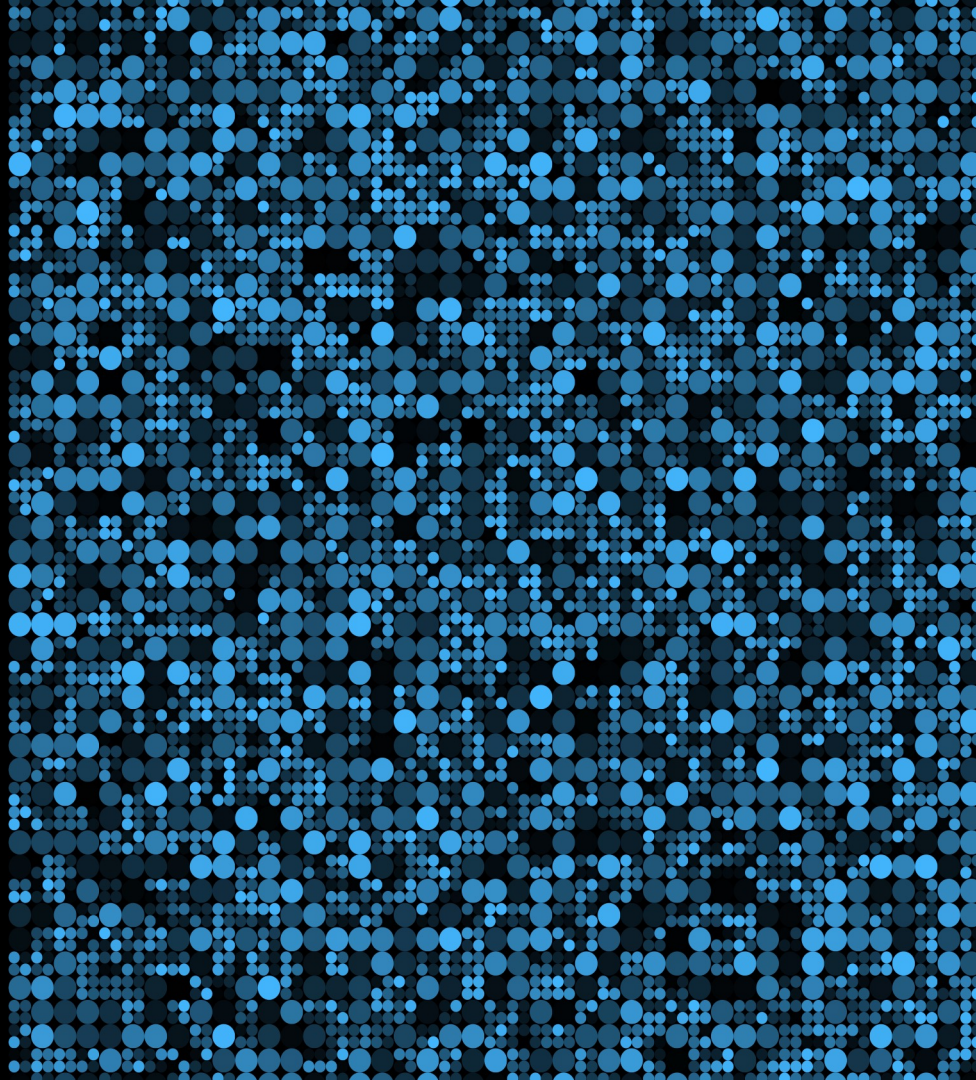
Cryptologie

Part 3

Bastien Vialla

bastien.vialla@orange.com

Année 2023-2024



Cours précédent

- **Partie maths :**
  - PGCD
  - Coefficients de Bézout & algorithme d'Euclide étendu
  - Arithmétique modulaire
  - Tests de primalité
  - Indicatrice d'Euler
  - Exponentiation rapide

- **Cryptographie asymétrique, ce qu'il faut retenir**
  - R.S.A.
  - El Gamal

Aujourd'hui : Applications

- **Fonction de hachage**
- **Echange de clé (Diffie-Helman)**
  - Comment créer une clé secrète en commun
- **Signatures numériques**
  - Certifier qu'un message n'est pas modifié
- **Certificats numériques**
  - Authentifier l'émetteur du message (personne, serveur, …)
- **TLS**
  - Etablir une connexion sécurisée avec un serveur

- **Cryptographie post-quantique**
  - L'impact de l'ordinateur quantique sur la cryptographie

Fonction de hachage

Fonctions de Hachages (ex. sha 256)

- **Fonction dont la taille de l'output est fixe (128bits, 256bits, …)**
  - Sha256('Hello') = 185f8db32271fe25f561a6fc938b2e264306ec304eda518007d1764826381969
  - Sha256('l') = acac86c0e609ca906f632b0e2dacccb2b77d22b0621f20ebece1a4835b93f6f0

Fonctions de Hachages (ex. sha 256)

- **Fonction dont la taille de l'output est fixe (128bits, 256bits, …)**

- **Maximise l'entropie de l'output**
  - 1bit changé dans l'input -> le hash de sortie est complétement différent
  - Sha256(2^20) = 90784ada4f1566a3d4cbd9a1b4c355d45c01a5ae6c8f9741699b835cea776d83
  - Sha256(2^20+1) = a6345f3f1a6df82a5d8aa5088b58ff576137502ee908cb1e09c6198eb07e831d

Fonctions de Hachages (ex. sha 256)

- **Fonction dont la taille de l'output est fixe (128bits, 256bits, …)**

- **Maximise l'entropie de l'output**
  - 1bit changé dans l'input -> le hash de sortie est complétement différent

- **Déterministe**
  - Pour 2 inputs identiques -> même hash en output

Fonctions de Hachages (ex. sha 256)

- **Fonction dont la taille de l'output est fixe (128bits, 256bits, …)**
- **Maximise l'entropie de l'output**
  - 1bit changé dans l'input -> le hash de sortie est complétement différent
- **Déterministe**
  - Pour 2 inputs identiques -> même hash en output
- **Sens unique**
  - Il est impossible de retrouver l'input à partir de l'output

Fonctions de Hachages (ex. sha 256)

- **Fonction dont la taille de l'output est fixe (128bits, 256bits, …)**
- **Maximise l'entropie de l'output**
  - 1bit changé dans l'input -> le hash de sortie est complétement différent
- **Déterministe**
  - Pour 2 inputs identiques -> même hash en output
- **Sens unique**
  - Il est impossible de retrouver l'input à partir de l'output
- **Résistant à la collision**
  - La probabilité que 2 inputs différents donnent le même hash est presque nulle
    - Sha256 : $< 2^{128}$

Fonctions de Hachages (ex. sha 256)

- **Résistant à la collision**
  - La probabilité que 2 inputs différents donnent le même hash est presque nulle
    - Sha256 : $< 2^{128}$

- **Attaque « brute force » sur sha256**
  - Chercher une collision
  - Le réseau bitcoin teste $\sim 4.56 \times 10^{20}$ hashs par seconde (13/10/2023)
  - 30758400 secondes par année
  - 2.42×10^{10} années de calcul
  - Age de l'univers 1.38×10^{10} années

Protocole d’échange de clés (Diffie-Hellman)

Diffie-Hellman

- **Objectif : Alice et Bob doivent s'accorder sur un secret en commun**
  - Le secret se nome « pre-master secret »
  - Le secret sert à générer une clé maître (master key)
    - Peut servir à créer de nouvelles clés en utilisant « key derivation function »
    - Peut servir de clé secrète pour cryptographie symétrique

Diffie-Hellman

- **Objectif : Alice et Bob doivent s'accorder sur un secret en commun**
  - Le secret se nome « pre-master secret »
  - Le secret sert à générer une clé maître (master key)
    - Peut servir à créer de nouvelles clés en utilisant « key derivation function »
    - Peut servir de clé secrète pour cryptographie symétrique

- **Se base sur le problème du logarithme discret**
  - Soit g un générateur de $(\mathbb{Z}/N\mathbb{Z}, \times)$, $N \geq 2$ un nombre entier. Soit $r \in \mathbb{Z}/N\mathbb{Z}$ aléatoire, le problème du logarithme discret est : retrouver r connaissant $g^r \bmod N$

Diffie-Hellman

- **Objectif : Alice et Bob doivent s'accorder sur un secret en commun**
  - Le secret se nome « pre-master secret »
  - Le secret sert à générer une clé maître (master key)
    - Peut servir à créer de nouvelles clés en utilisant « key derivation function »
    - Peut servir de clé secrète pour cryptographie symétrique

- **Se base sur le problème du logarithme discret**
  - Soit g un générateur de $(\mathbb{Z}/N\mathbb{Z}, \times)$, $N \geq 2$ un nombre entier. Soit $r \in \mathbb{Z}/N\mathbb{Z}$ aléatoire, le problème du logarithme discret est : retrouver r connaissant $g^r \bmod N$

- **Alice et Bob se partagent en g et N, ainsi que H une fonction de hachage**

Diffie-Hellman

Alice

g, N

g, N, H

Bob

g, N

Diffie-Hellman

Alice

g, N

$a \in [2, N-1]$ **aléatoire**

g, N, H

Bob

g, N

$b \in [2, N-1]$ **aléatoire**

Diffie-Hellman

Alice

g, N

$a \in [2, N-1]$ **aléatoire**

$A = g^a \bmod N$

g, N, H

Bob

g, N

$b \in [2, N-1]$ **aléatoire**

$B = g^b \bmod N$

Diffie-Hellman

Alice

Bob

g, N, H

$g^a \bmod N, g^b \bmod N$

g, N

$a \in [2, N-1]$ **aléatoire**

$A = g^a \bmod N$

$B = g^b \bmod N$

g, N

$b \in [2, N-1]$ **aléatoire**

$B = g^b \bmod N$

$A = g^a \bmod N$

Diffie-Hellman

g, N, H

$g^a \bmod N, g^b \bmod N$

Alice

g, N

$a \in [2, N-1]$ **aléatoire**

$A = g^a \bmod N$

$B = g^b \bmod N$

$(B)^a = g^{ab} \bmod N$

Bob

g, N

$b \in [2, N-1]$ **aléatoire**

$B = g^b \bmod N$

$A = g^a \bmod N$

$(A)^b = g^{ab} \bmod N$

Diffie-Hellman

g, N, H

$g^a \bmod N, g^b \bmod N$

Alice

g, N

$a \in [2, N-1]$ **aléatoire**

$A = g^a \bmod N$

$B = g^b \bmod N$

$(B)^a = g^{ab} \bmod N$

Bob

g, N

$b \in [2, N-1]$ **aléatoire**

$B = g^b \bmod N$

$A = g^a \bmod N$

$(A)^b = g^{ab} \bmod N$

pre-master secret

Diffie-Hellman

g, N, H

$g^a \bmod N, g^b \bmod N$

Alice

g, N

$a \in [2, N-1]$ **aléatoire**

$A = g^a \bmod N$

$B = g^b \bmod N$

$(g^b)^a = g^{ab} \bmod N$

$sk = H(g^{ab} \bmod N)$

Bob

g, N

$b \in [2, N-1]$ **aléatoire**

$B = g^b \bmod N$

$A = g^a \bmod N$

$(g^a)^b = g^{ab} \bmod N$

$sk = H(g^{ab} \bmod N)$

master secret key

Signatures numériques

Signatures numériques

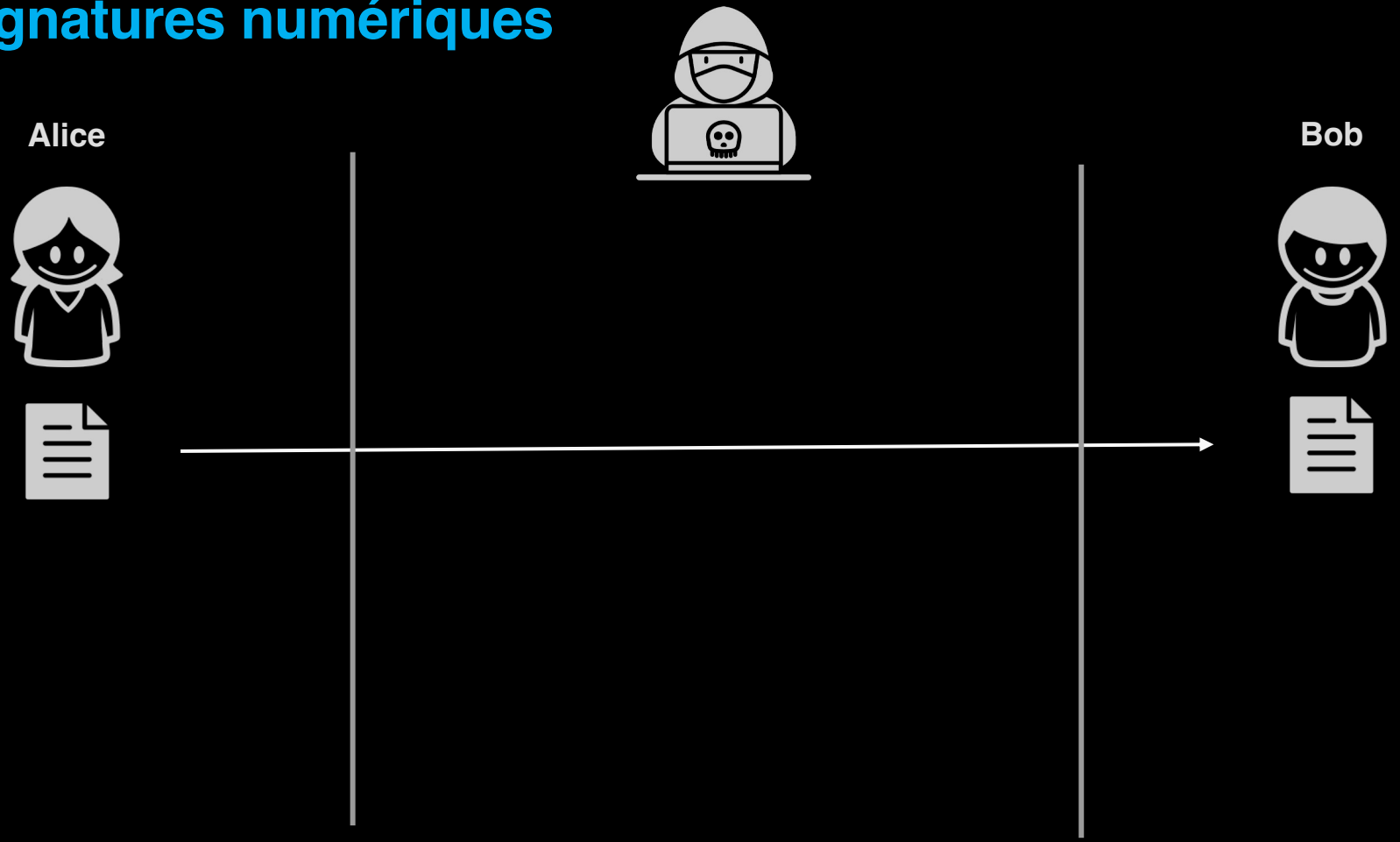
- **Principes**
  - Authentification :
    - Permet de s’assurer de l’origine du message
  - Intégrité :
    - Permet de s’assurer que le message échangé n’est pas modifié

Signatures numériques

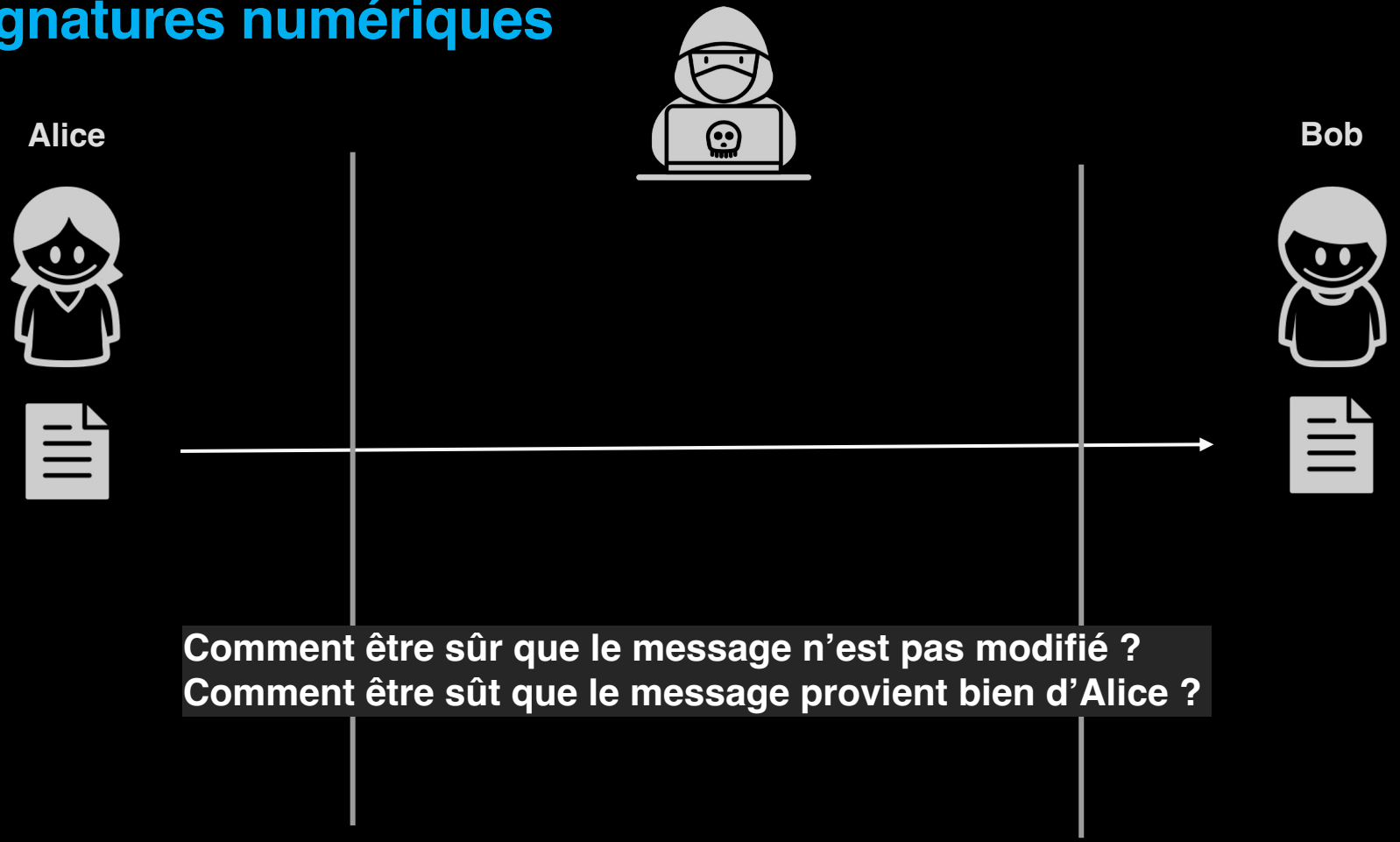
Alice

Bob

Signatures numériques

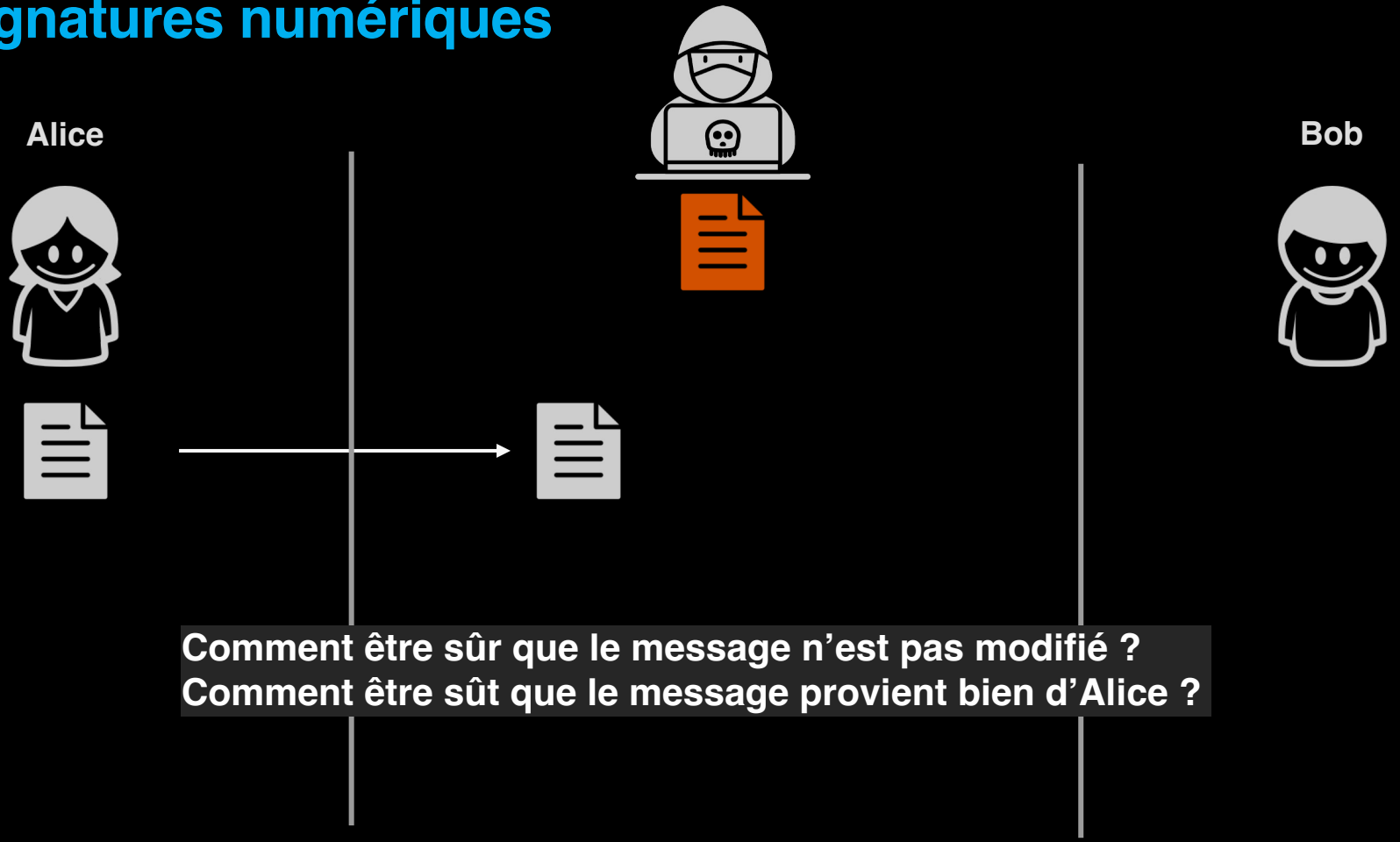


Signatures numériques



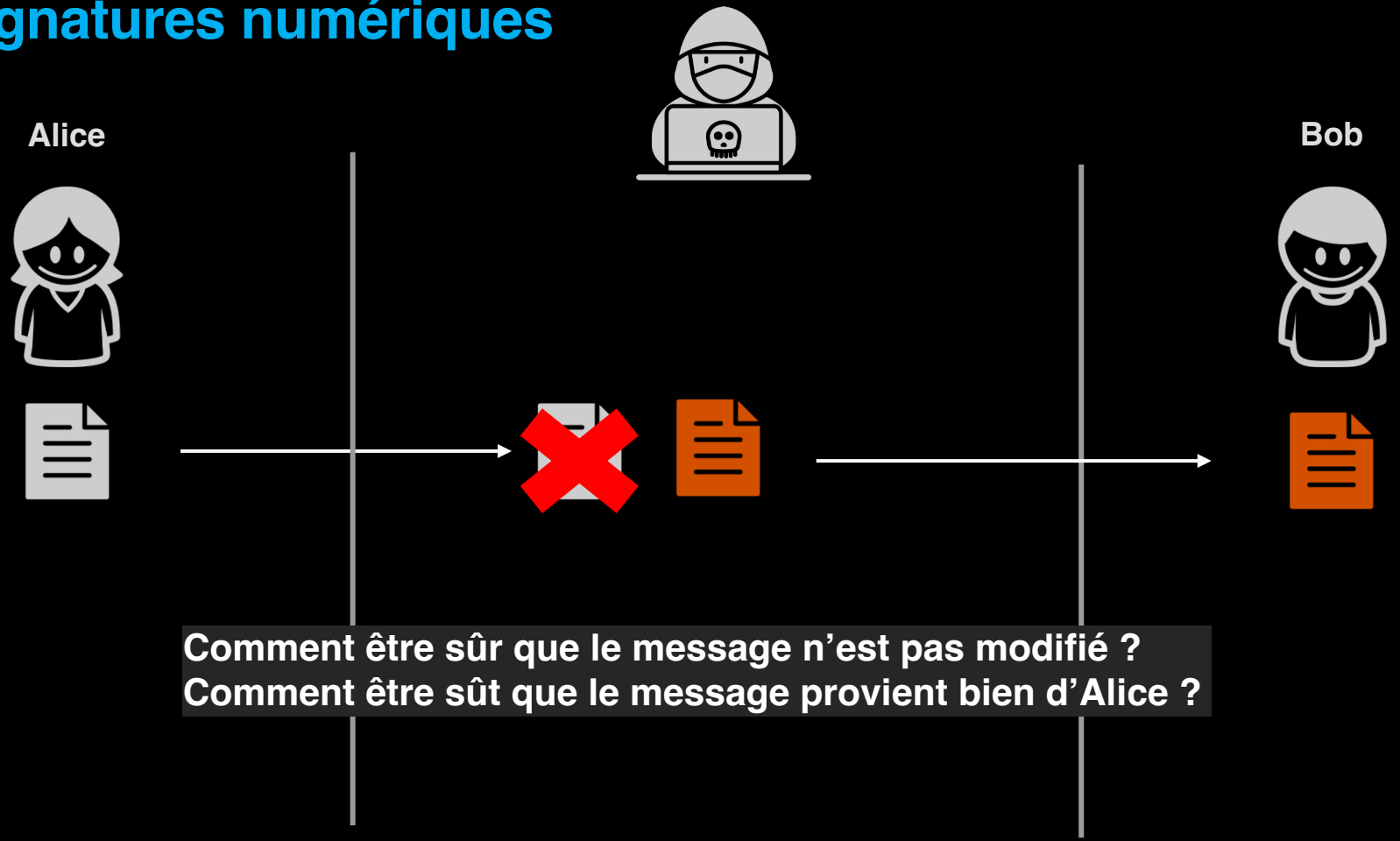
Comment être sûr que le message n’est pas modifié ?
Comment être sût que le message provient bien d’Alice ?

Signatures numériques



Comment être sûr que le message n'est pas modifié ?
Comment être sût que le message provient bien d'Alice ?

Signatures numériques



Comment être sûr que le message n'est pas modifié ?
Comment être sût que le message provient bien d'Alice ?

Signatures numériques

- **Intégrité :**
  - Calcul d’un hash du message
  - Comment garantir que le est bien générer par Alice ? (Authentification)

Signatures numériques

- **Intégrité :**
  - Calcul d'un hash du message
  - Comment garantir que le est bien générer par Alice ? (Authentification)

- **R.S.A.**
  - Soit $N = p \cdot q$ le produit de deux premiers.
  - Soient e, d deux entiers tels que $ed \equiv 1 \bmod \phi(N)$
  - $pk = (N, e),\ sk = (d, p, q)$
  - Chiffrer : $m \in \mathbb{Z}/N\mathbb{Z},\ c \equiv m^e \bmod N$
  - Déchiffrer : $c^d \bmod N = (m^e)^d \bmod N = m$

Signatures numériques

- **R.S.A.**
  - Soit $N = p \cdot q$ le produit de deux premiers.
  - Soient e, d deux entiers tels que $ed \equiv 1 \bmod \phi(N)$
  - $pk = (N, e),\ sk = (d, p, q)$
  - Chiffrer :
    - avec pk : $m \in \mathbb{Z}/N\mathbb{Z},\ c \equiv m^e \bmod N$
    - avec sk : $m \in \mathbb{Z}/N\mathbb{Z},\ c \equiv m^d \bmod N$
  - Déchiffrer :
    - avec sk : $c^d \bmod N = (m^e)^d \bmod N = m^{ed}\ mod\ N = m$
    - avec pk : $c^e \bmod N = \left(m^d\right)^e \bmod\ N = m^{de} \bmod N = m$

Signatures numériques

- **R.S.A.**
  - Chiffrer avec la clé publique, déchiffrer avec la clé secrète

Alice

Bob

Daniel

Signatures numériques

- **R.S.A.**
  - Chiffrer avec la clé publique, déchiffrer avec la clé secrète

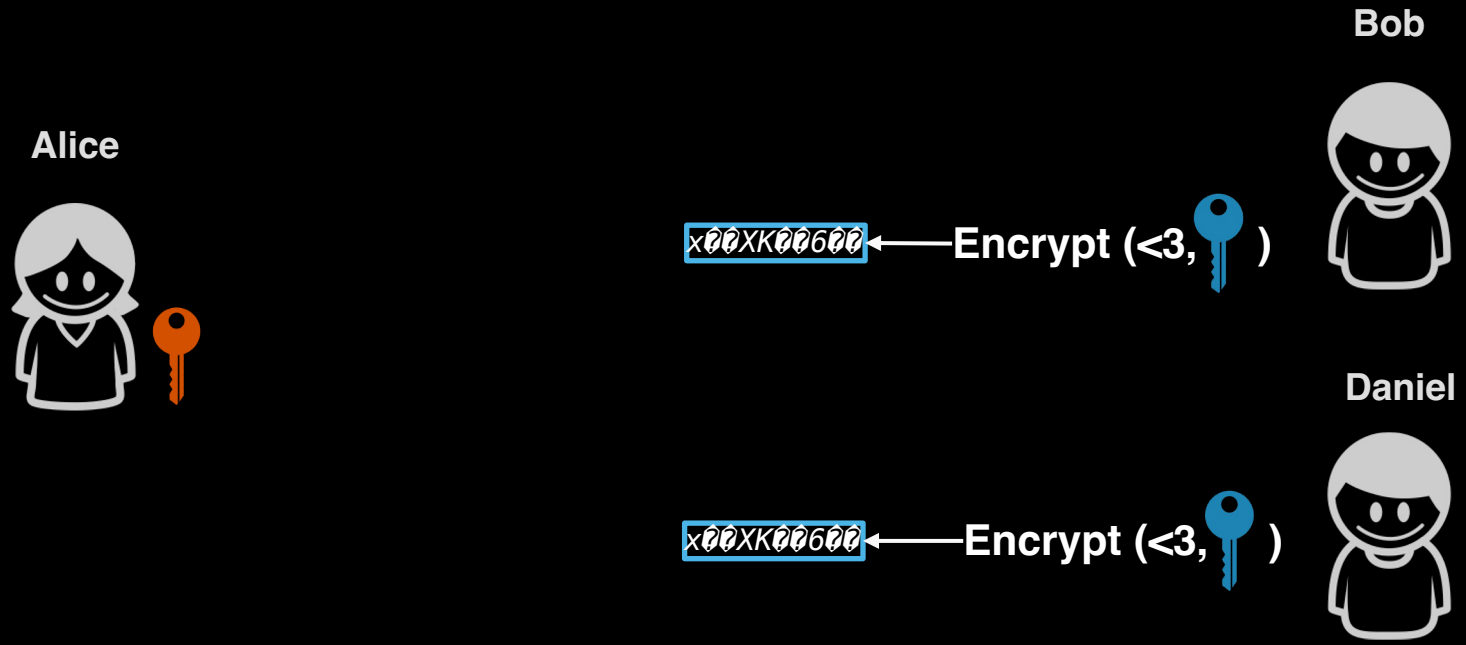
Alice

Bob

Daniel

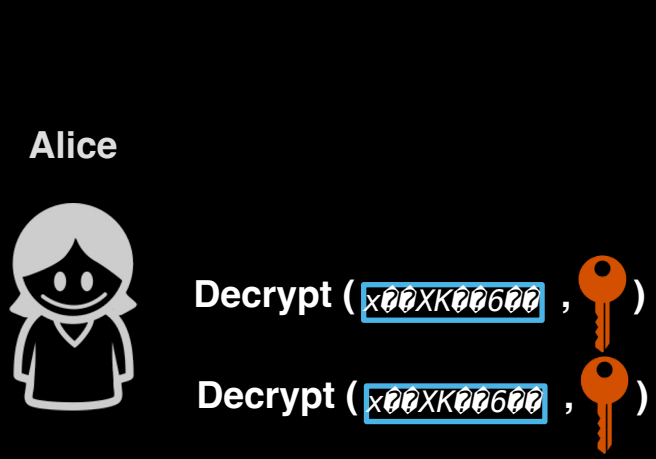
Signatures numériques

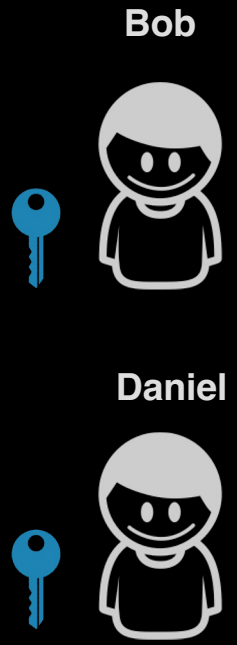
- **R.S.A.**
  - Chiffrer avec la clé publique, déchiffrer avec la clé secrète



Signatures numériques

- **R.S.A.**
  - Chiffrer avec la clé publique, déchiffrer avec la clé secrète





Signatures numériques

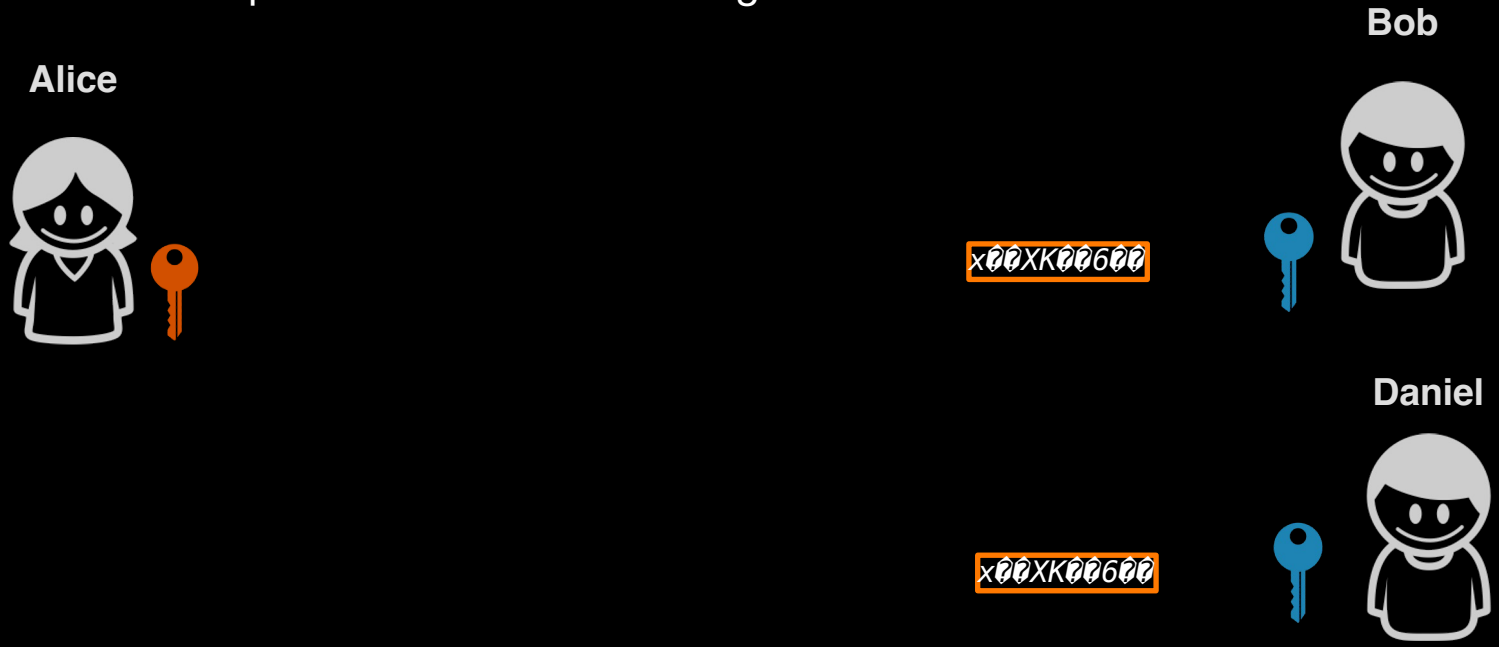
- **R.S.A.**
  - Chiffrer avec la clé publique, déchiffrer avec la clé secrète
  - Permet d'être sûr que seul le destinataire du message peut le lire

Signatures numériques

- **R.S.A.**
  - Chiffrer avec la clé secrète , déchiffrer avec la clé publique
  - Permet d'être sûr que le message provient de l'émetteur, seule la personne ayant la clé secrète peut avoir chiffré le message

Signatures numériques

- **R.S.A.**
  - Chiffrer avec la clé secrète , déchiffrer avec la clé publique
  - Permet d'être sûr que le message provient de l'émetteur, seule la personne ayant la clé secrète peut avoir chiffré le message



Signatures numériques

- **R.S.A.**
  - Chiffrer avec la clé secrète , déchiffrer avec la clé publique
  - Permet d'être sûr que le message provient de l'émetteur, seule la personne ayant la clé secrète peut avoir chiffré le message

Signatures numériques

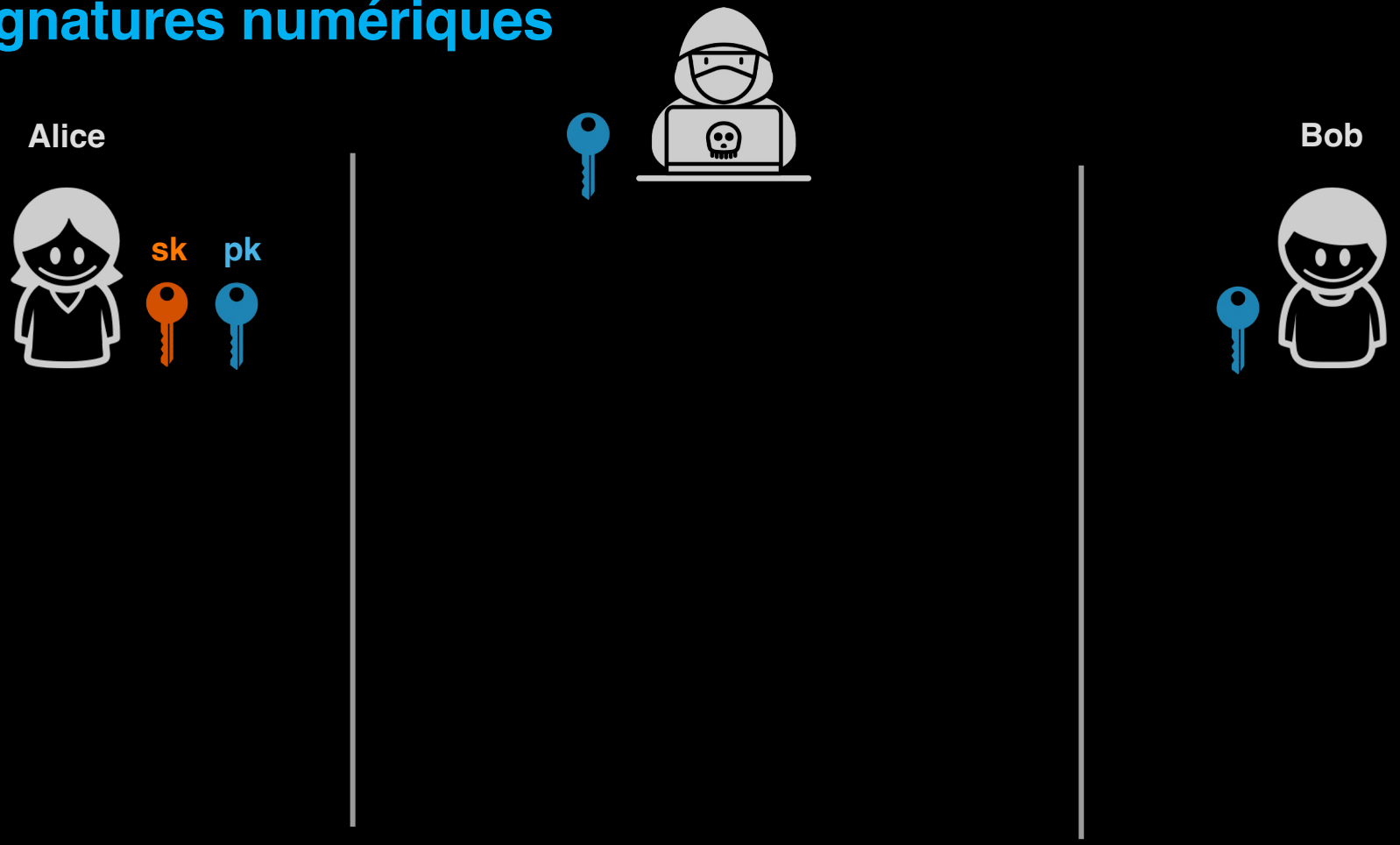
Alice

sk

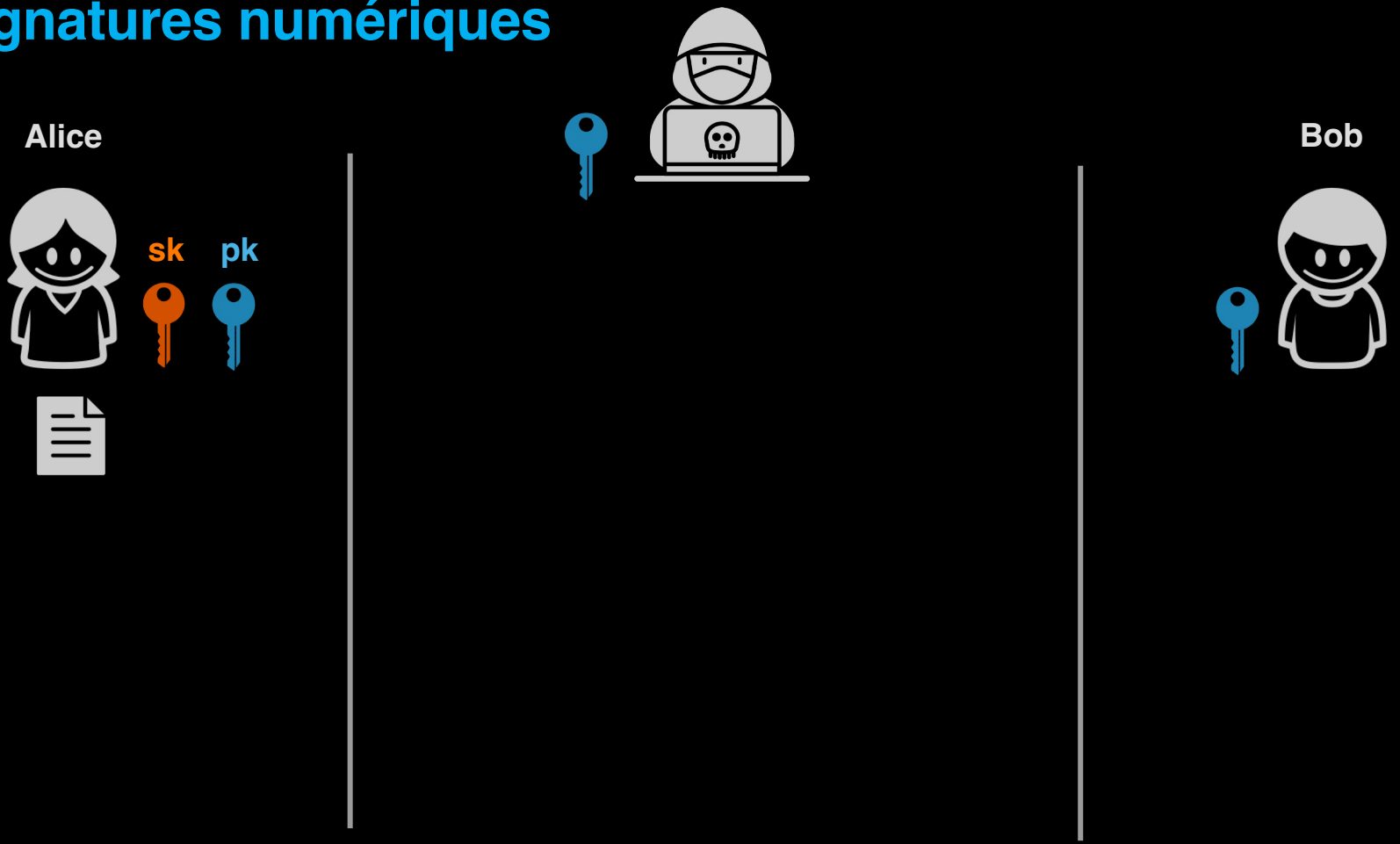
pk

Bob

Signatures numériques



Signatures numériques



Signatures numériques

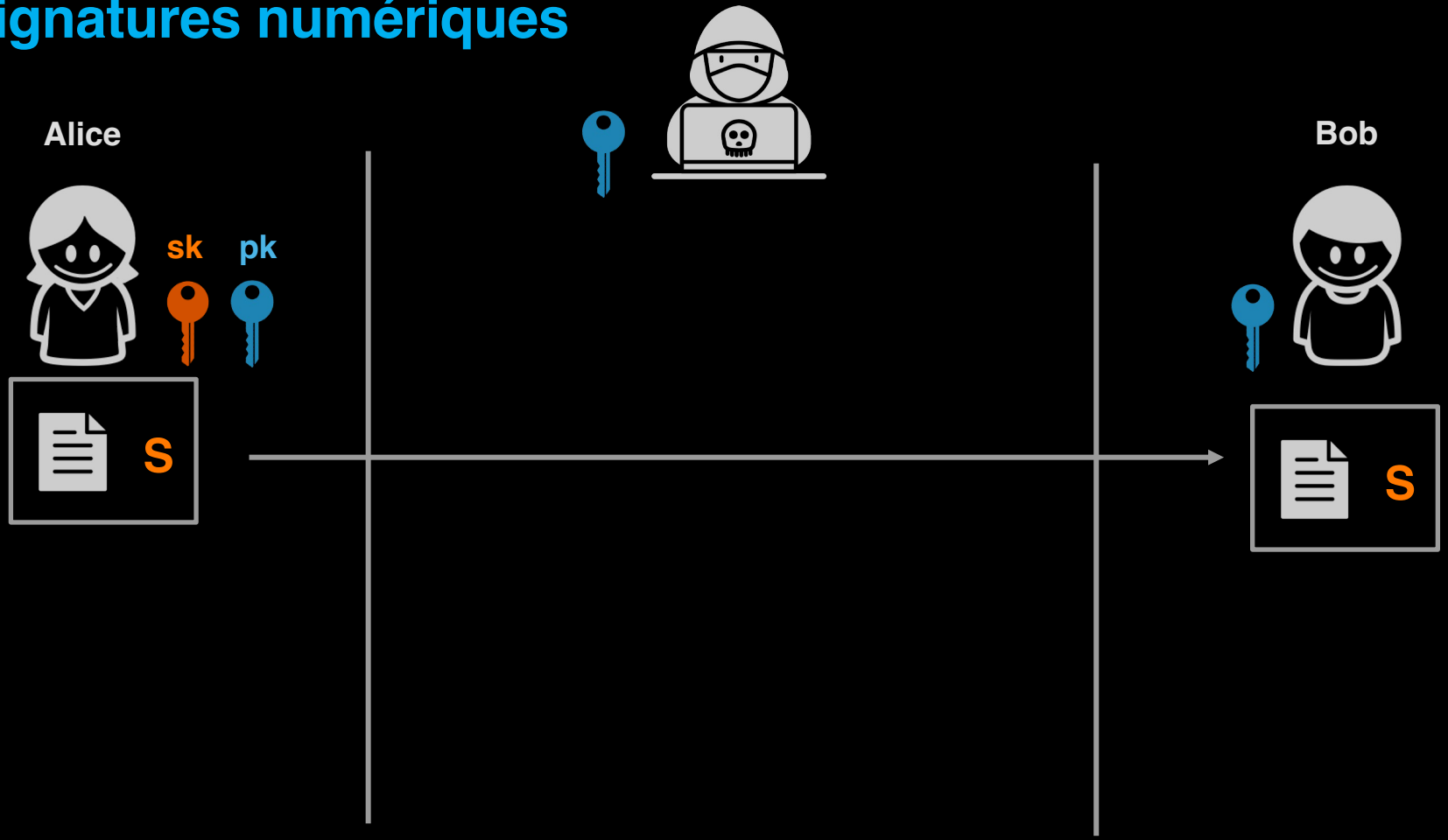
Alice

sk pk

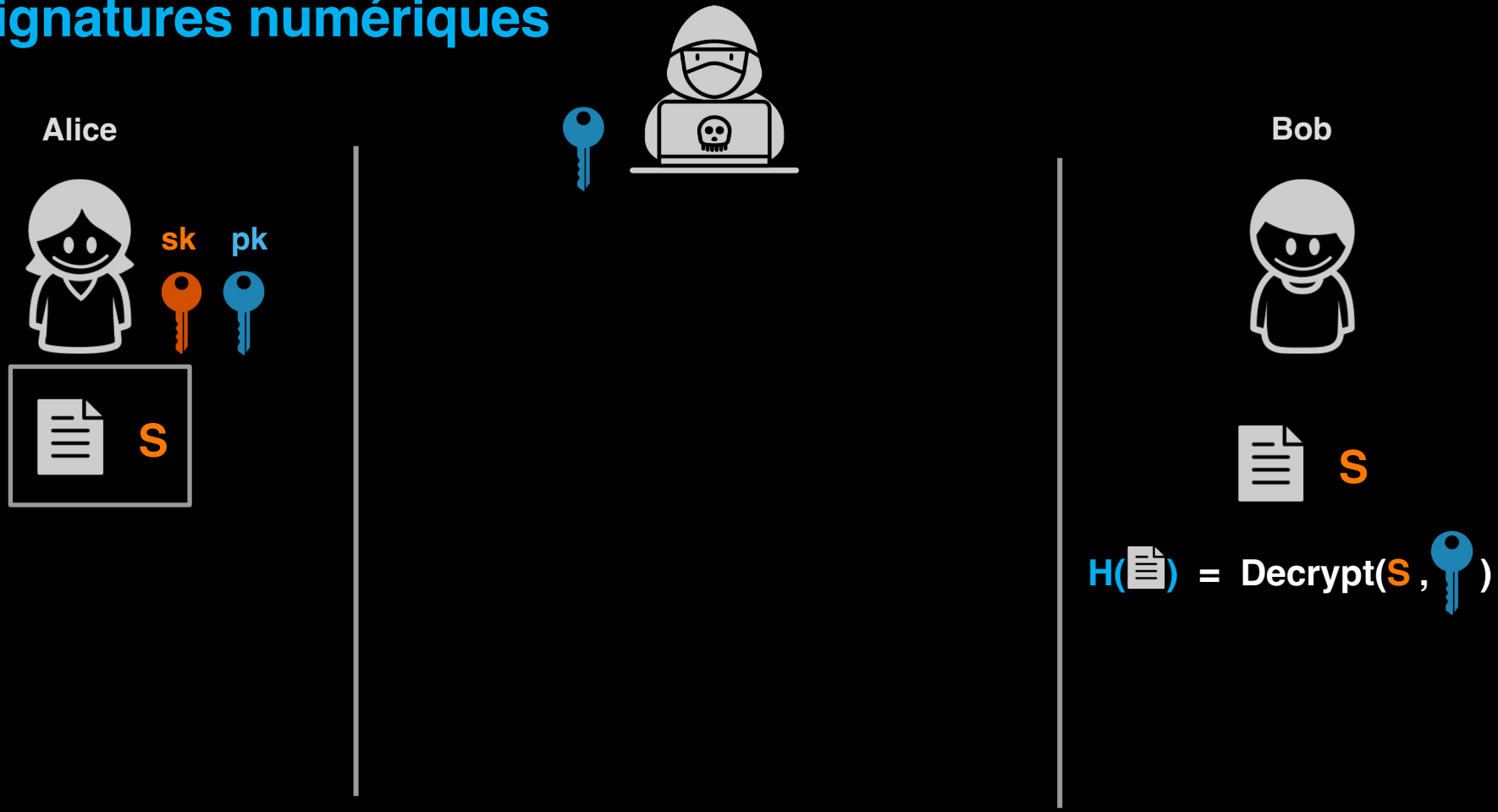
S = Encrypt(H(), )

Bob

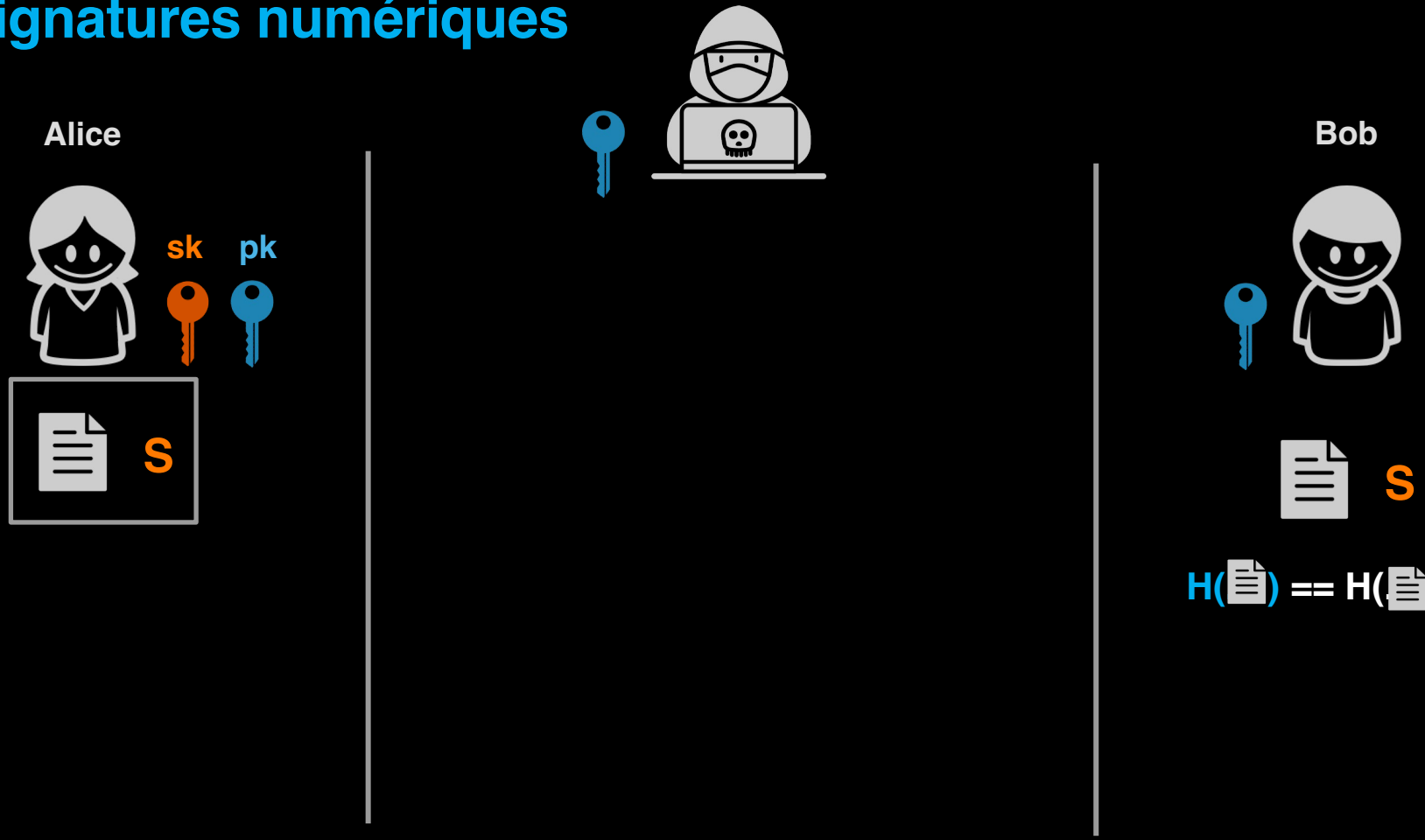
Signatures numériques



Signatures numériques



Signatures numériques



Signatures numériques

- **Attaques :**
  - Si le documents est modifié, Bob peut le détecter
  - L’attaquant ne peut créer une nouvelle signature, il n’a pas la clé secrète

Signatures numériques

- **Attaques :**
  - Si le documents est modifié, Bob peut le détecter
  - L'attaquant ne peut créer une nouvelle signature, il n'a pas la clé secrète

- **Conditions :**
  - Il faut que Bob connaisse Alice pour récupérer sa clé publique
  - Comment Bob fait s'il ne connait pas Alice ?

Certificats numériques

Certificats numériques

- **Alice veut se connecter à *unicaen.fr***
  - Comment être sûr qu'elle se connecte sur le bon site / serveur ?

Alice

unicaen.fr

Certificats numériques

- **Un certificat est une message avec une signature**
  - Le message contient :
    - Les noms de domaines associés (*.unicaen.fr)
    - L'organisation (Université de Caen Normandie)
    - Localisation (Normandie, FR)
    - Date de validité
    - Algorithme de signature (SHA256 & RSA)
    - Clé publique du serveur
    - Identité de l'émetteur du certificat
    - …

Certificats numériques

- **Un certificat est une message avec une signature**
  - Le message contient :
    - Les noms de domaines associés (*.unicaen.fr)
    - L'organisation (Université de Caen Normandie)
    - Localisation (Normandie, FR)
    - Date de validité
    - Algorithme de signature (SHA256 & RSA)
    - Clé publique du serveur
    - Identité de l'émetteur du certificat
    - …

- **Par qui est signé le certificat ?**

Certificats numériques

- **Un certificat est une message avec une signature**
  - Le message contient :
    - Les noms de domaines associés (*.unicaen.fr)
    - L'organisation (Université de Caen Normandie)
    - Localisation (Normandie, FR)
    - Date de validité
    - Algorithme de signature (SHA256 & RSA)
    - Clé publique du serveur
    - Identité de l'émetteur du certificat
    - …

- **Par qui est signé le certificat ?**
  - Une autorité de certification (Sectigo Limited)

Autorité de certifications

- **Rôle :**
  - Émettre des certificats :
    - Vérifie l'identité du demandeur et des infos fournis
      - Plusieurs de vérification suivant le prix payé
  - Renouveler les certificats
  - Révoquer les certificats :
    - Publie une liste de certificats révoqués
    - Propose un « Online Certificate Status Protocol » (OCSP) pour tester en temps-réel la validité

- **La clé publique de l'autorité est distribuée dans les navigateurs et les OS**
- **La secrète sert à signer les certificats**

Certificats numériques

unicaen.fr

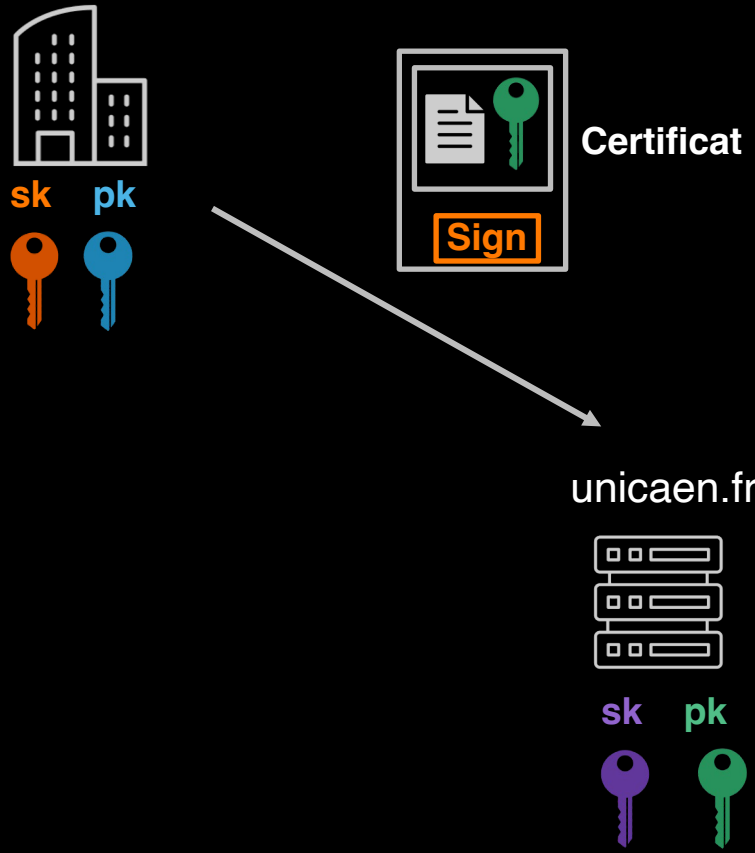
Certificats numériques

Identité, infos, clé publique

unicaen.fr

UNIVERSITÉ CAEN NORMANDIE

Certificats numériques



Certificats numériques

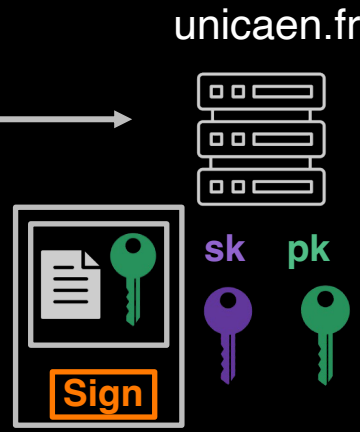
Alice

unicaen.fr

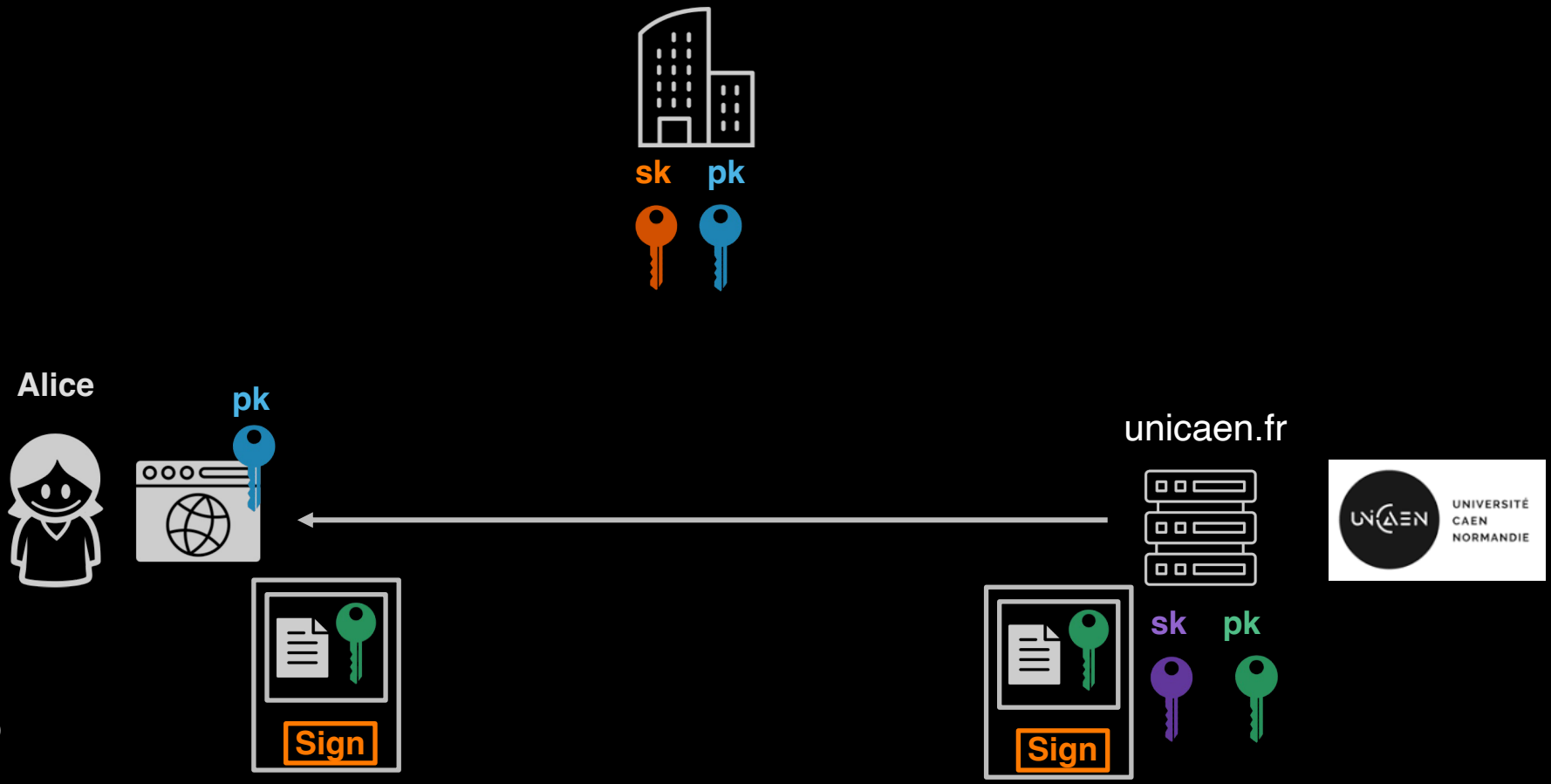
UNIVERSITÉ CAEN NORMANDIE

Certificats numériques

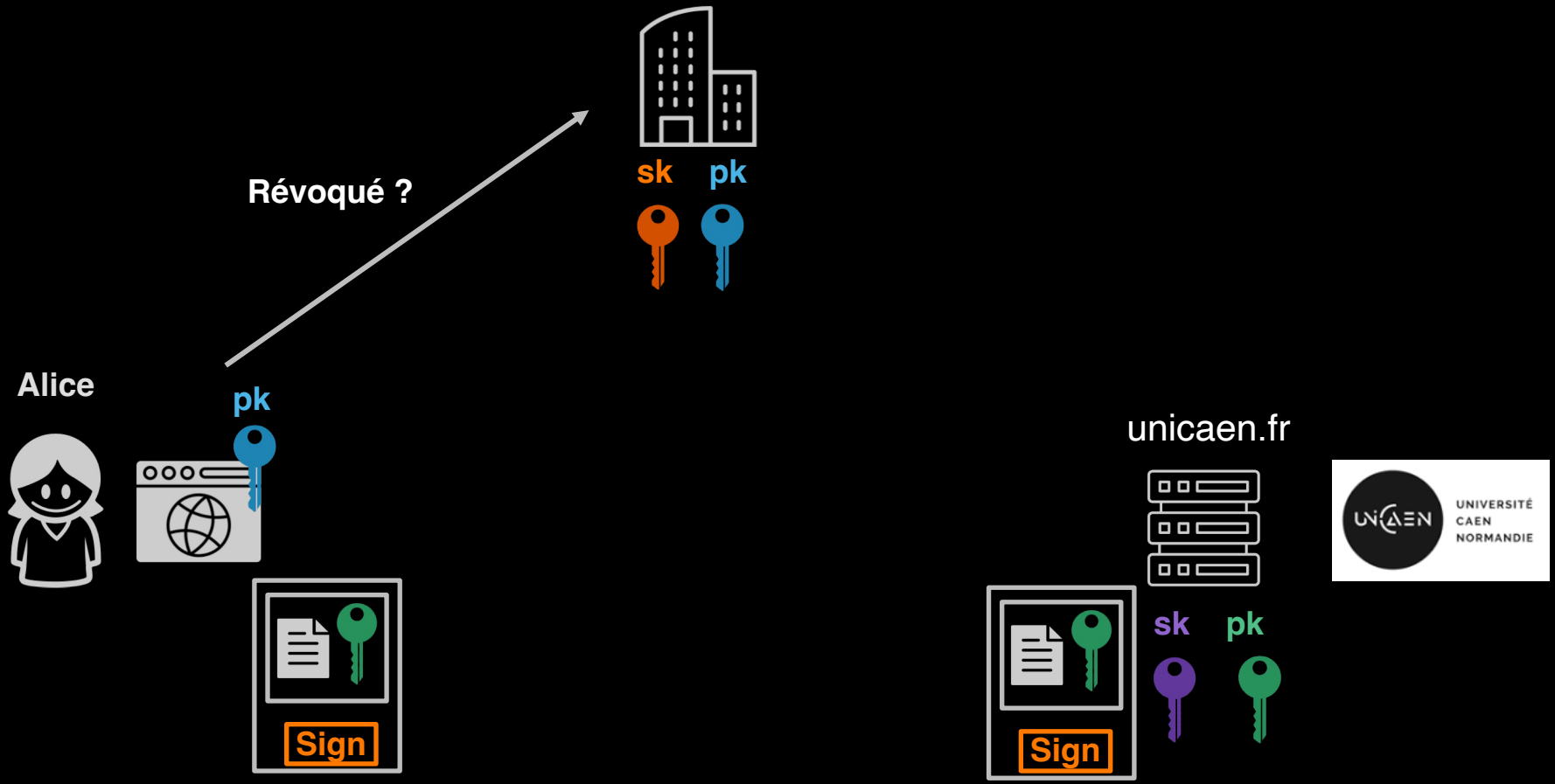
Hello



Certificats numériques



Certificats numériques

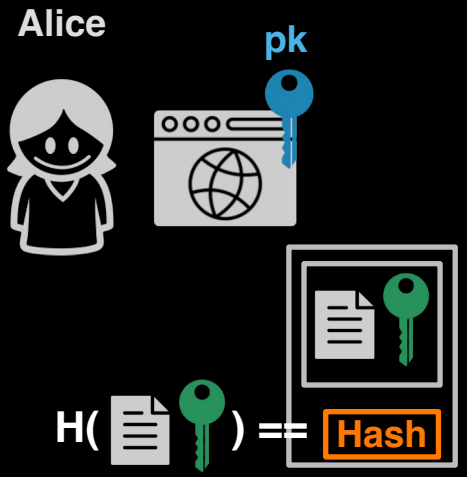


Certificats numériques

Alice

unicaen.fr

Certificats numériques



Certificats numériques

Exemple unicaen.fr

Certificats numériques : unicean.fr

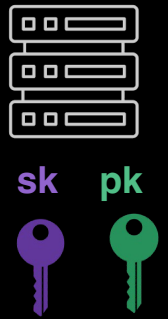
USERTrust RSA

Autorité de certification

Sectigo Limited

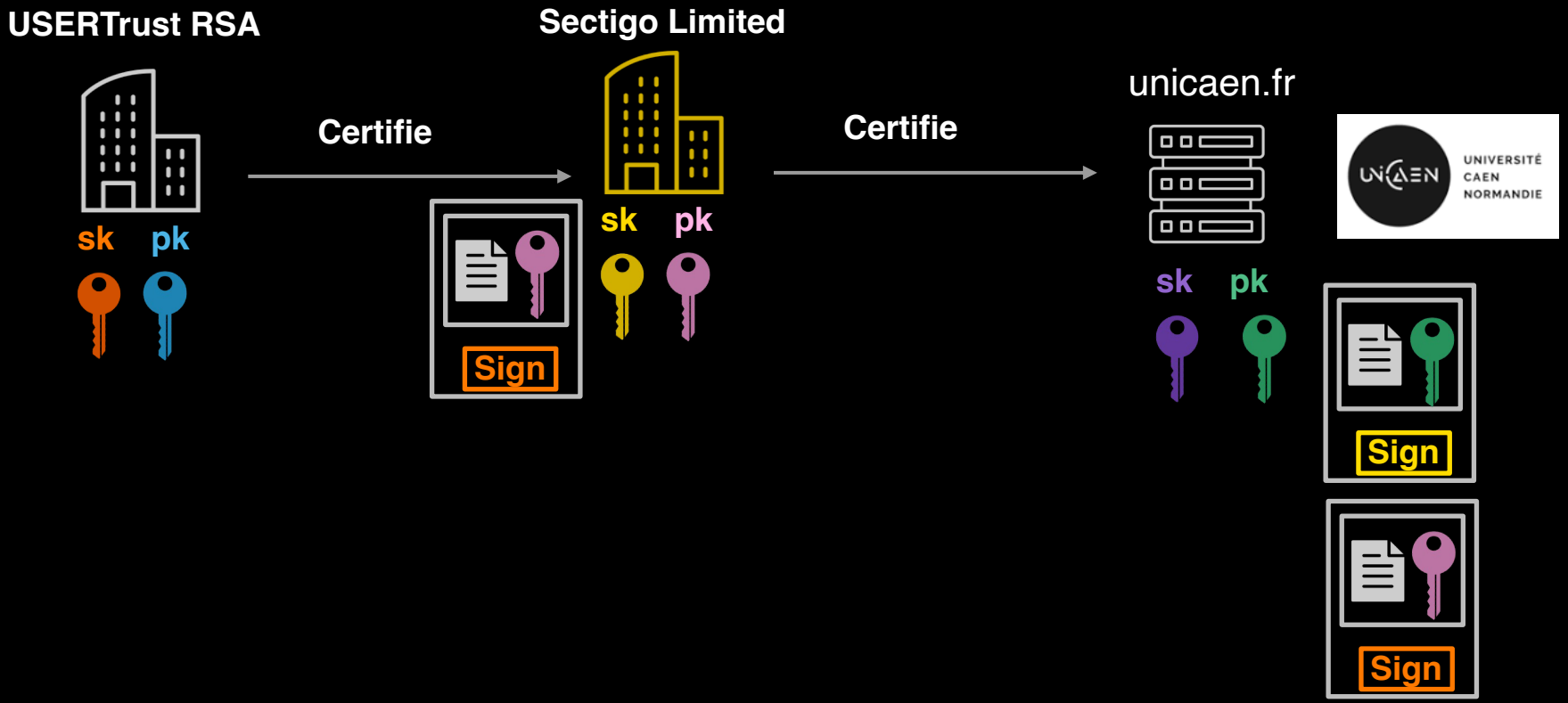
Autorité de certification intermédiaire

unicaen.fr

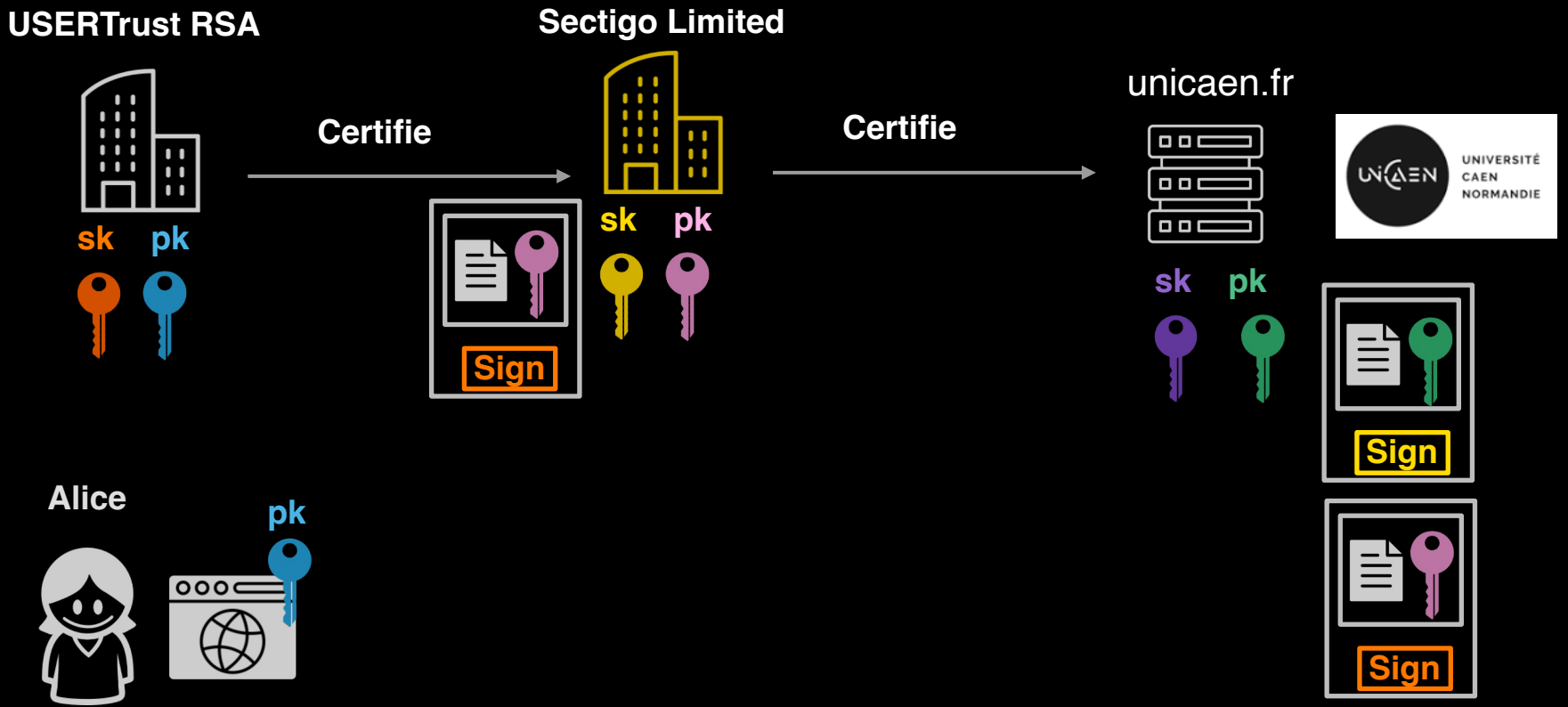


Certificats numériques : unicean.fr

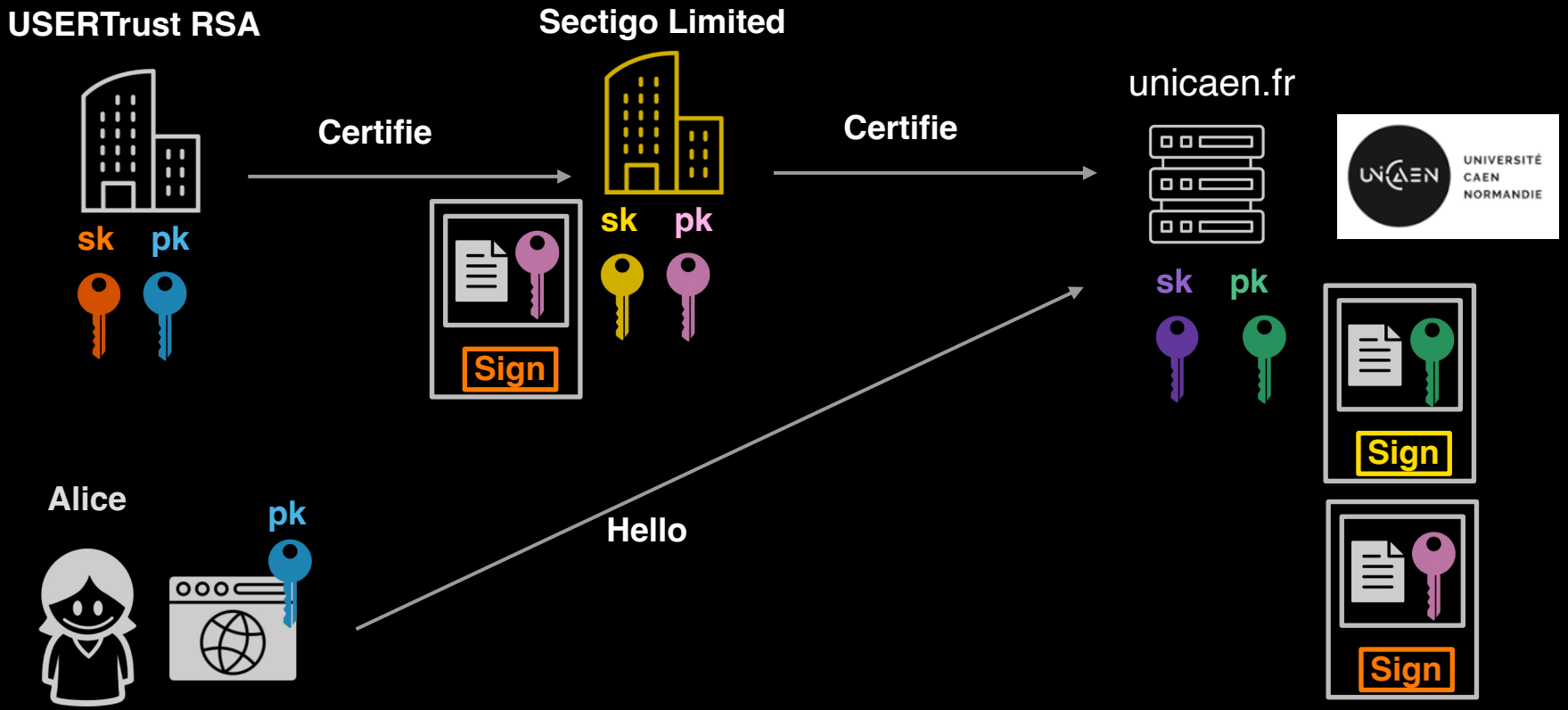
Certificats numériques : unicean.fr



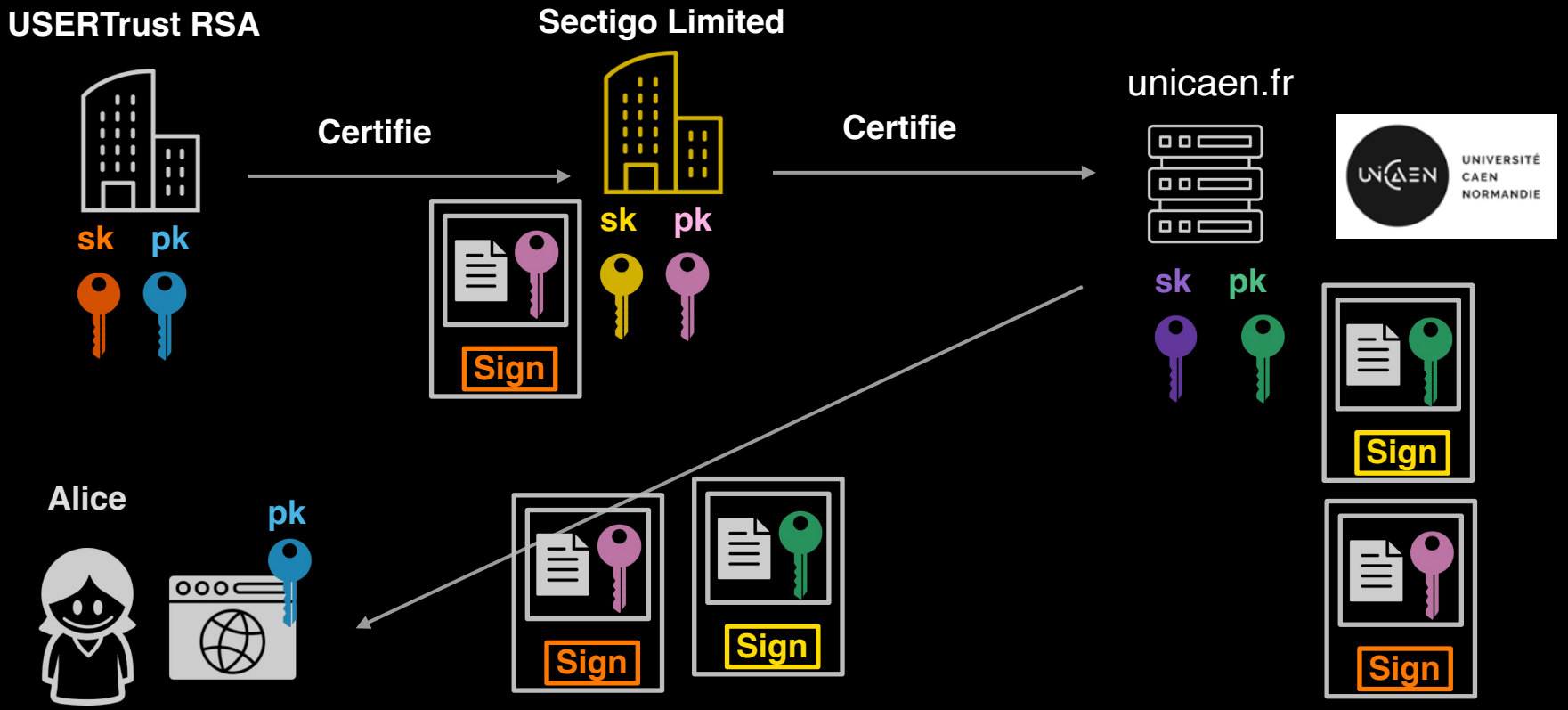
Certificats numériques : unicean.fr



Certificats numériques : unicean.fr



Certificats numériques : unicean.fr



TLS

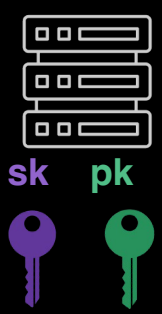
TLS

- Transport Layer Security (TLS)
- Permet de sécuriser les communications
- Le S dans https
- Utilise tout ce qu'on a vu jusqu'à présent

TLS

Alice

unicaen.fr



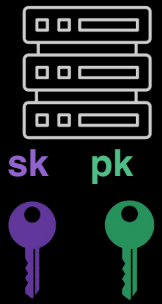
TLS : Etape 1 Hello

Alice

« Hello Client »

{Versions TLS supportées,
nombre aléatoire CR,
Liste des algos crypto supportés}

unicaen.fr



TLS : Etape 2 Hello

Alice

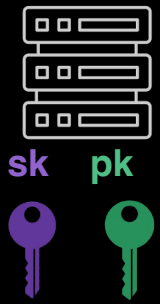
« Hello Client »

« Hello Server »

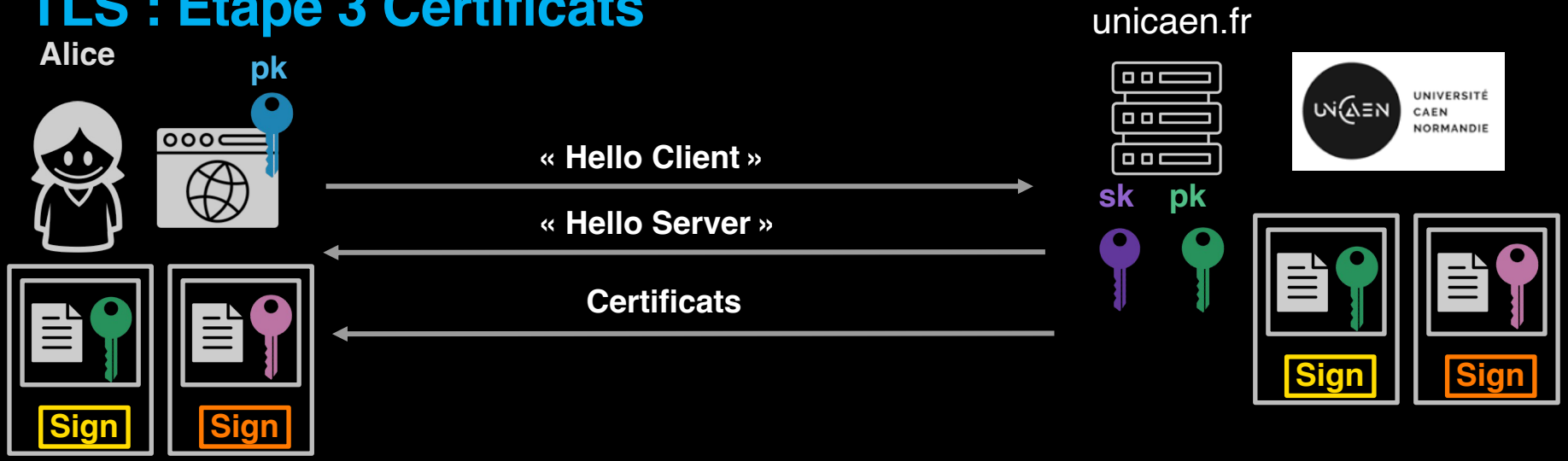
{Version TLS,
nombre aléatoire SR,
liste des algos crypto choisis}

TLS_DHE_RSA_WITH_AES_128_SHA256

unicaen.fr

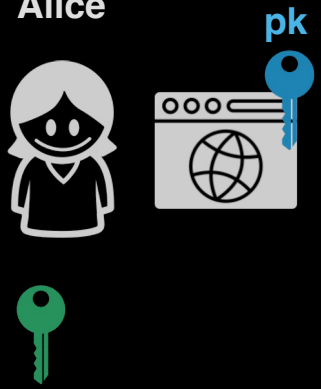


TLS : Etape 3 Certificats

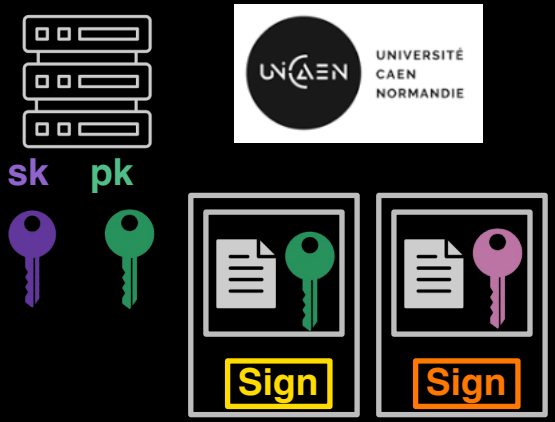


TLS : Etape 4 échange de clés (DH)

Alice



unicaen.fr



« Hello Client »

« Hello Server »

Certificats

Diffie-Hellman

$g, p, g^b \bmod p$

Sign

Choisi p premier et g un générateur. Tire b aléatoire

TLS : Etape 4 échange de clés (DH)

Alice

« Hello Client »

« Hello Server »

Certificats

Diffie-Hellman

$g, p, g^b \bmod p$

Tire a aléatoire,
Calcule $g^a \bmod p$

$g^a \bmod p$

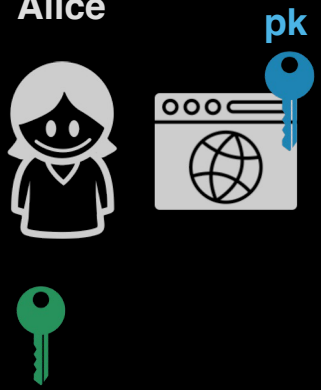
unicaen.fr

sk pk

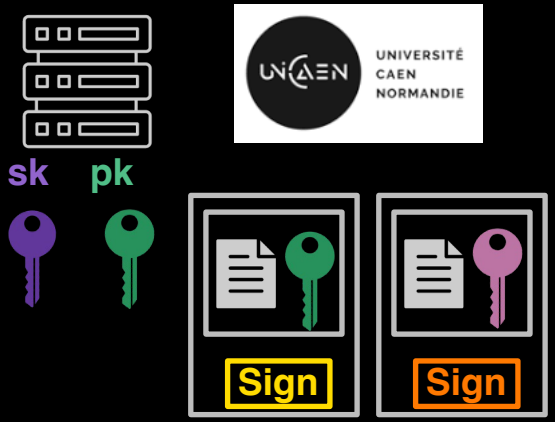
Choisi p premier et g un générateur. Tire b aléatoire, calcule $g^b \bmod p$

TLS : Etape 4 échange de clés (DH)

Alice



unicaen.fr



« Hello Client »

« Hello Server »

Certificats

Diffie-Hellman

$g, p, g^b \bmod p$

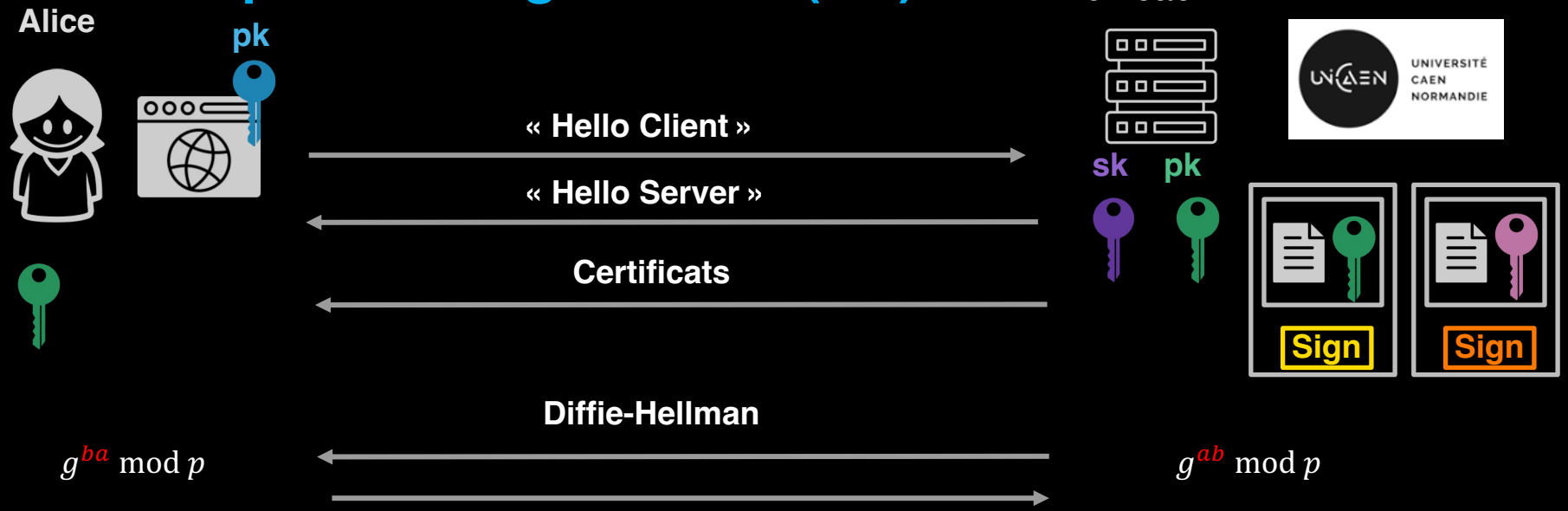
Tire a aléatoire, Calcule $g^a \bmod p$

Calcule $g^{ba} \bmod p$

Choisi p premier et g un générateur. Tire b aléatoire, calcule $g^b \bmod p$

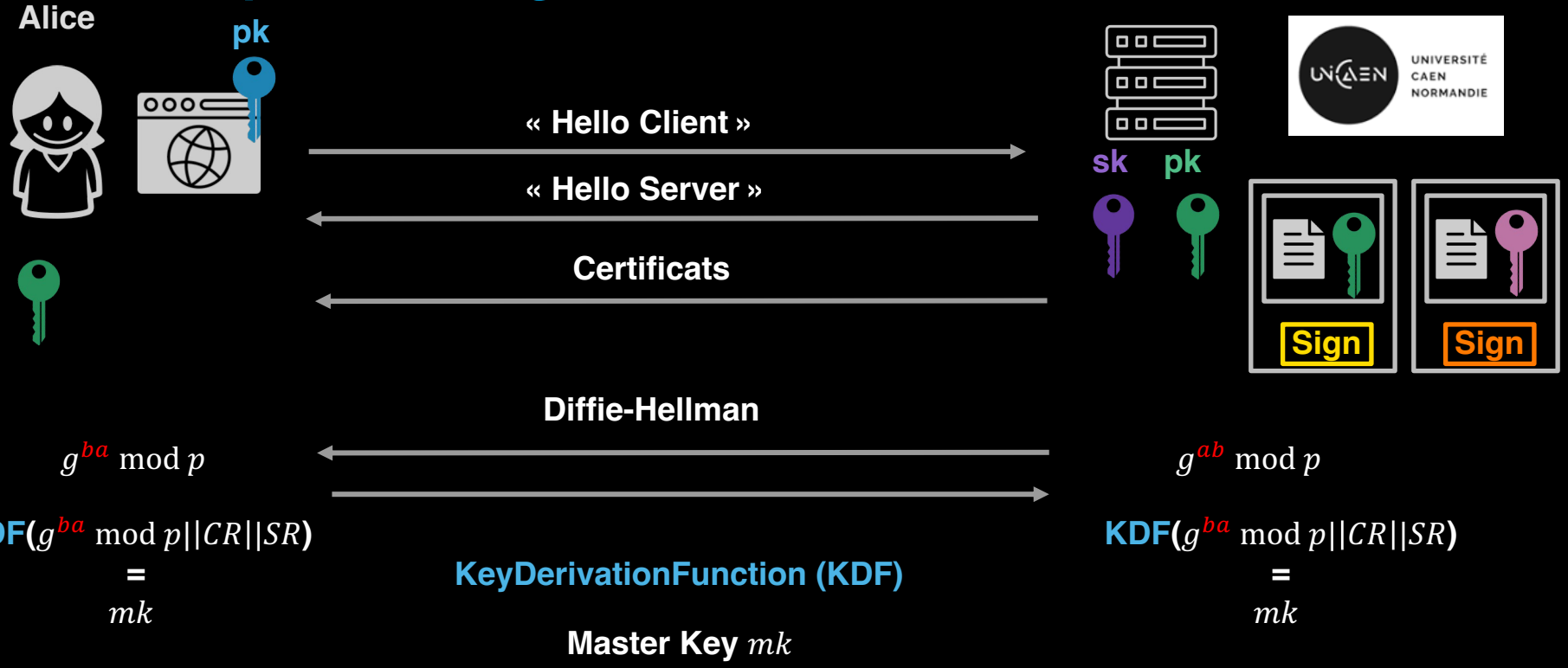
Calcule $g^{ab} \bmod p$

TLS : Etape 4 échange de clés (DH)



$g^{ba} \bmod p$ **pre-master secret**

TLS : Etape 4 échange de clés (DH)



TLS : Etape 5 « finished »

TLS : Etape 5 « finished »

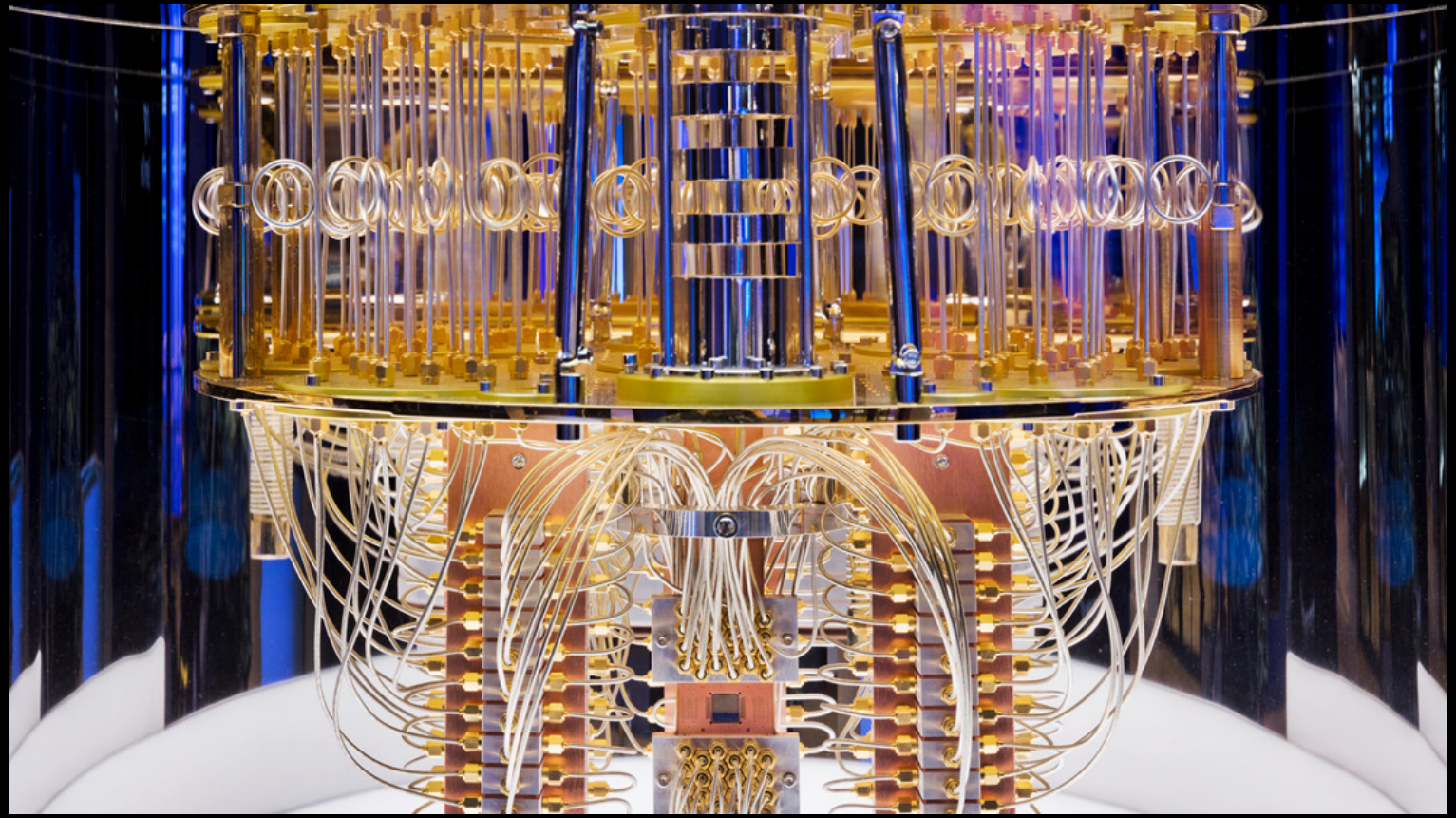
- **Le message « finished » :**
  - Composé du hash de tous les messages échangés entre Alice et le serveur
  - Alice et le serveur calculent le hash chacun de leur côté
  - Chiffré avec la master key
  - Alice et le serveur vérifient qu'ils ont le même hash
    - Si les hashs sont identiques, la connexion est sécurisée

Cryptographie post-quantique

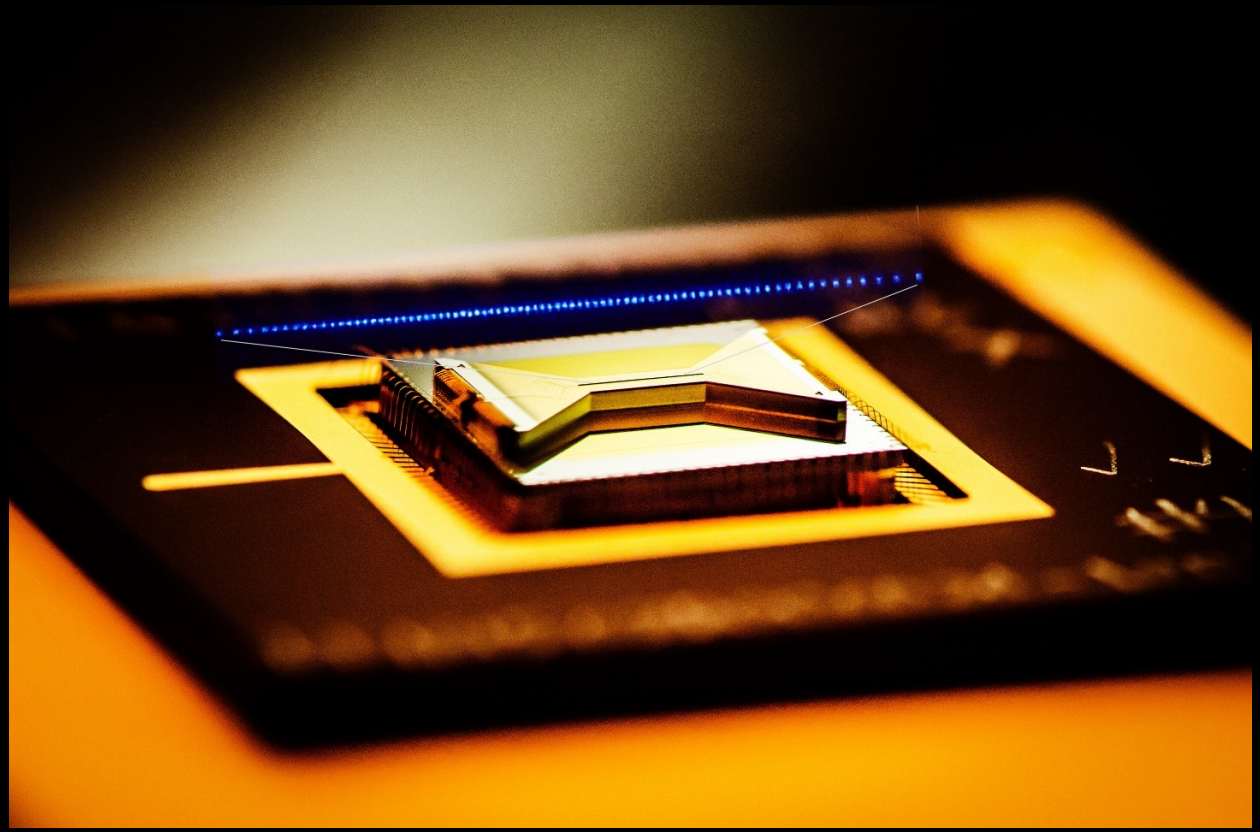
(Slides Loïc Ferreira Orange Innovation)

Ordinateur quantique

Ordinateur quantique



CPU quantique



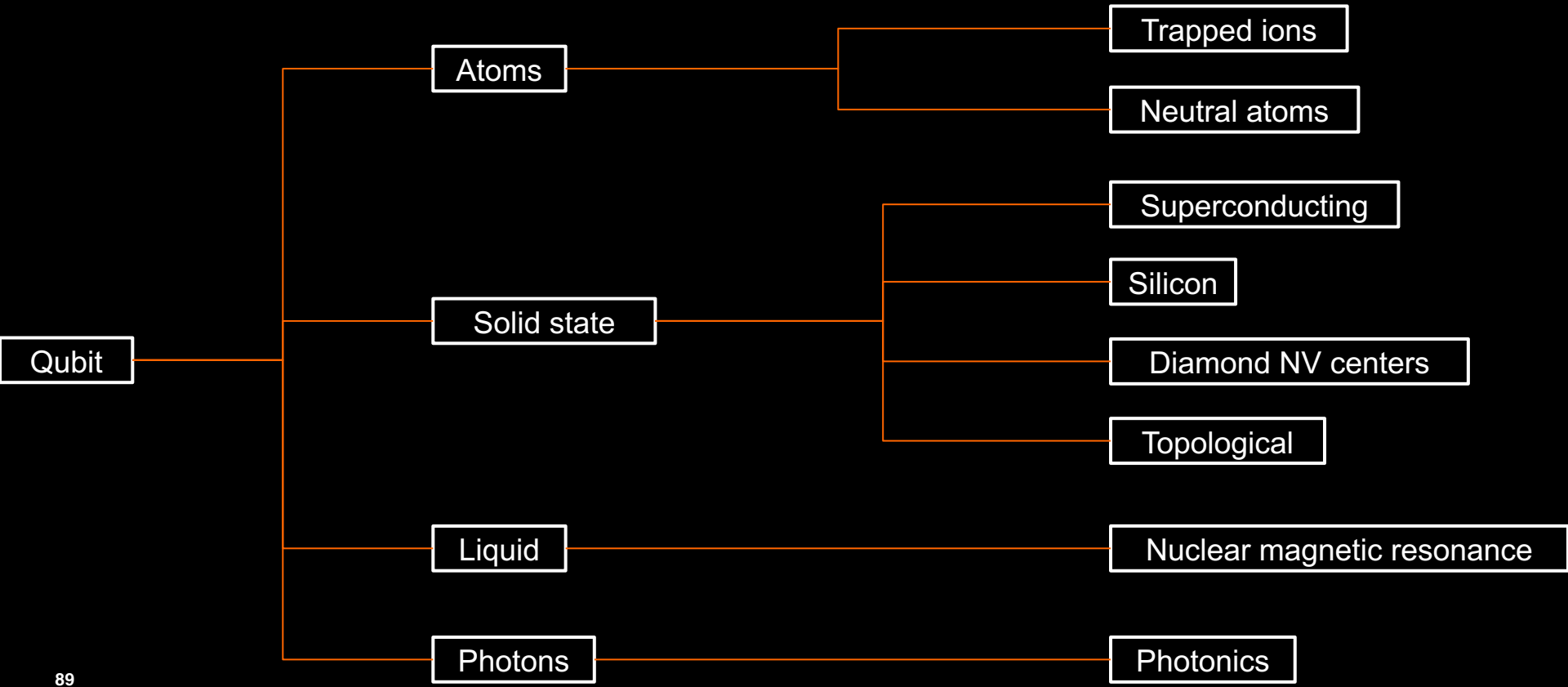
Ordinateur quantique « desktop »

SpinQ (Chine)
Gemini (2 Qubits)

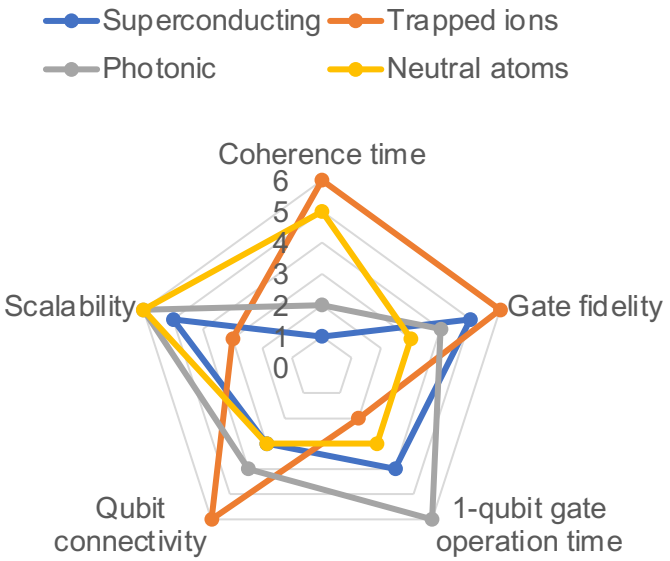
SpinQ (Chine)
Triangulum (3 Qubits)

Quantum Brillance (Australie – Allemagne)
Gen1 (2 Qubits)

Types de Qubits



Propriétés des Qubits



Facteurs déterminants l'efficacité d'un ordinateur quantique:

1. **Le nombre de qubits physiques**
2. **Le nombres de portes logiques qui peuvent êtres appliquées à 1 qubits avant de perdre l'information**
3. **La connectivité de la machine**
4. **Le nombre d'opérations réalisable en parallèles**

Fabricants d'ordinateurs quantique

Photoniques

XANADU

PsiQuantum

ORCA COMPUTING

Superconducteurs

intel

Google

rigetti

IBM

本源量子 ORIGIN QUANTUM

Alibaba Group

QuTech

ALICE & BOB

D-Wave The Quantum Computing Company™

Trapped ions

IONQ

QUANTINUUM

AQT

Topologiques

Microsoft

Atomes neutres

PASQAL

ColdQuanta

QuEra COMPUTING INC.

atom computing

NMR

IQM

Ordinateurs quantiques existants (2023)

Constructeur	Pays	Technologies	Nombres de Qubits
IBM	USA	Superconducting	433
Atom Computing	USA	Neutral atoms	100
Rigetti	USA	Superconducting	79
Google	USA	Superconducting	72
Intel	USA	Superconducting	49
IonQ	USA	Trapped ions	32
Quantinuum	USA	Trapped ions	20
Xanadu	Canada	Photonics	216
D-Wave	Canada	Superconducting (quantum annealing)	5760
University of Science and Technology of China (USTC)	China	Photonics	113
USTC	China	Superconducting	66
Origin Quantum	China	Superconducting	60
SpinQ	China	Superconducting	8
SpinQ	China	Nuclear magnetic resonance (NMR)	6
Pasqal	France	Neutral atoms	100
Oxford Quantum Circuits (OQC)	United Kingdom	Superconducting	8
QuTech	The Netherlands	Superconducting	5
Alpine Quantum Technologies (AQT)	Austria	Trapped ions	20

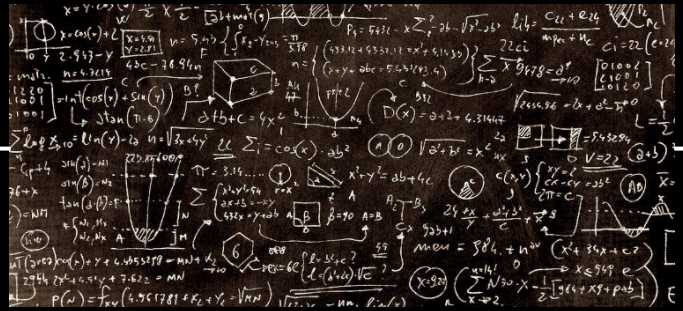
L’ordinateur D-Wave n’est pas générique, Trade off pour un Meilleur passage à l’echelle.

Applications des ordinateurs quantiques

- **Les ordinateurs sont prometteurs pour les industries suivantes :**
  - Machine learning & IA
  - Chimie,
  - Pharmaceutique,
  - Finance,
  - Télécommunication,
  - Energie,
  - Cryptographie (cryptanalyse),
  - …

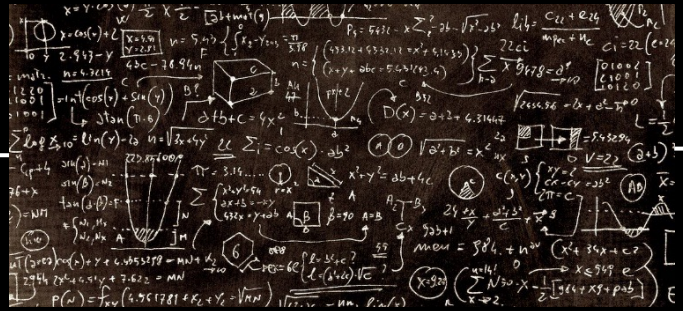
Problèmes mathématiques

2 familles de problèmes utilisées en cryptographie à clé publique classique



Problèmes mathématiques

2 familles de problèmes utilisées en cryptographie à clé publique classique

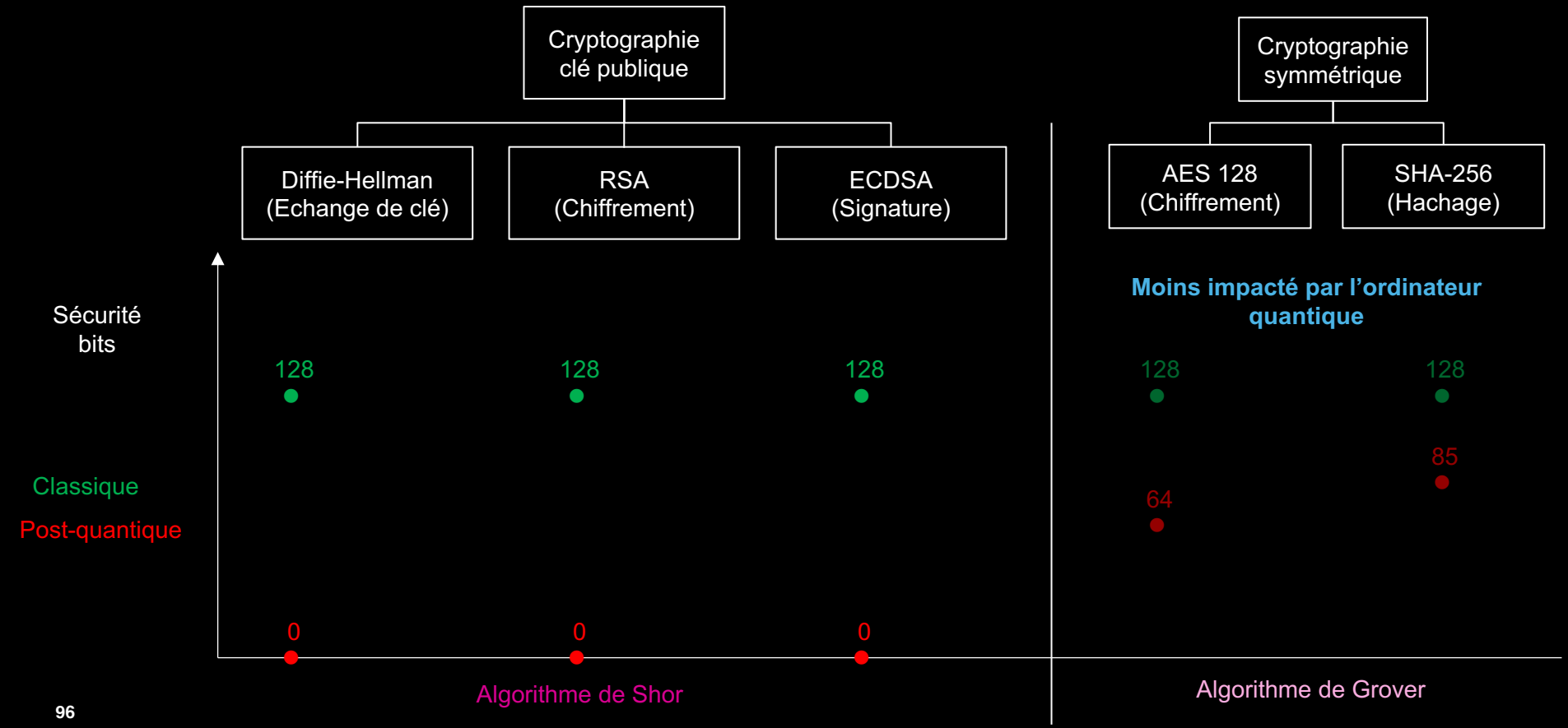


Factorisation
Retrouver (p, q)
connaissant $n = p \cdot q$
Sécurise R.S.A
Sûr pour n **3072 bits**

Logarithme discret
Retrouver x
connaissant (g, g^x)
Sécurise Diffie-Hellman
Sûr pour n **3072 bits**

« Facile » à résoudre avec l'algorithme de Shor pour ordinateur quantique

Impact cryptographie à clé publique



Impact

Web sécurisé

Messageries chiffrées

Mails chiffrés

Viso-conférence

…

Online payments

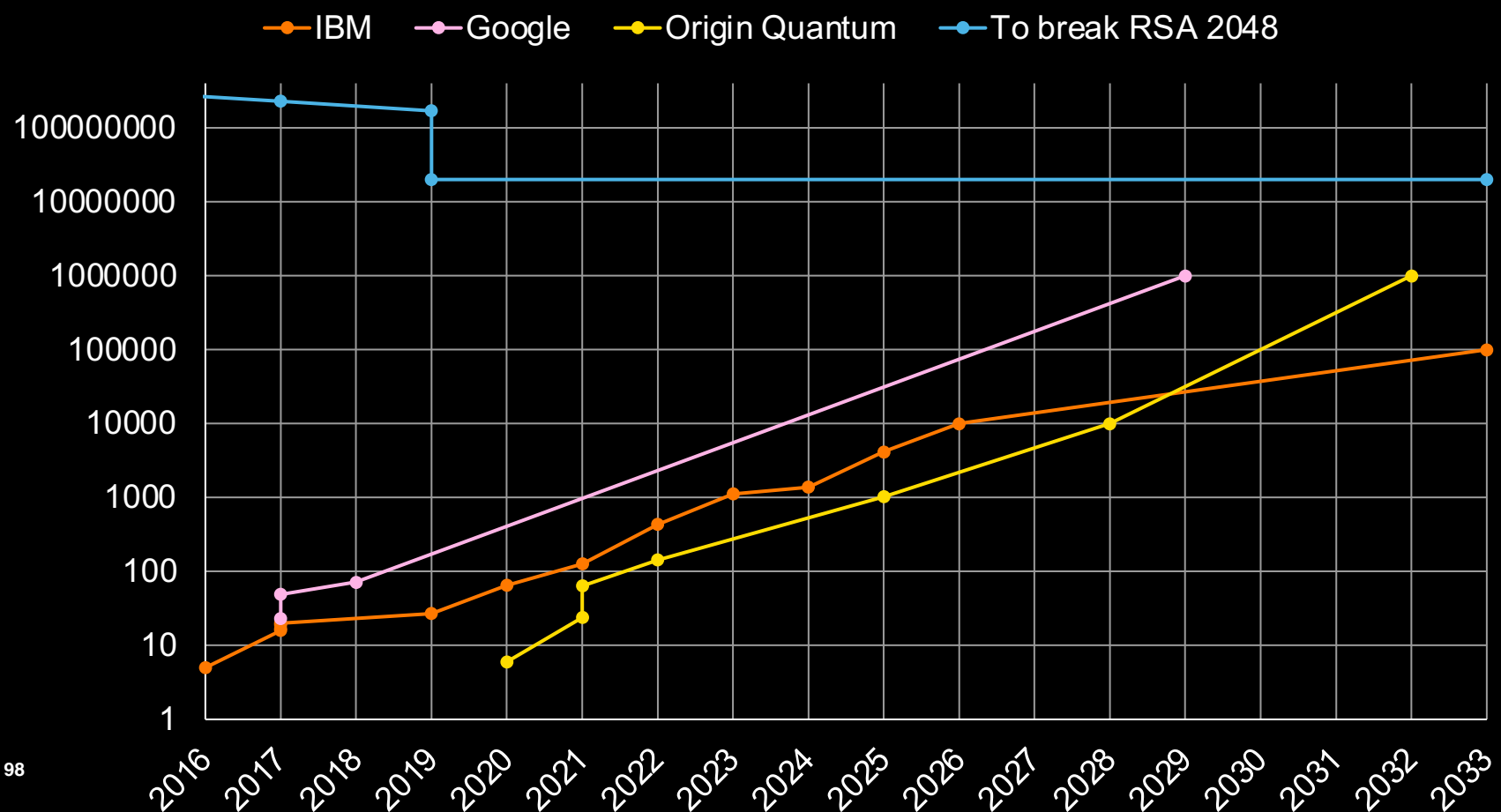
PKI

VPN

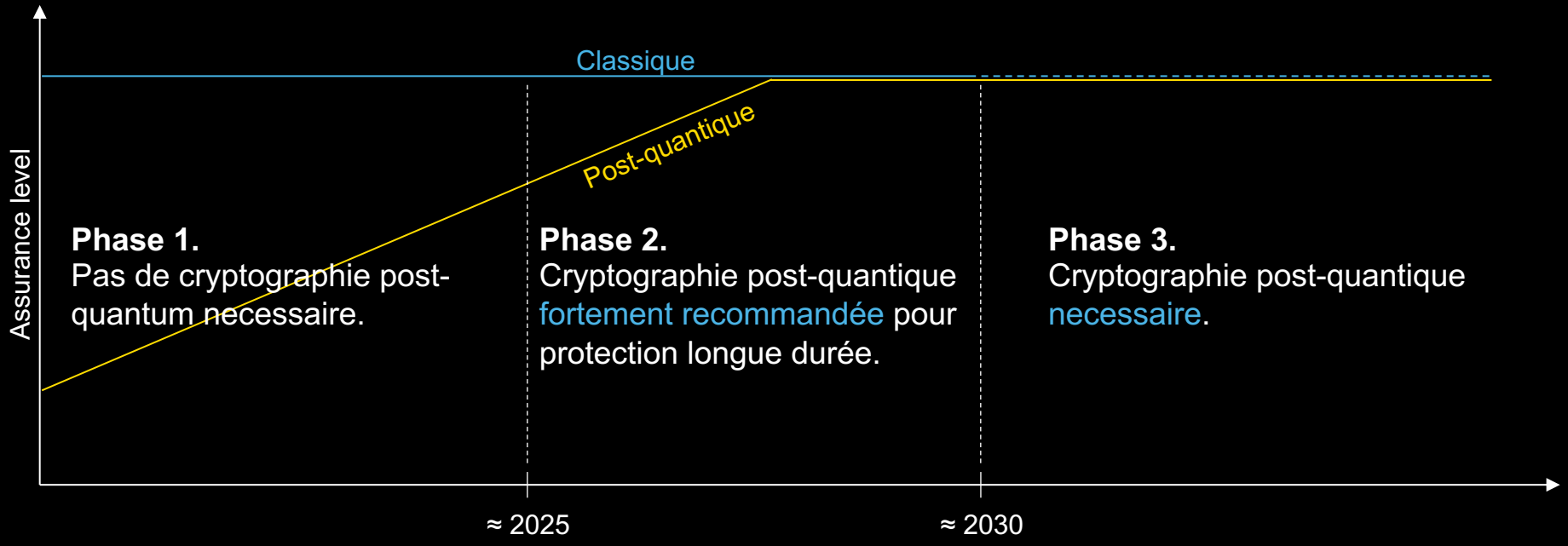
Cryptomonnaies

…

Projection puissance des ordinateurs quantiques



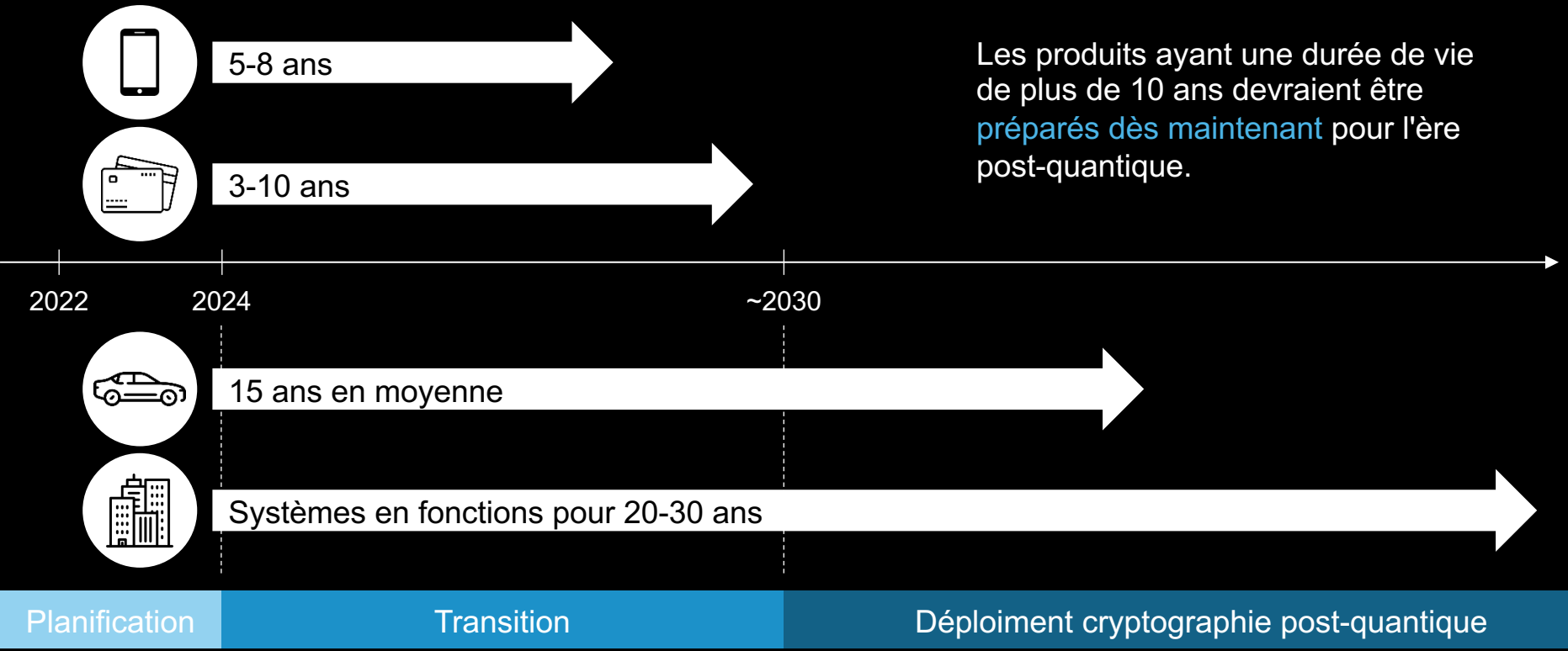
Recommandations ANSSI



L'ANSSI recommande d'introduire la PQC le plus tôt possible pour les produits de sécurité visant à offrir une protection durable de l'information (au delà de 2030) ou qui seront potentiellement utilisés après 2030 sans mises à jour.

https://www.ssi.gouv.fr/en/publication/anssi-views-on-the-post-quantum-cryptography-transition/

Recommandations ANSSI



Les produits ayant une durée de vie de plus de 10 ans devraient être préparés dès maintenant pour l'ère post-quantique.

Takeaways

- **Cryptographie classique à risque**
  - Cryptographie à clé publique fortement impactée
  - Cryptographie à clé secrète faiblement impactée

- **Confidentialité (chiffrement)**
  - Données actuelles déjà à risque
  - L'impact dépend de la durée de vie et sensitivité
    - Les données devant restées sûres longtemps doivent êtres chiffrées avec des algorithmes post-quantique maintenant.

- **Authentification**
  - Pas à risque pour l'instant, durée de vie limitée des certificats