Sécurisation des communications

Enseignant : M. COLOMB

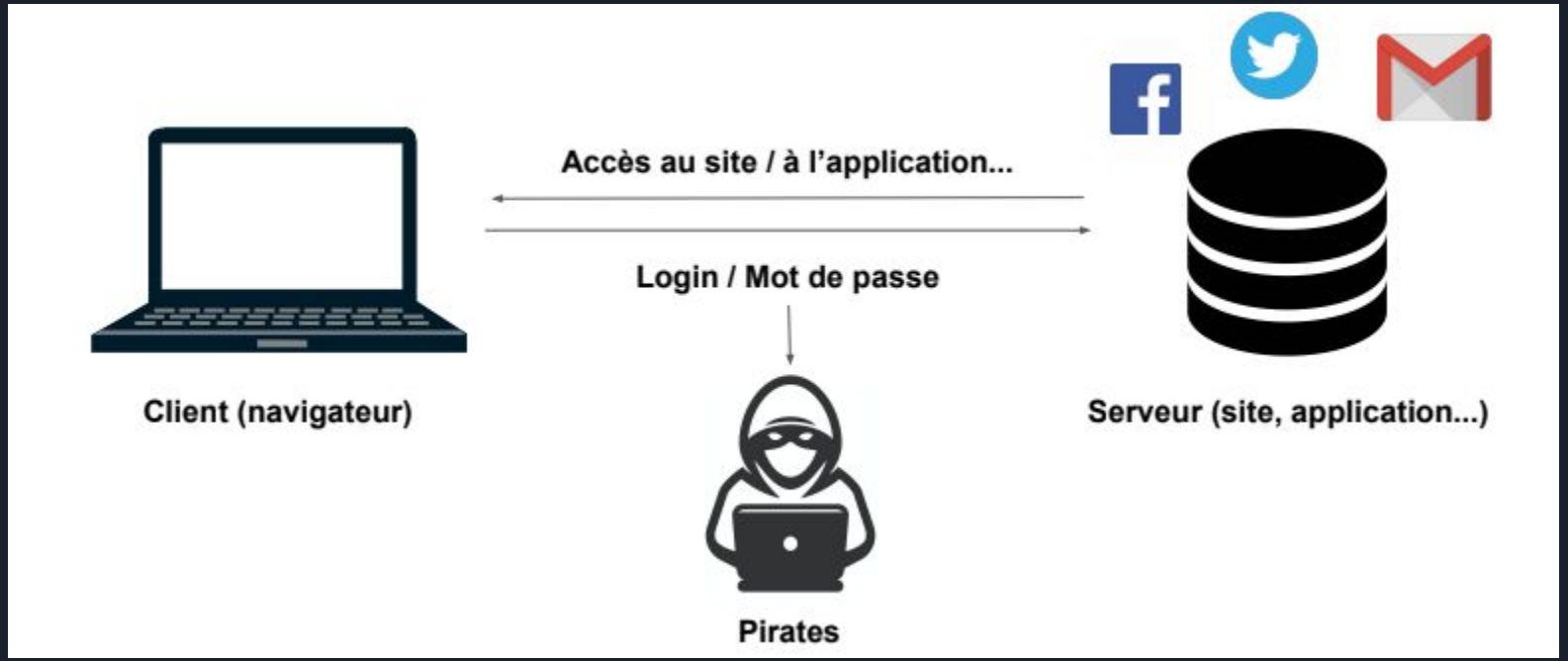
Introduction

Afin d'éviter que les échanges d'informations entre 2 machines soient visibles par autrui, il est nécessaire de chiffrer l'information.

Historiquement, les premiers messages chiffrés étaient généralement destinées à des fins militaires pour chiffrer les messages contenant les plans d'attaque.

Aujourd'hui il est essentiel de chiffrer les informations transmises sur le Web pour ne pas que d'autres personnes malveillantes puisse intercepter ces échanges et récupérer des informations bancaires par exemple.

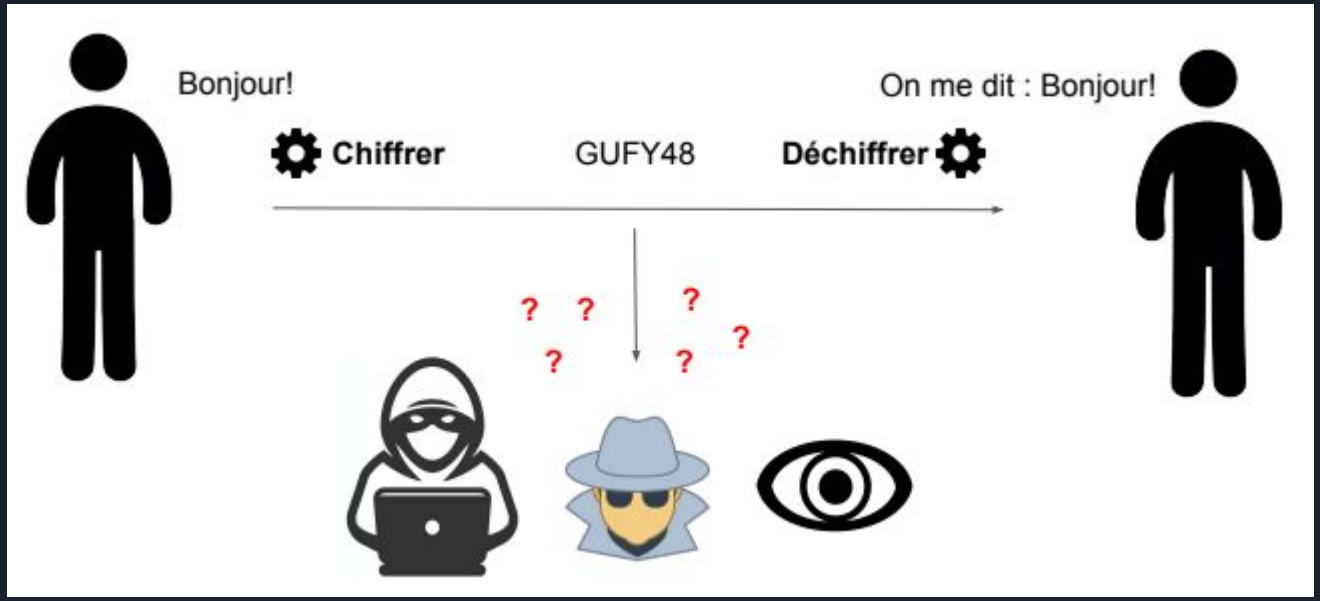
Introduction



Introduction

Pour éviter cette situation, on va chiffrer nos messages.

Introduction



Chiffrement de César

Principe :

1. On choisit un nombre entre 1 et 26, appellé clé. On en aura besoin pour chiffrer et déchiffrer un message.
2. Pour chiffrer, on prend chaque lettre du texte “en clair” et on la remplace par la lettre situé à la distance du nombre choisi pour la clé, en allant vers la droite, en respectant l’ordre alphabétique.
3. Si jamais, en chiffrant, on dépasse la fin de l'alphabet, on continue en revenant au début. (Une clé “3” sur un “Y” donne donc “B”).
4. Pour déchiffrer, on prend chaque lettre du texte “chiffré” et on la remplace par la lettre situé à la distance du nombre choisi pour la clé, en allant vers la gauche, en respectant l’ordre alphabétique.
5. Si jamais, en déchiffrant, on dépasse le début de l'alphabet, on continue en allant à la fin. (Une clé “3” sur un “B” donne donc “Y”).

Chiffrement de César

Exemple : On chiffre le message “BONJOUR” avec une clé = 2 :

B	O	N	J	O	U	R
D	Q	P	L	Q	W	U

On décale chaque lettre de 2 cran vers la droite dans l’alphabet et on obtient notre message chiffré par le chiffrement de César avec une clé = 2.

Pour déchiffrer le message il suffit que le destinataire connaisse la clé de chiffrement pour pouvoir déchiffrer le message.

Chiffrement de César

Cette méthode de chiffrement est très basique et est très facilement cassable. Si je ne connais pas la clé de chiffrement il me suffit d'essayer les 26 clés possibles et de regarder les résultats pour trouver le bon.

Message chiffré :

Qiwweki wigvix

Clé	Message déchiffré
1	Phvvdjh vhfuhw
2	Oguucig ugetgv
3	Nfttbhf tfdsfu
4	Message secret
5	Ldrrzfd rdbqds
…	…

Substitution monoalphabétique

La substitution monoalphabétique est un autre exemple de chiffrement. Une lettre est codée par une autre lettre (mais pas nécessairement par décalage, comme avec le code de César...)

<u>Exemple :</u>

BONJOUR -> VSDHSTZ

Ce code peut être facilement cassable par un analyse fréquentielle de l'apparition des lettres.

A	N
B	V
C	B
D	C
E	X
F	W
G	M

H	L
I	K
J	H
K	J
L	G
M	F
N	D

O	S
P	Q
Q	A
R	Z
S	E
T	R
U	T

V	Y
W	U
X	I
Y	O
Z	P

Sécurisation des communications

Toutes la machines d'un même réseau peuvent observer les communications au sein de ce réseaux. Il est donc important de vérifier que les données que l'on envoie soient chiffrées. Les Wifi publics sont des réseaux à risque pour cette raison.

Sur le Web on peut vérifier simplement que la communication entre notre machine et le serveur Web est chiffrée ou non. Il suffit de regarder le protocole utilisé.

Sécurisation des communications

HTTP : Hypertext Transfer Protocol

Non sécurisé, les données transmises sur le réseau sont en “clair” (non chiffrées).

URL commençant par http://…

HTTPS : Hypertext Transfer Protocol Secure

Sécurisé, les données transmises sur le réseau sont chiffrées, via un autre protocole nommé TLS.

TSL : Transport Layer Security (assure le chiffrement des données)

URL commençant par https://...

Algorithmes symétriques

Un algorithme de chiffrement symétrique est un algorithme utilisant un élément appelé “clé” afin de chiffrer et déchiffrer des messages. La notion de symétrie vient du fait qu’on utilise la même clé pour chiffrer ou déchiffrer le message.
Nous avons vu le code César (où la clé correspond à la valeur du décalage) qui peut être considéré comme un algorithme de chiffrement symétrique.

De nos jours, des algorithmes bien plus performants (et difficilement cassables) ont été mis au point:

- Chiffrement de Fernet
- DES (Data Encryption Standard)
- 3DES (Triple DES = DES appliqué 3 fois)
- AES (Advanced Encryption Standard)
- Etc...

Il reposent tous sur la même logique d’utilisation : Génération d’une clé et deux algorithmes utilisant cette même clé, pour chiffrer et déchiffrer des messages.

Chiffrement symétrique

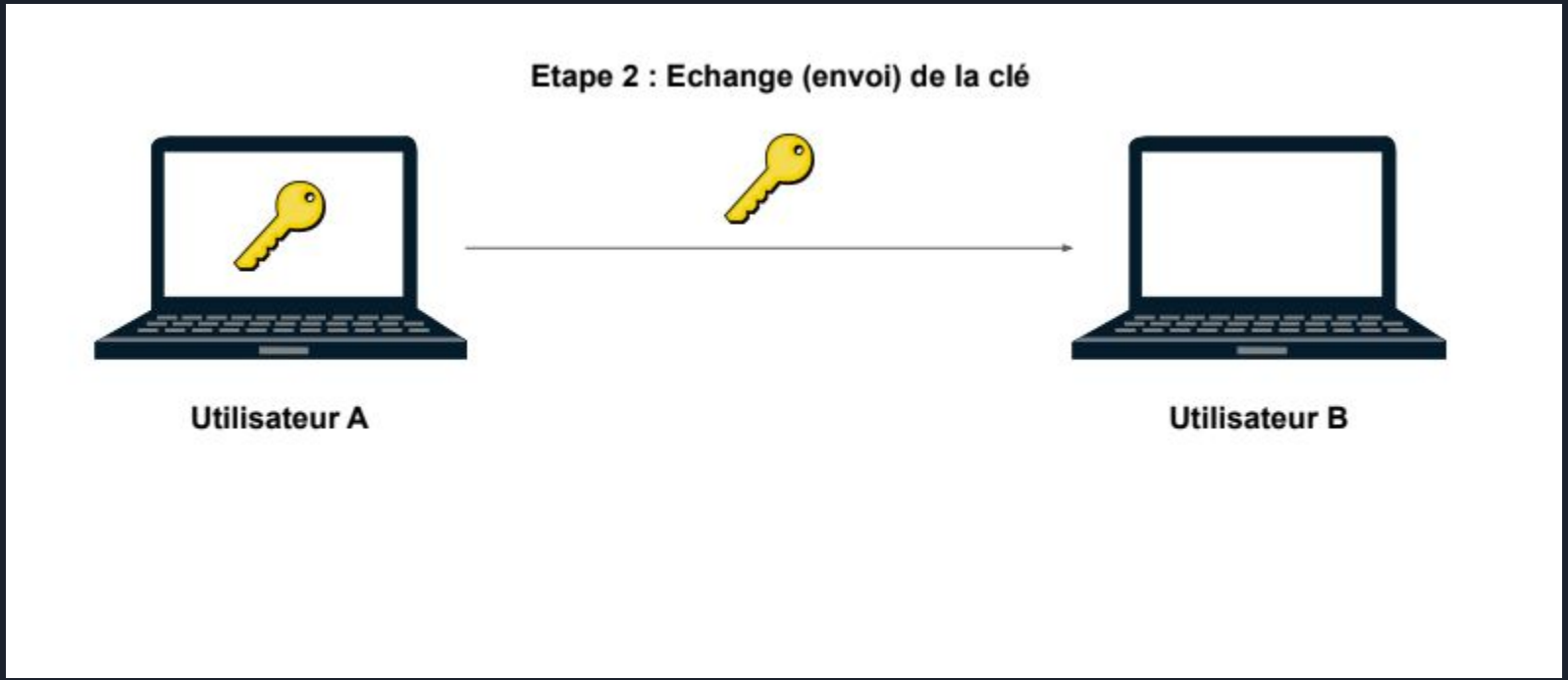
Etape 1 : Génération de la clé de chiffrement symétrique

Utilisateur A

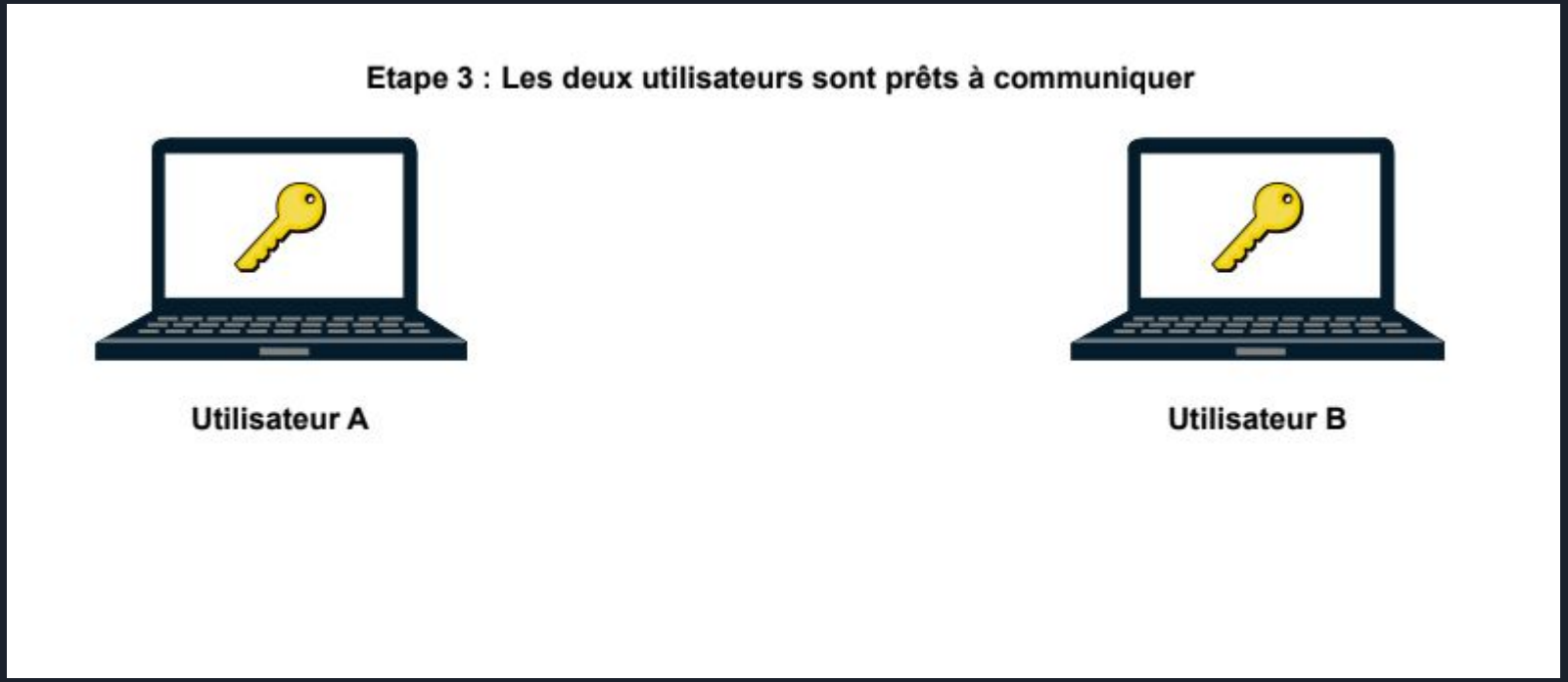
Génère la clé

Utilisateur B

Chiffrement symétrique



Chiffrement symétrique



Chiffrement symétrique

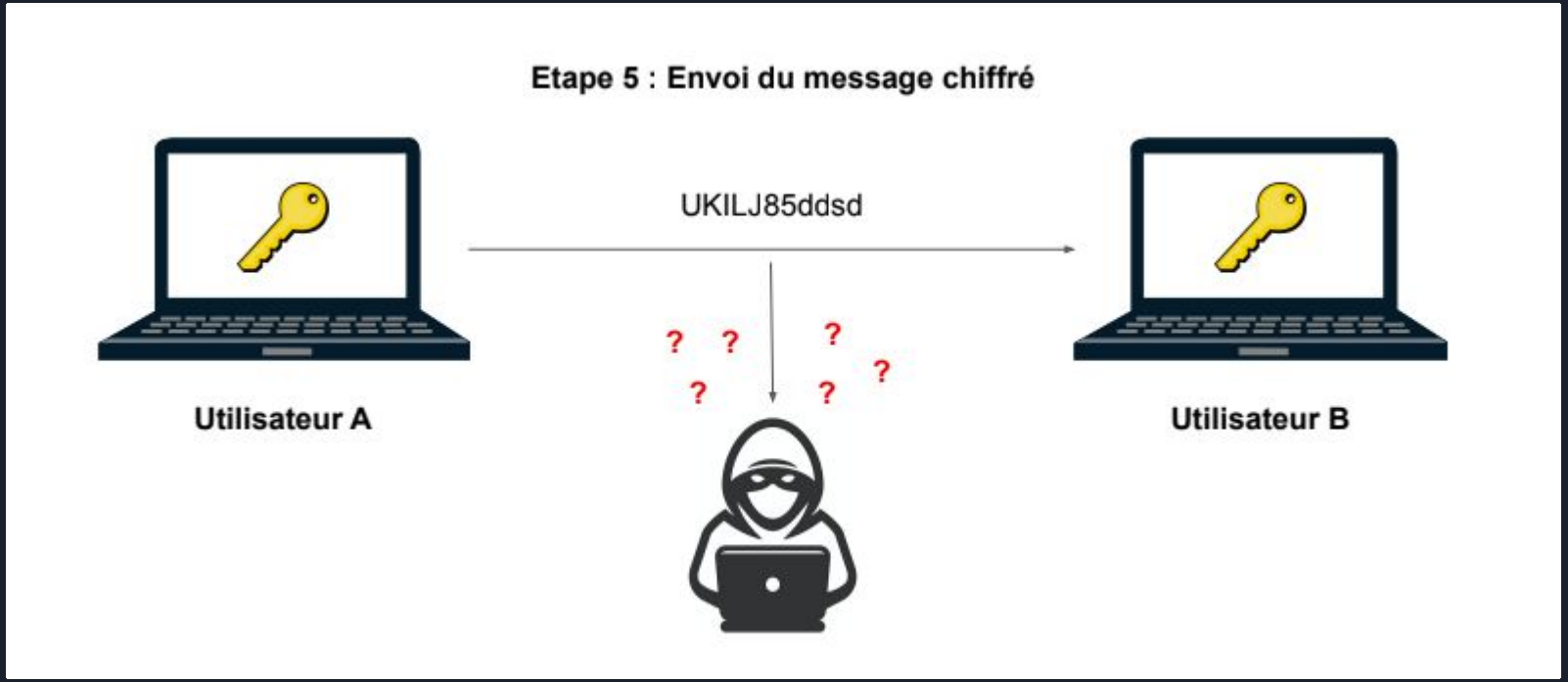
Etape 4 : Chiffrage avec la clé

Utilisateur A

Chiffre le message avec la clé “Bonjour” : “UKILJ85ddsd”

Utilisateur B

Chiffrement symétrique



Chiffrement symétrique

Etape 6 : Déchiffrage avec la clé

Utilisateur A

Utilisateur B

Déchiffre le message avec la clé “UKILJ85ddsd” : “Bonjour”

Chiffrement symétrique

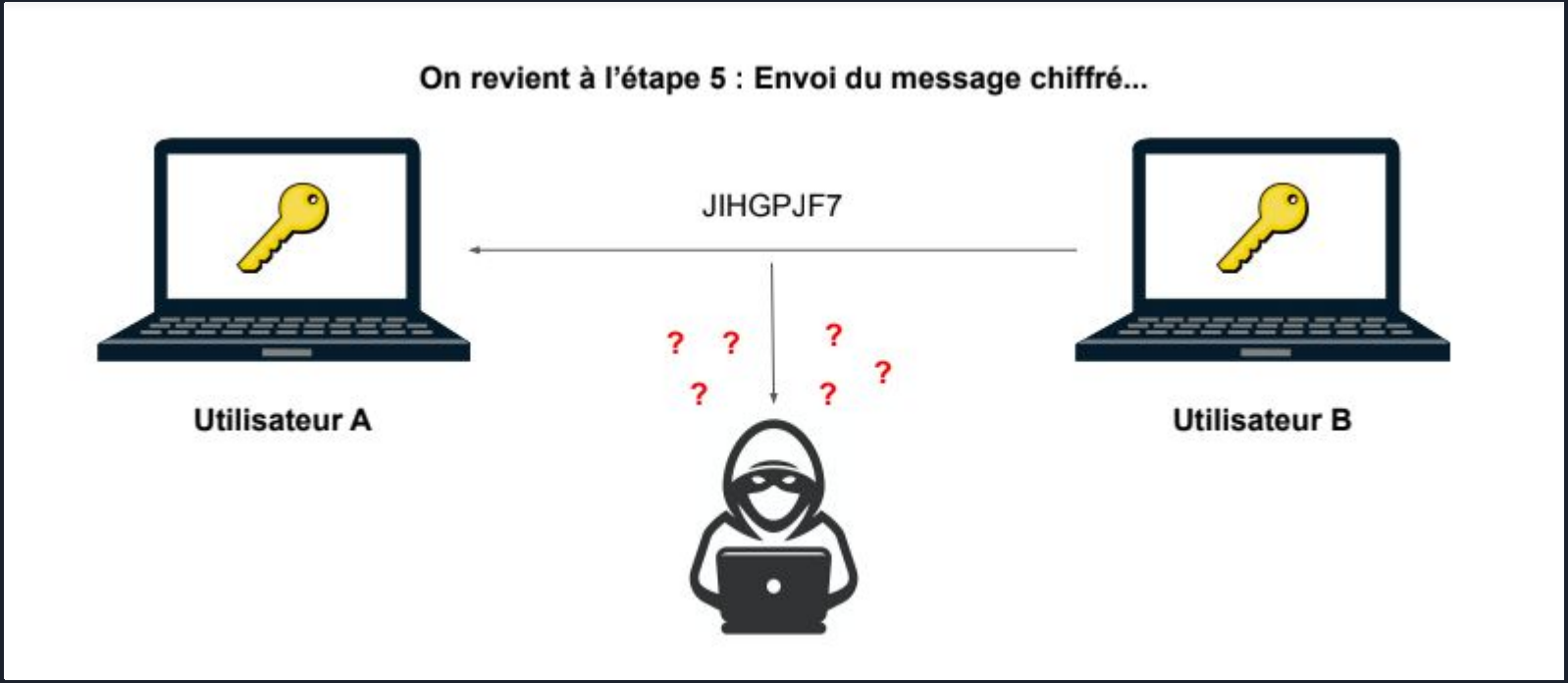
On revient à l'étape 4 : Chiffrage avec la clé...

Utilisateur A

Utilisateur B

Chiffre le message avec la clé "Sécurisé!" : "JIHGPJF7"

Chiffrement symétrique



Chiffrement symétrique

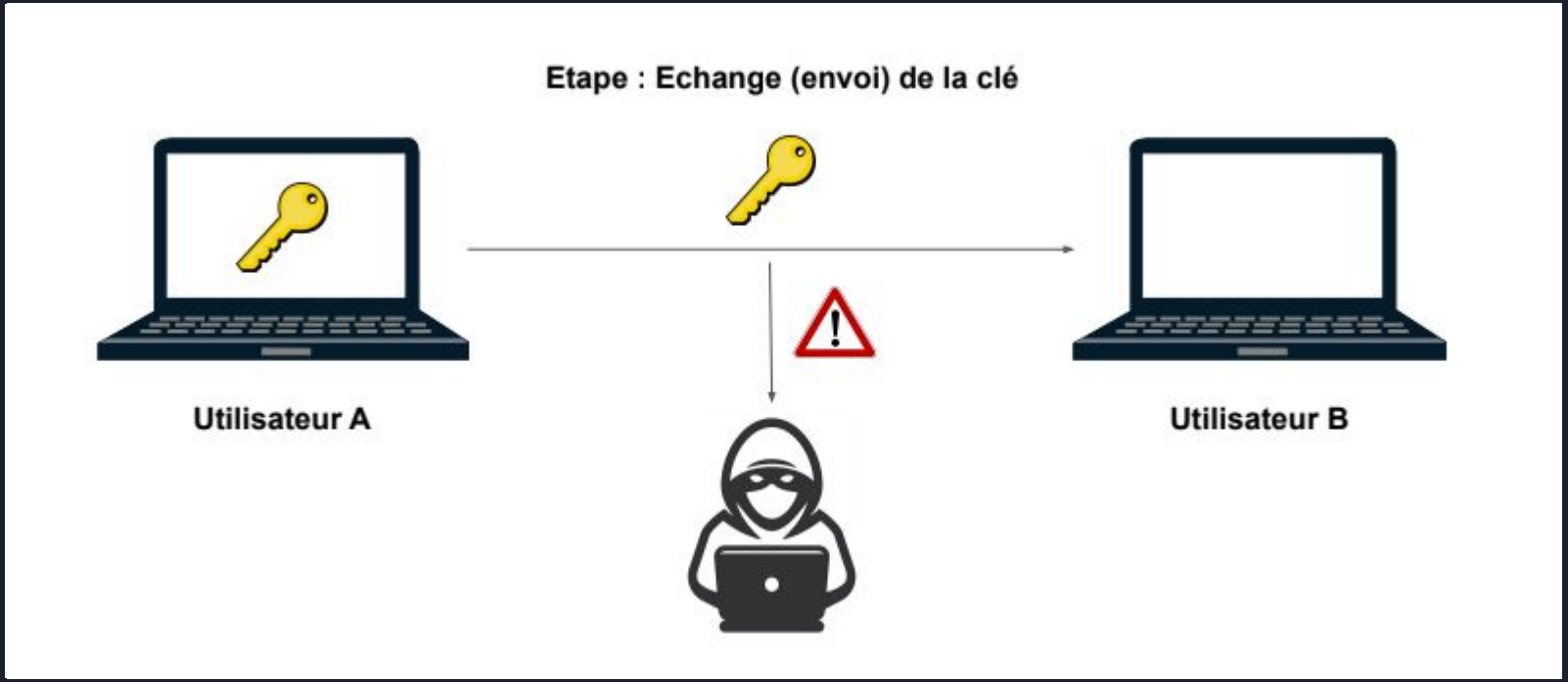
On revient à l'étape 6 : Déchiffrage avec la clé

Utilisateur A

Déchiffre le message avec la clé "JIHGPJF7" : "Sécurisé!"

Utilisateur B

Chiffrement symétrique : Faille !



Chiffrement symétrique : Faille !

Chiffrement symétrique : Faille !

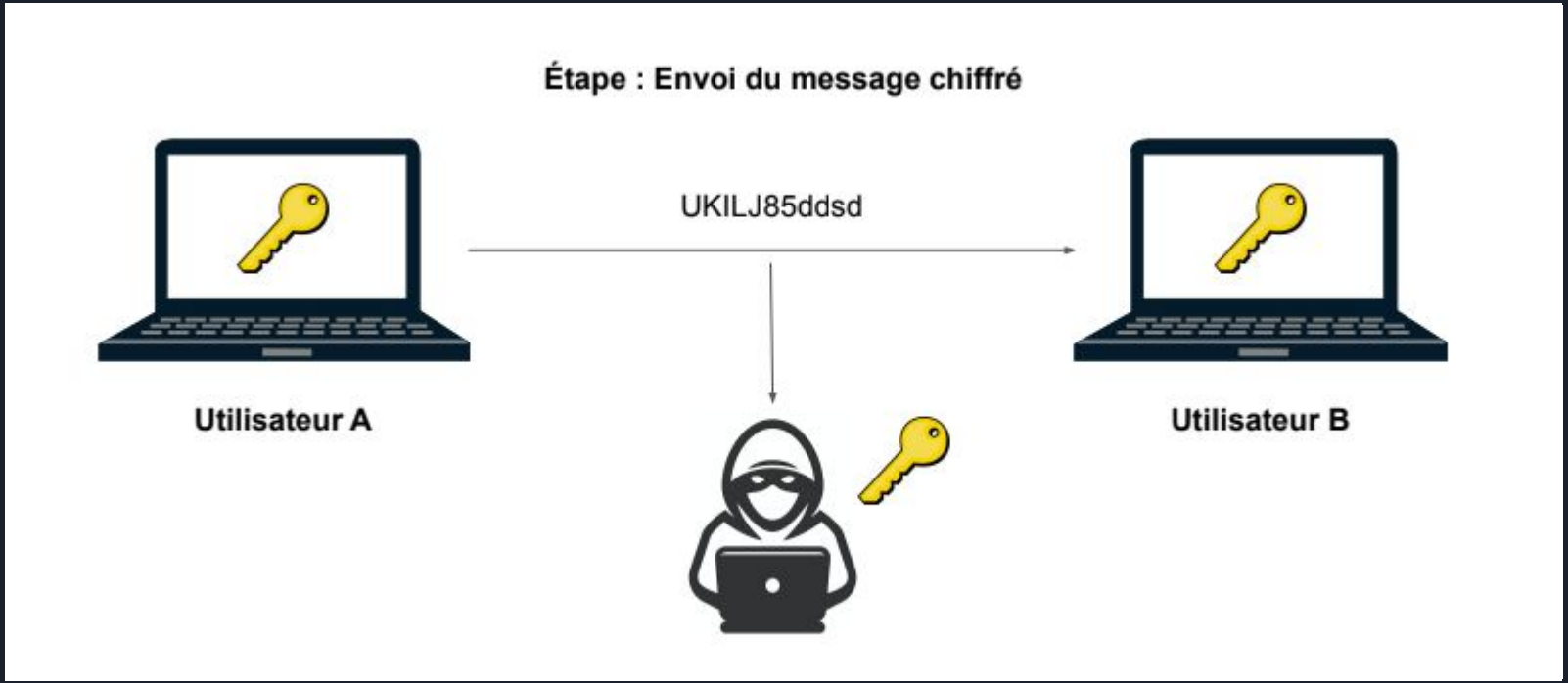
Étape : Chiffrage avec la clé

Utilisateur A

Chiffre le message avec la clé
“Bonjour” : “UKILJ85ddsd”

Utilisateur B

Chiffrement symétrique : Faille !



Chiffrement symétrique : Faille !

Étape : Déchiffrage avec la clé

Utilisateur A

Déchiffre le message avec la clé
“UKILJ85ddsd” : “Bonjour”

Utilisateur B

Déchiffre le message avec la clé
“UKILJ85ddsd” : “Bonjour”

Chiffrement asymétrique

Pour corriger cette faille, une nouvelle méthode de chiffrement est mise en place : le chiffrement asymétrique. Un algorithme de chiffrement asymétrique est un algorithme utilisant deux “clés” afin de chiffrer et déchiffrer des messages.

La notion d’asymétrie vient du fait qu’on utilise :

- Une clé dite “publique” permettant de chiffrer des messages, mais pas de les déchiffrer.
- Une clé dite “privée” permettant de déchiffrer des messages qui ont été chiffré avec la clé publique.

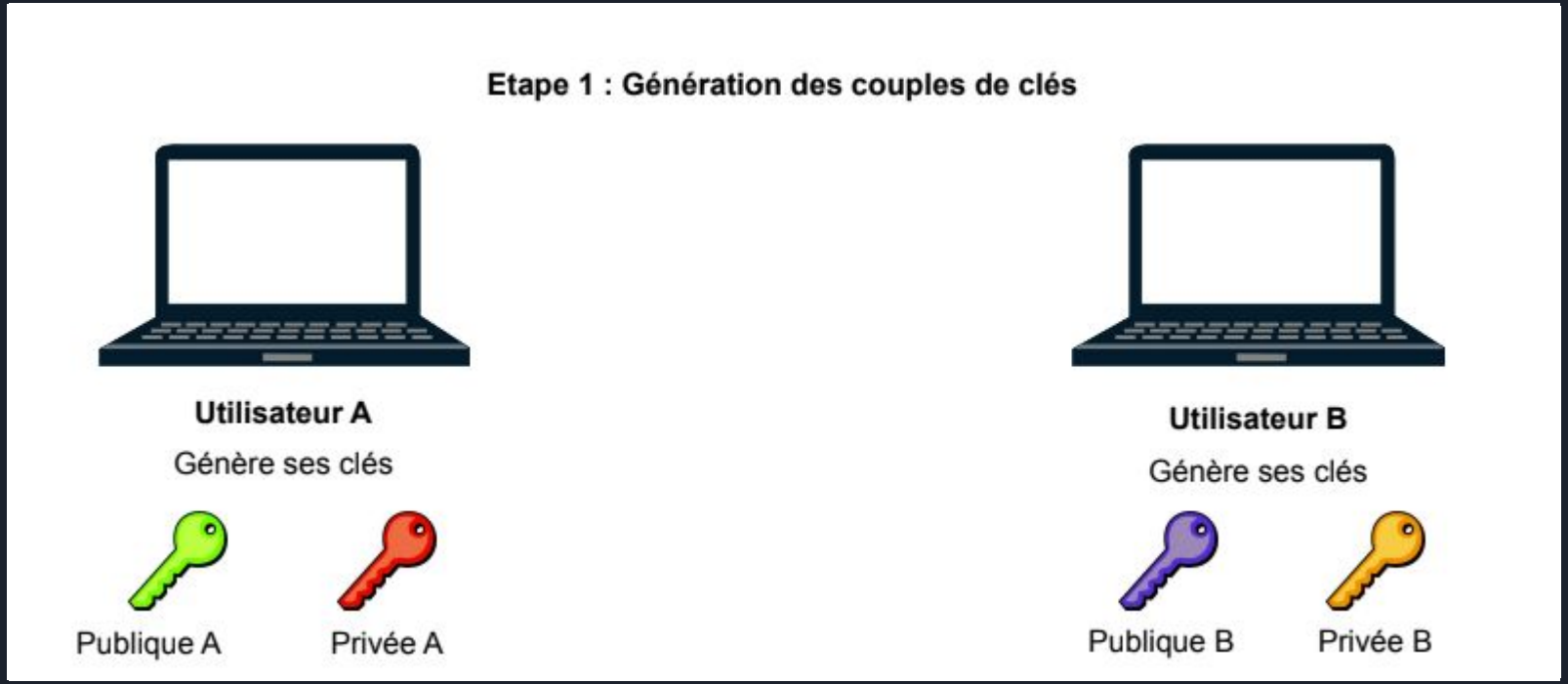
Chiffrement asymétrique

La logique est la suivante :

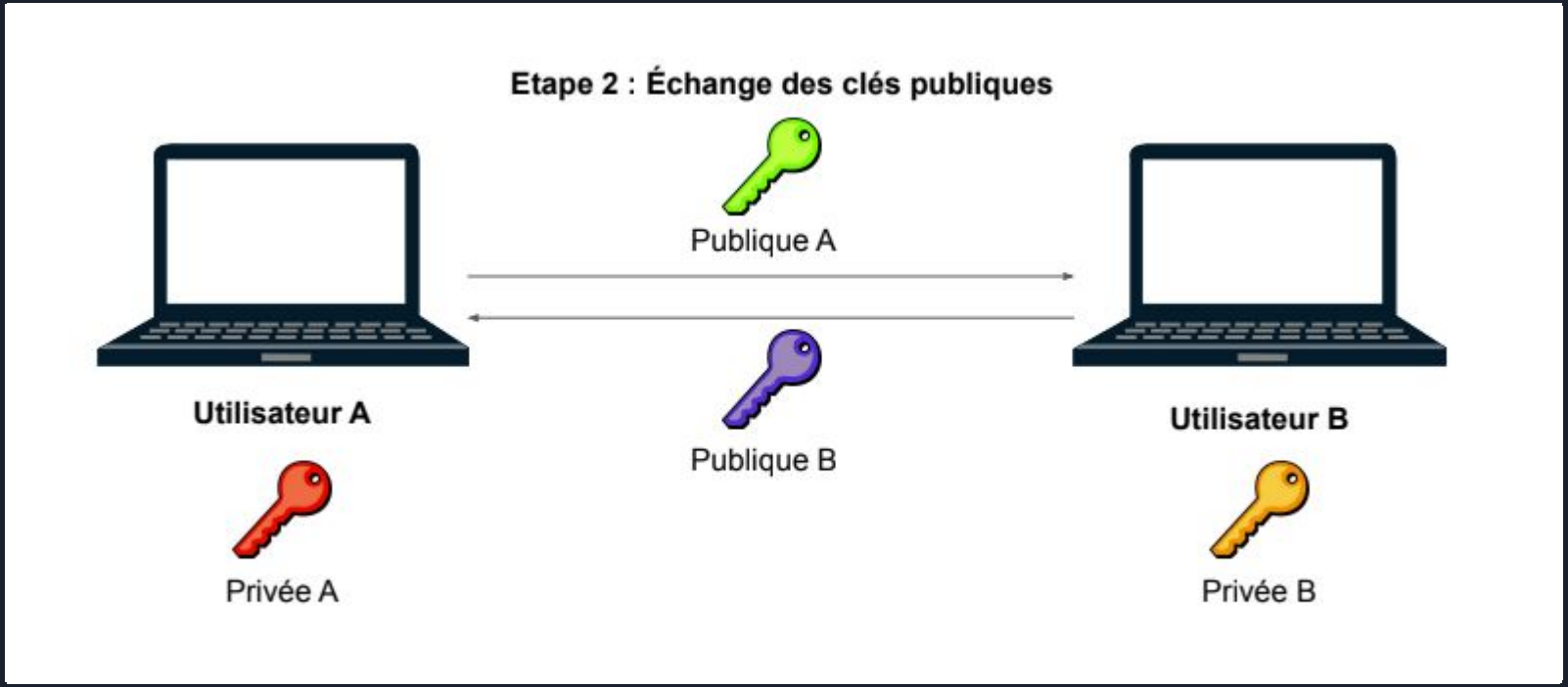
1. Les entités voulant communiquer génèrent chacune un couple de clés (publique / privé).
2. Elles s'envoient leur clé publique de chiffrage.
3. Quand quelqu'un veut envoyer un message, elle le chiffre avec la clé reçue.
4. L'entité recevant les messages les déchiffre avec la clé privée qu'elle a généré au début.

La clé de déchiffrage est dite privée car elle n'est jamais diffusée, contrairement à la clé publique.

Chiffrement asymétrique

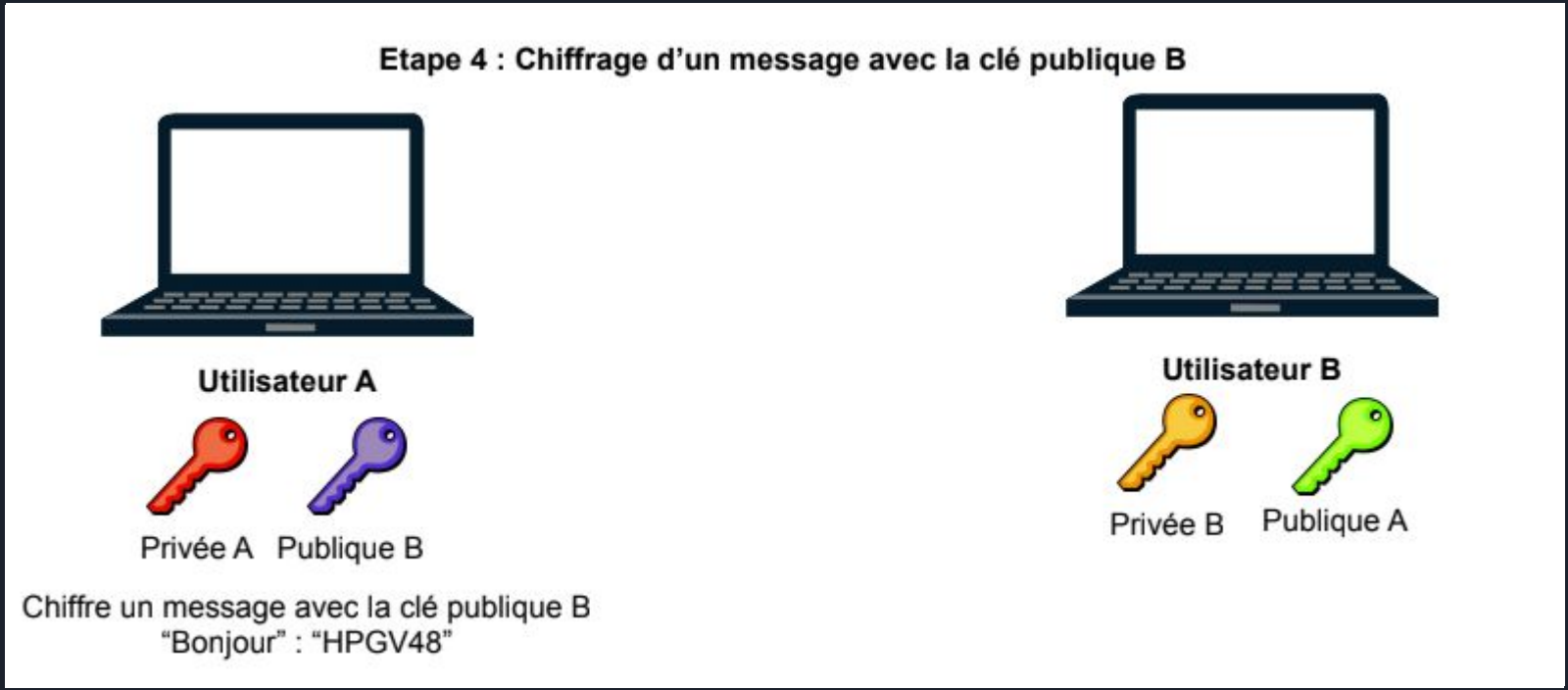


Chiffrement asymétrique

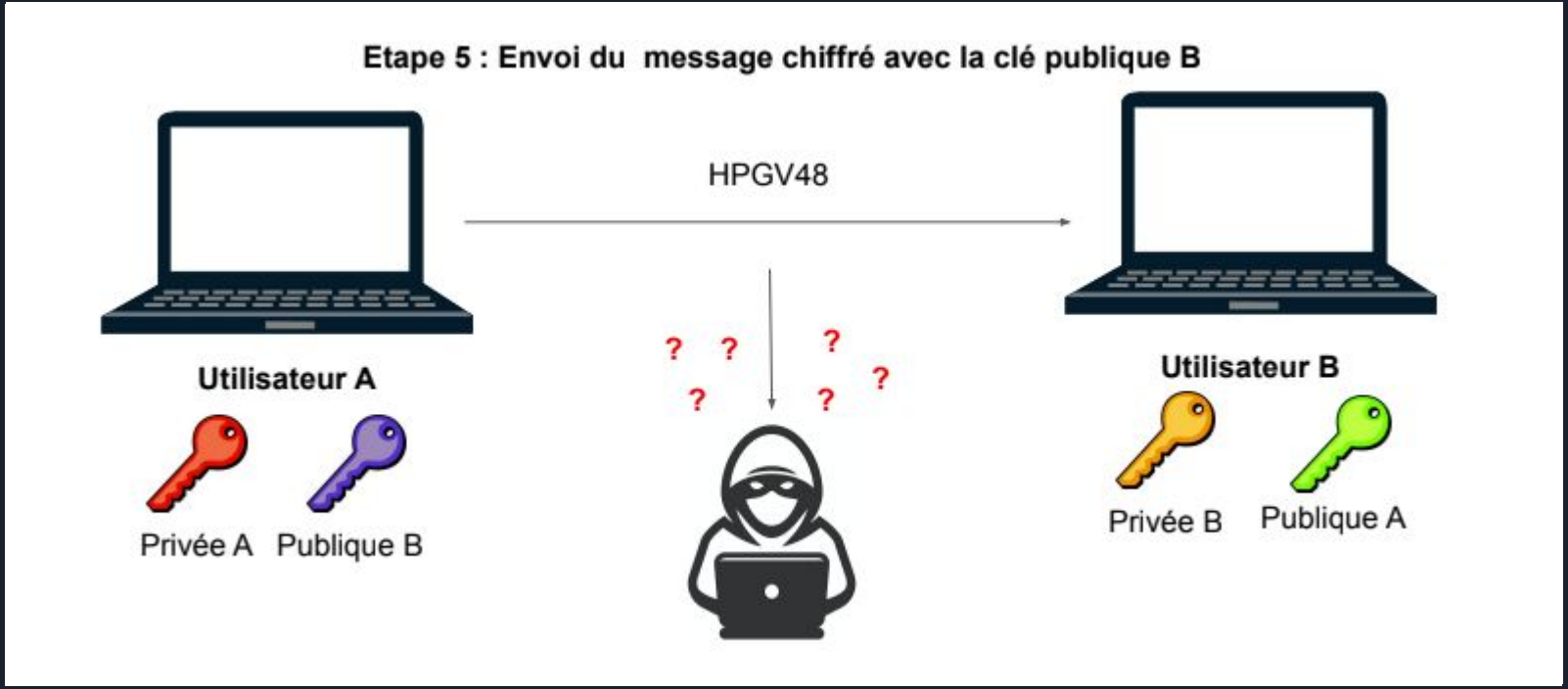


Chiffrement asymétrique

Chiffrement asymétrique



Chiffrement asymétrique



Chiffrement asymétrique

Etape 6 : Déchiffrage d'un message avec la clé privée B

Utilisateur A

Privée A Publique B

Utilisateur B

Privée B Publique A

Déchiffre le message avec la clé privée B
"HPGV48" : "Bonjour"

Chiffrement asymétrique

Etape : Chiffrage d'un message avec la clé publique A

Utilisateur A

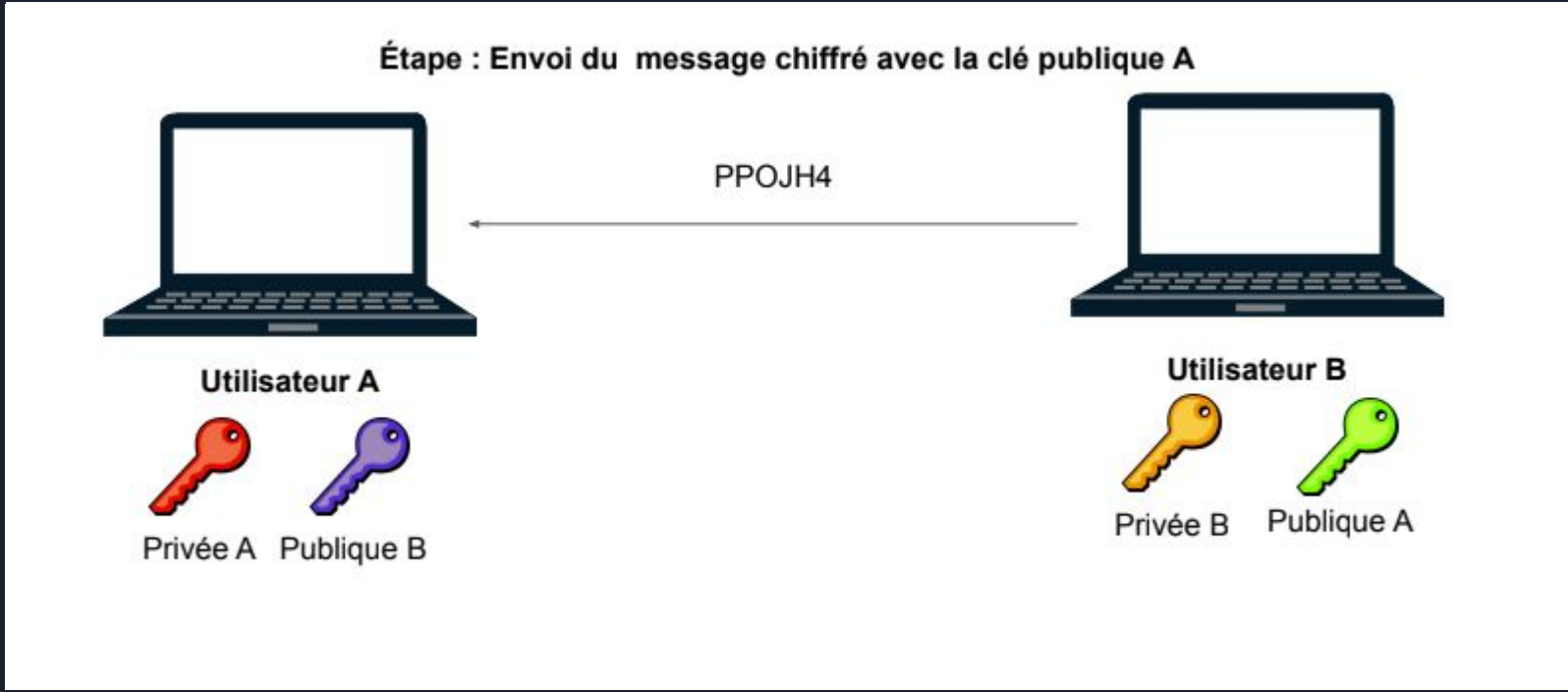
Privée A Publique B

Utilisateur B

Privée B Publique A

Chiffre un message avec la clé publique A
"Secret!" : "PPOJH4"

Chiffrement asymétrique



Chiffrement asymétrique

Etape : Déchiffrage d'un message avec la clé privée A

Utilisateur A

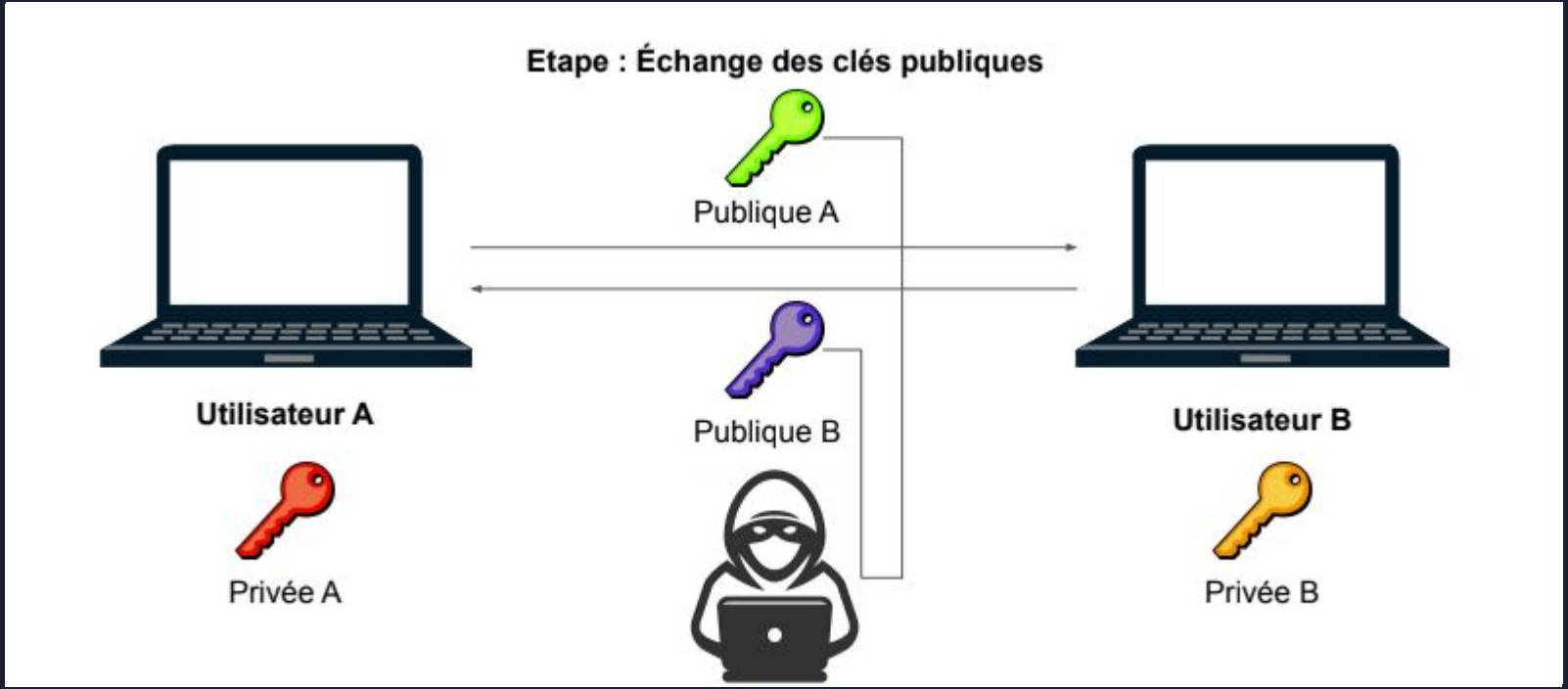
Privée A Publique B

Déchiffre le message avec la clé privée A
"PPOJH4" : "Secret!"

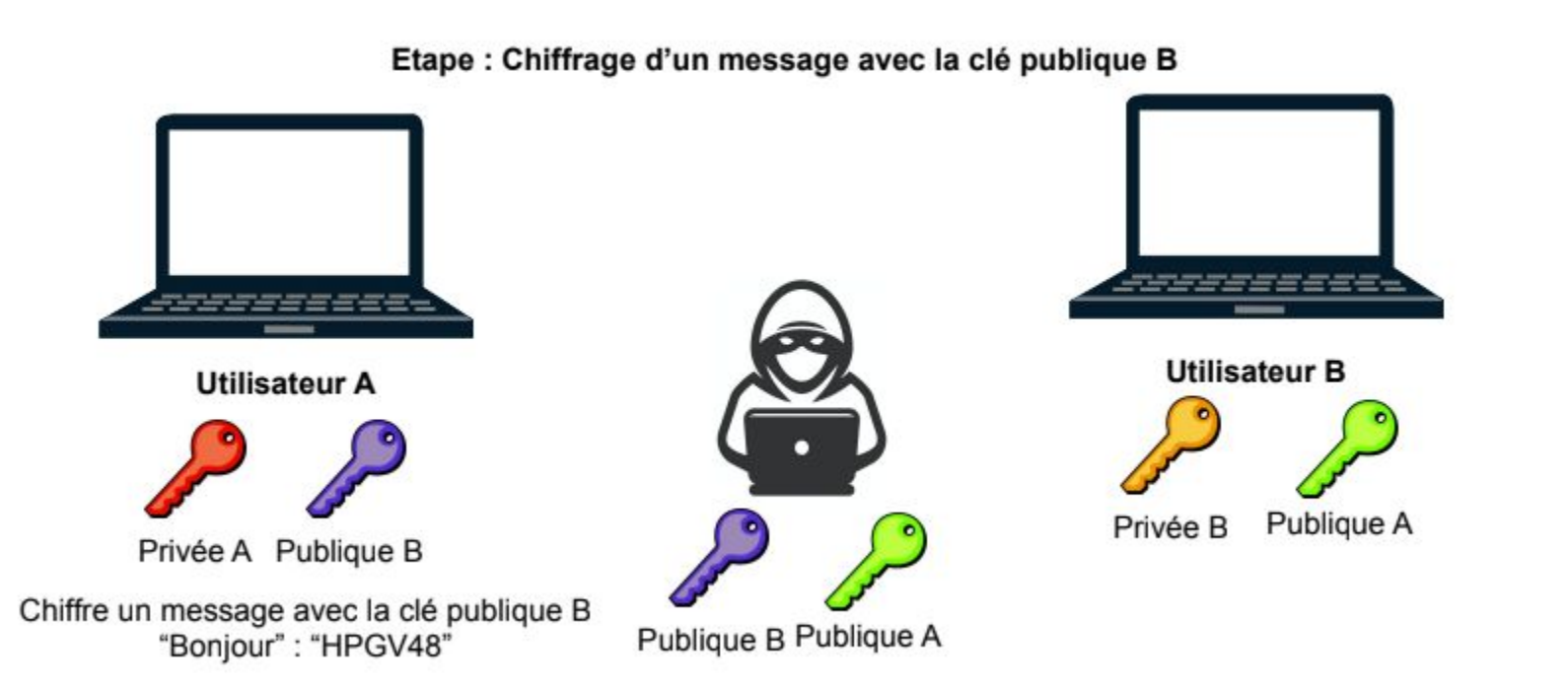
Utilisateur B

Privée B Publique A

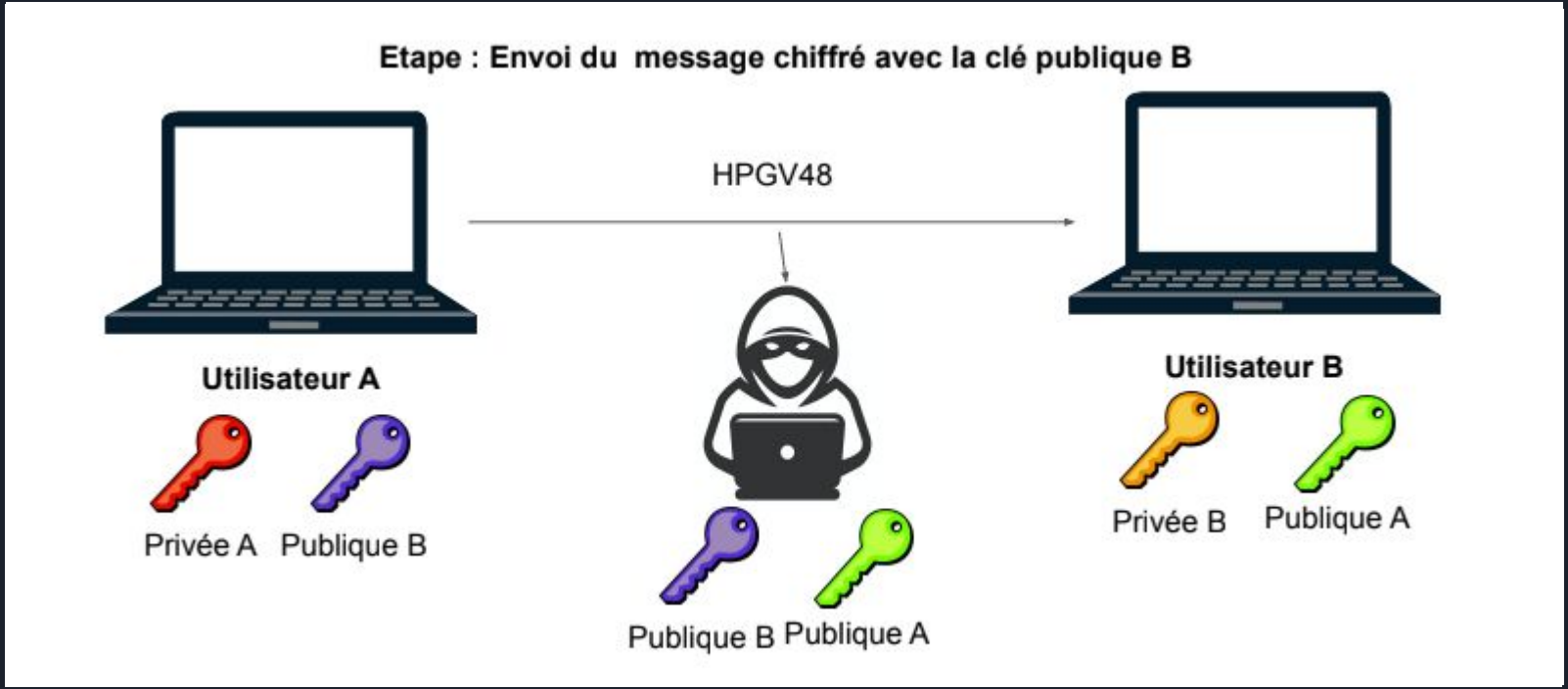
Chiffrement asymétrique - Sécurité



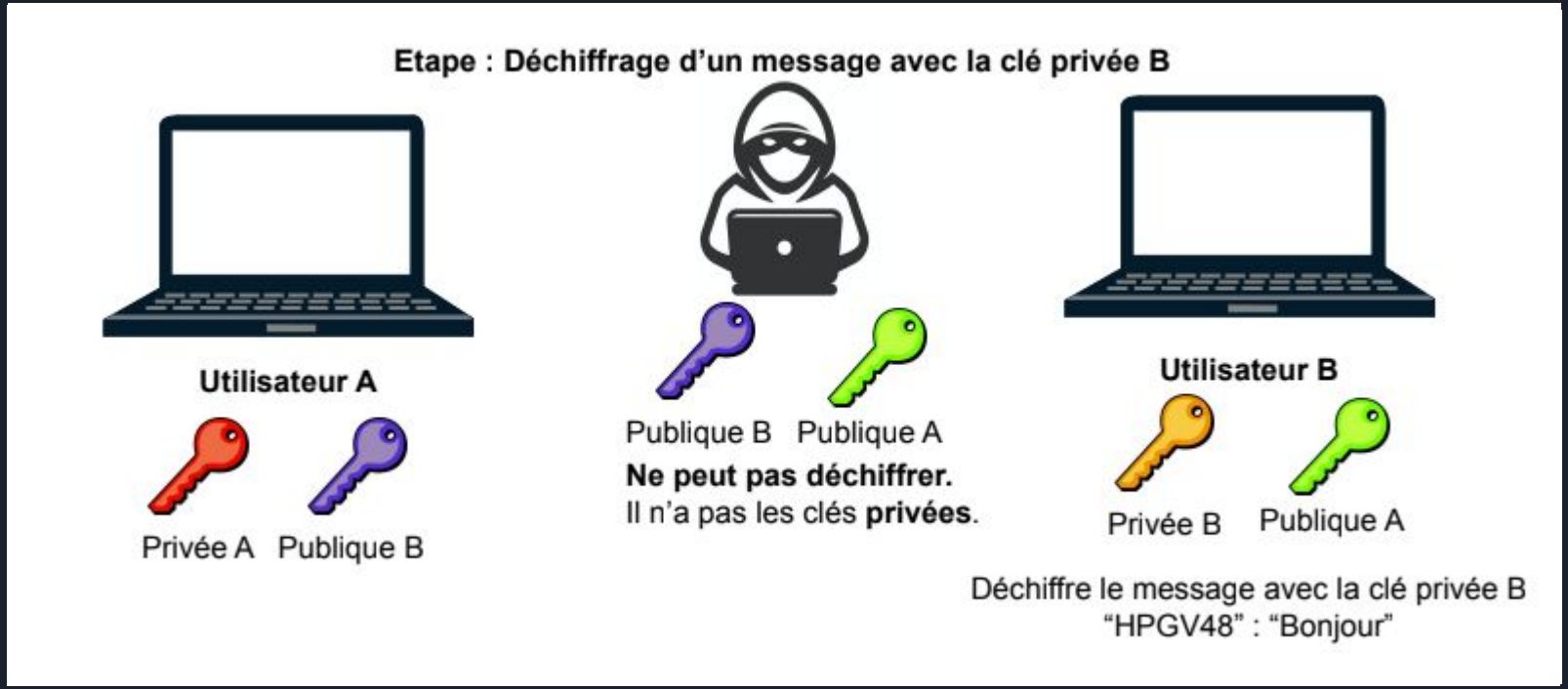
Chiffrement asymétrique - Sécurité



Chiffrement asymétrique - Sécurité



Chiffrement asymétrique - Sécurité



Chiffrement asymétrique - Problème

Le chiffrement asymétrique garantit bien la sécurisation des communications.

Malheureusement, les algorithmes de chiffrement et déchiffrement, de par la nature asymétrique du procédé, sont beaucoup moins efficaces temporellement, et prennent beaucoup trop de temps comparé à chiffrement symétrique. Pour un réseau où toutes les données doivent être chiffrées, cela est très gênant. Il n'est donc pas envisageable d'utiliser cette technique, notamment pour les communications sur Internet.

Heureusement, il existe une méthode combinant chiffrement asymétrique et symétrique pour garantir la sécurité des données transmises tout en conservant l'efficacité temporelle de la méthode symétrique !

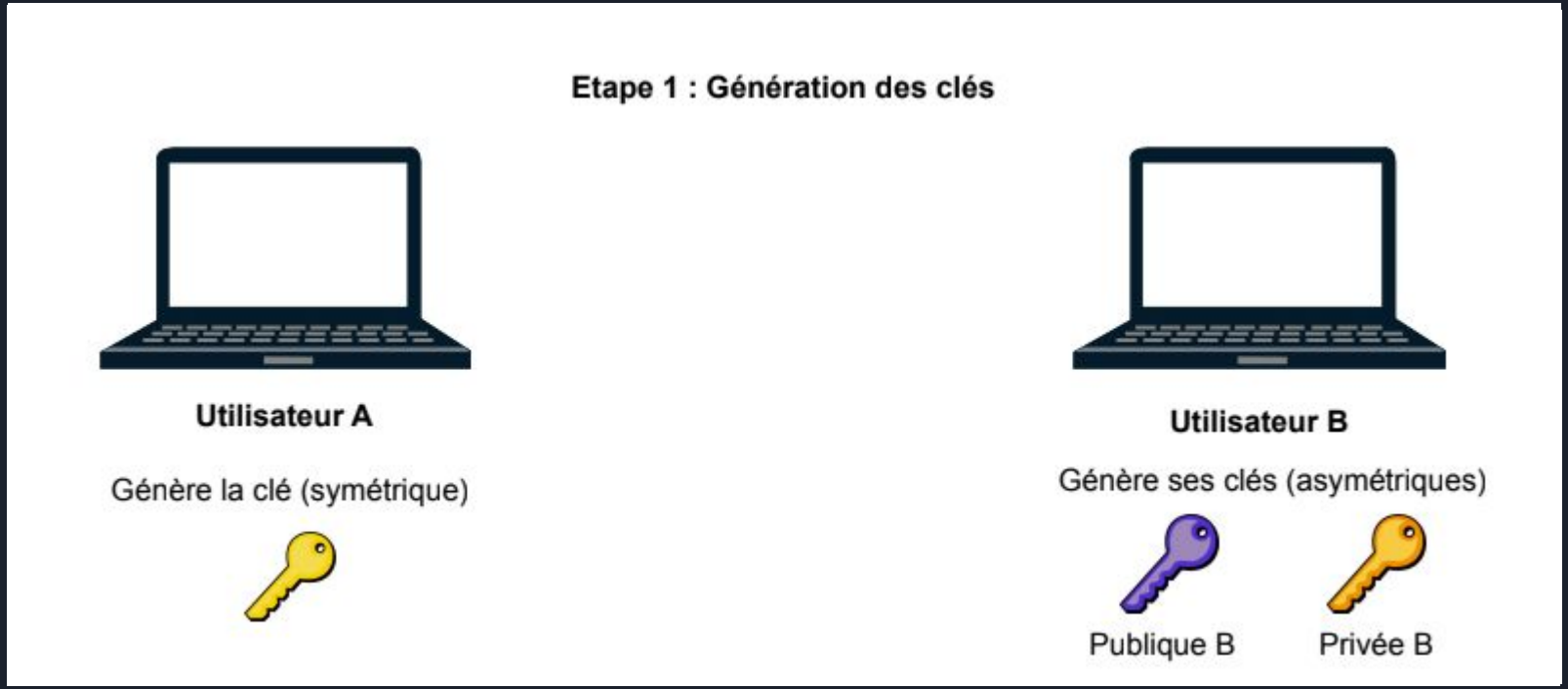
Chiffrement asymétrique + symétrique

La méthode est la suivante, pour un émetteur et un destinataire qui veulent communiquer :

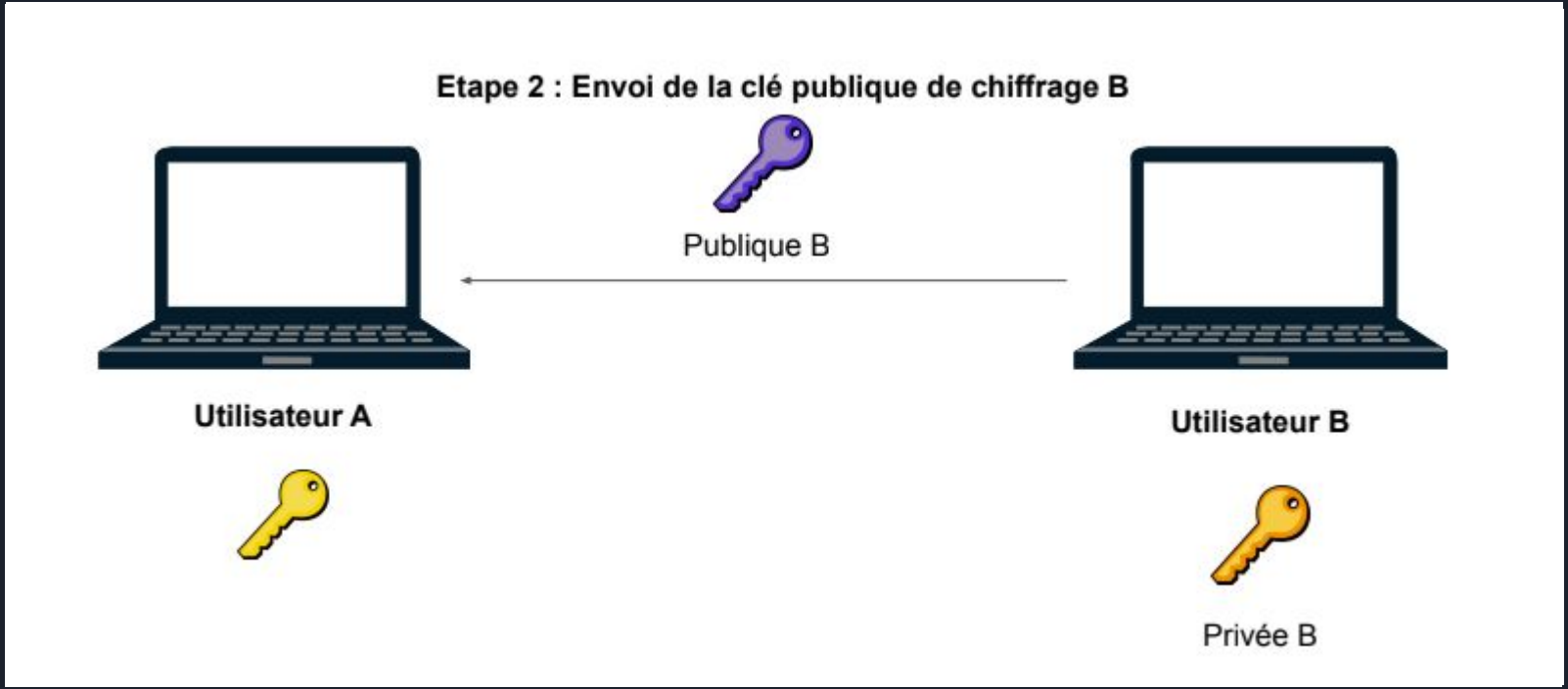
1. Le destinataire, en suivant la méthode asymétrique, génère un couple de clés (une publique / une privée).
2. Il transmet la clé publique de chiffrage à l'émetteur.
3. L'émetteur génère une clé de chiffrement symétrique.
4. A l'aide de la clé publique reçue, il chiffre sa clé de chiffrement symétrique.
5. Il envoi sa clé de chiffrement symétrique sous la forme d'un message chiffré au destinataire .
6. A l'aide de sa clé privée de déchiffrage, le destinataire déchiffre la clé reçue.
7. Les deux interlocuteurs possèdent maintenant une clé symétrique, ils l'utilisent pour communiquer.

Ici, on échange une clé symétrique en suivant un procédé asymétrique, ce qui garantit la sécurisation de cet échange (la clé ne sera pas "volée" par un intermédiaire).

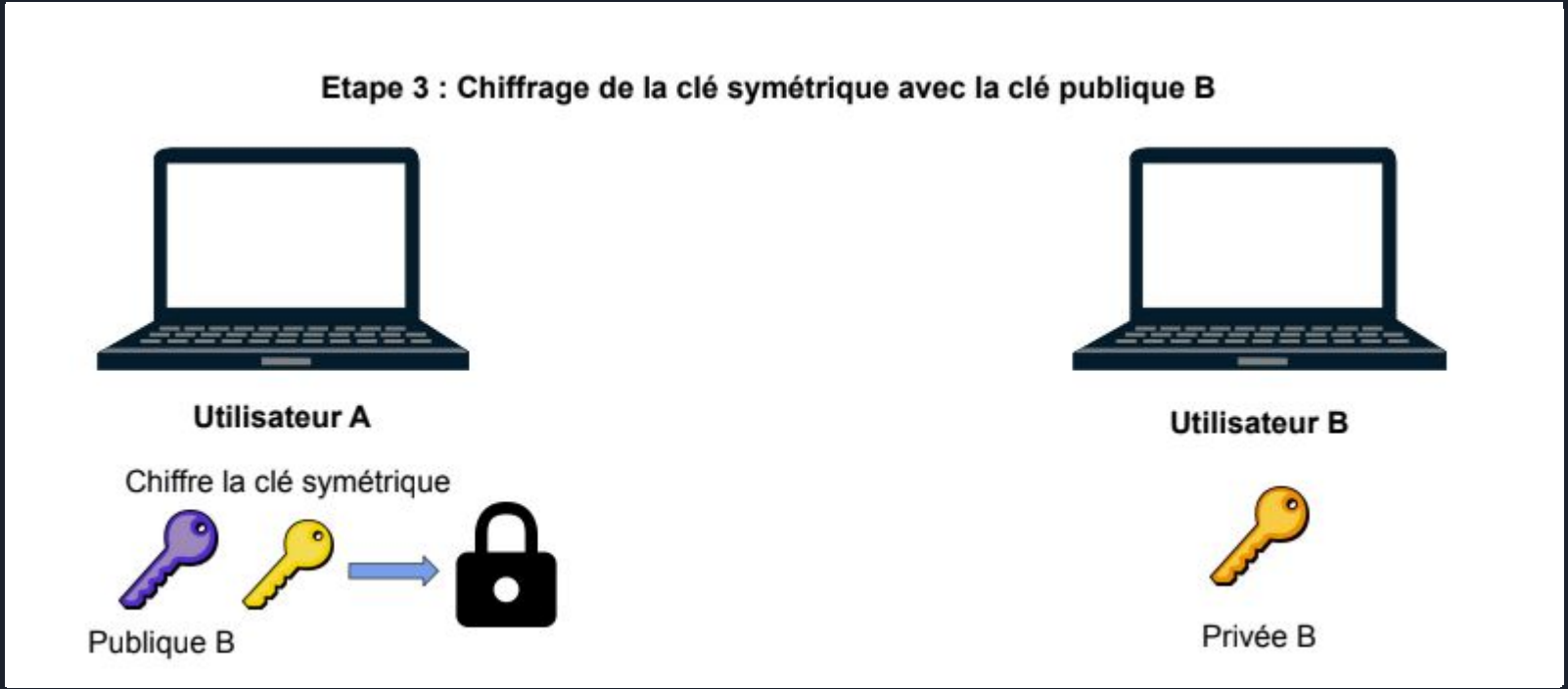
Chiffrement asymétrique + symétrique



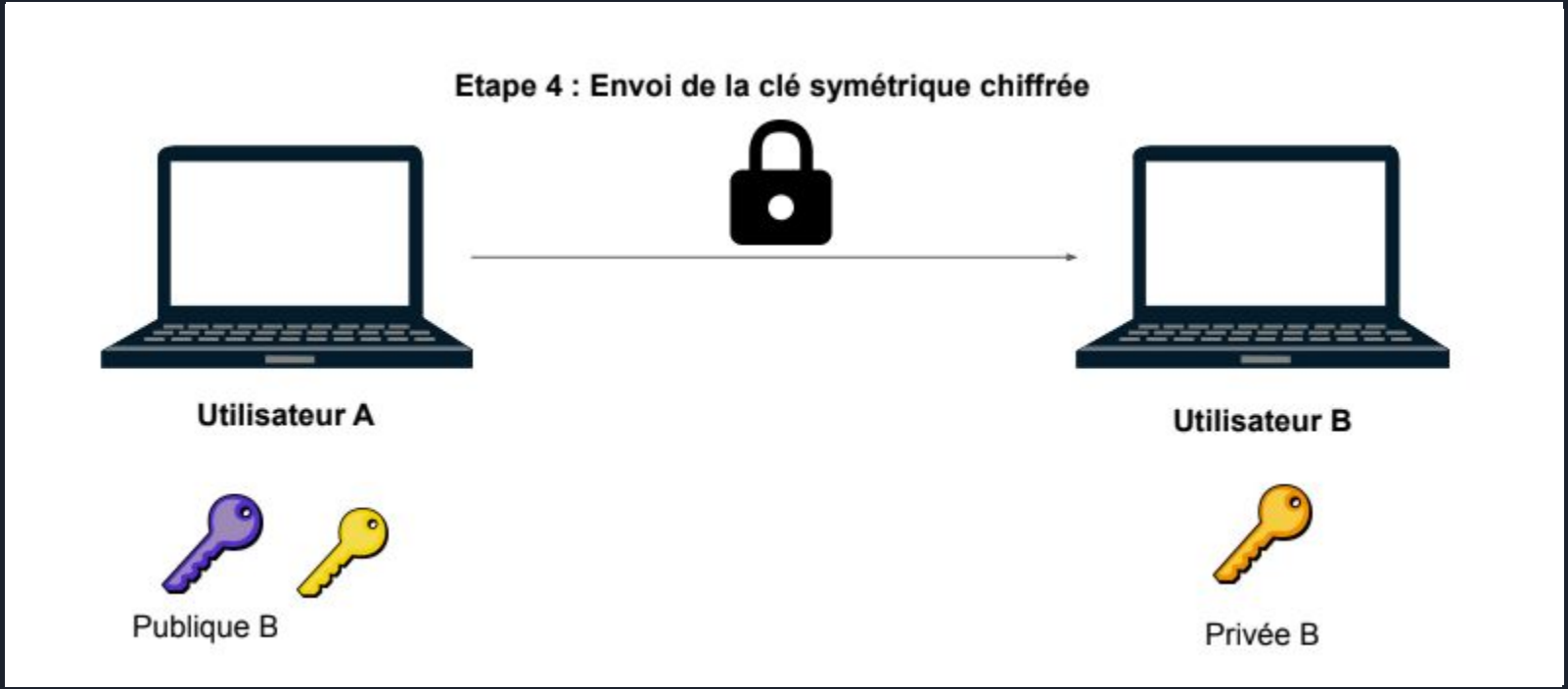
Chiffrement asymétrique + symétrique



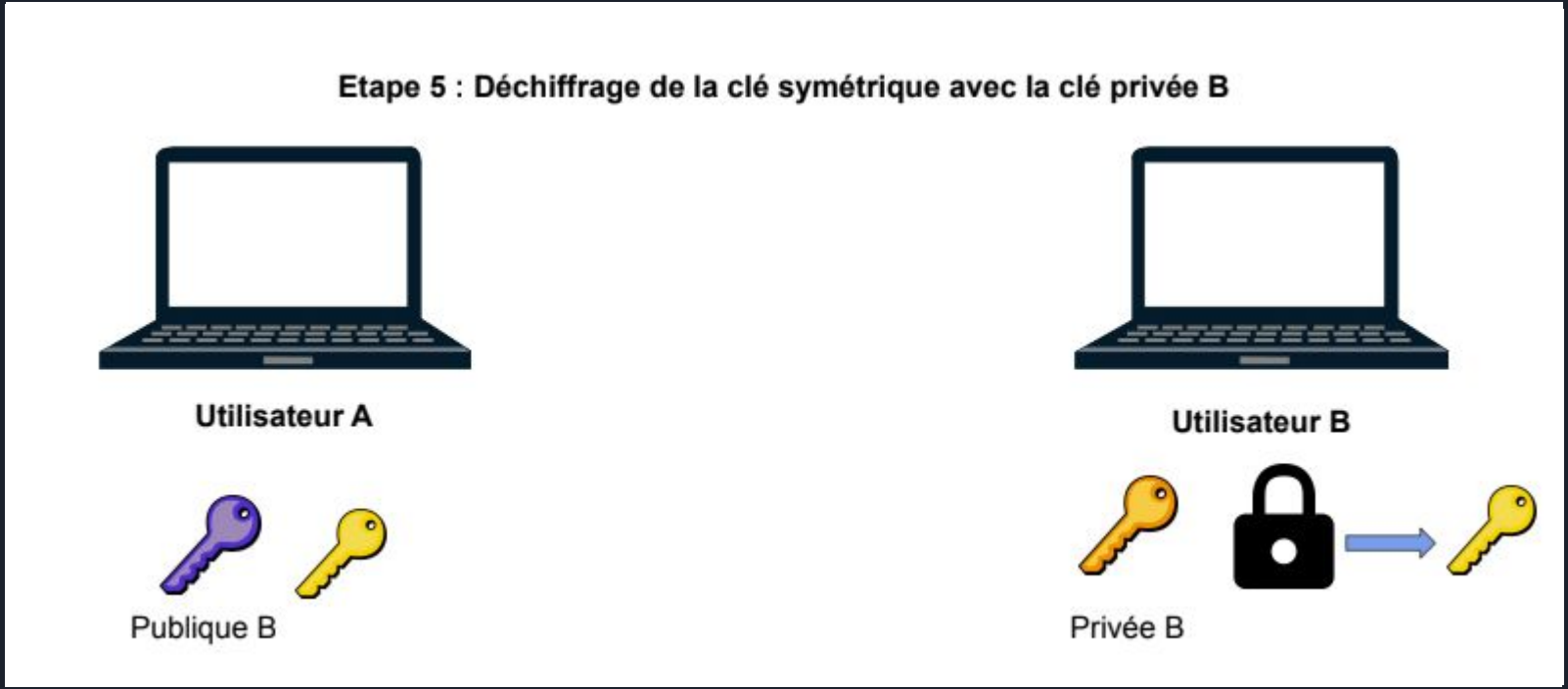
Chiffrement asymétrique + symétrique



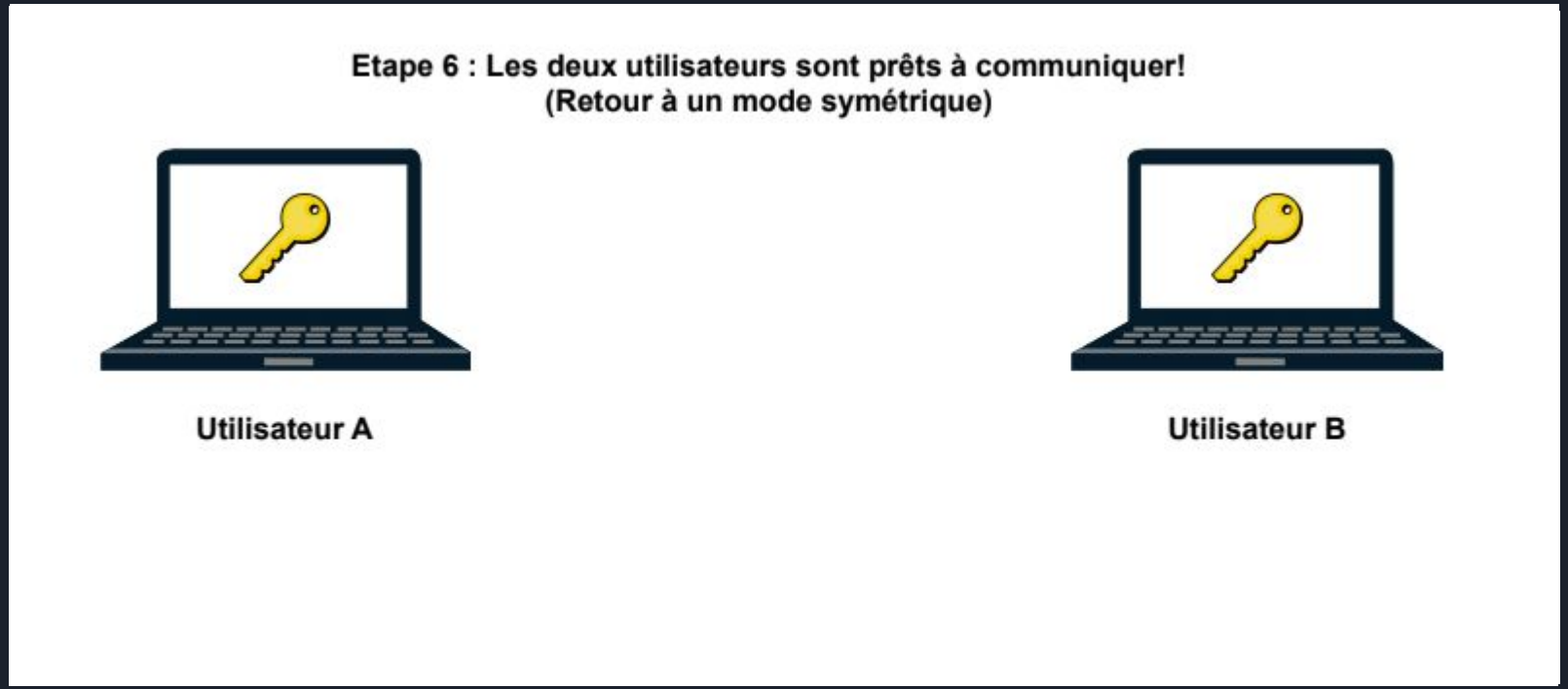
Chiffrement asymétrique + symétrique



Chiffrement asymétrique + symétrique



Chiffrement asymétrique + symétrique



Chiffrement asymétrique + symétrique

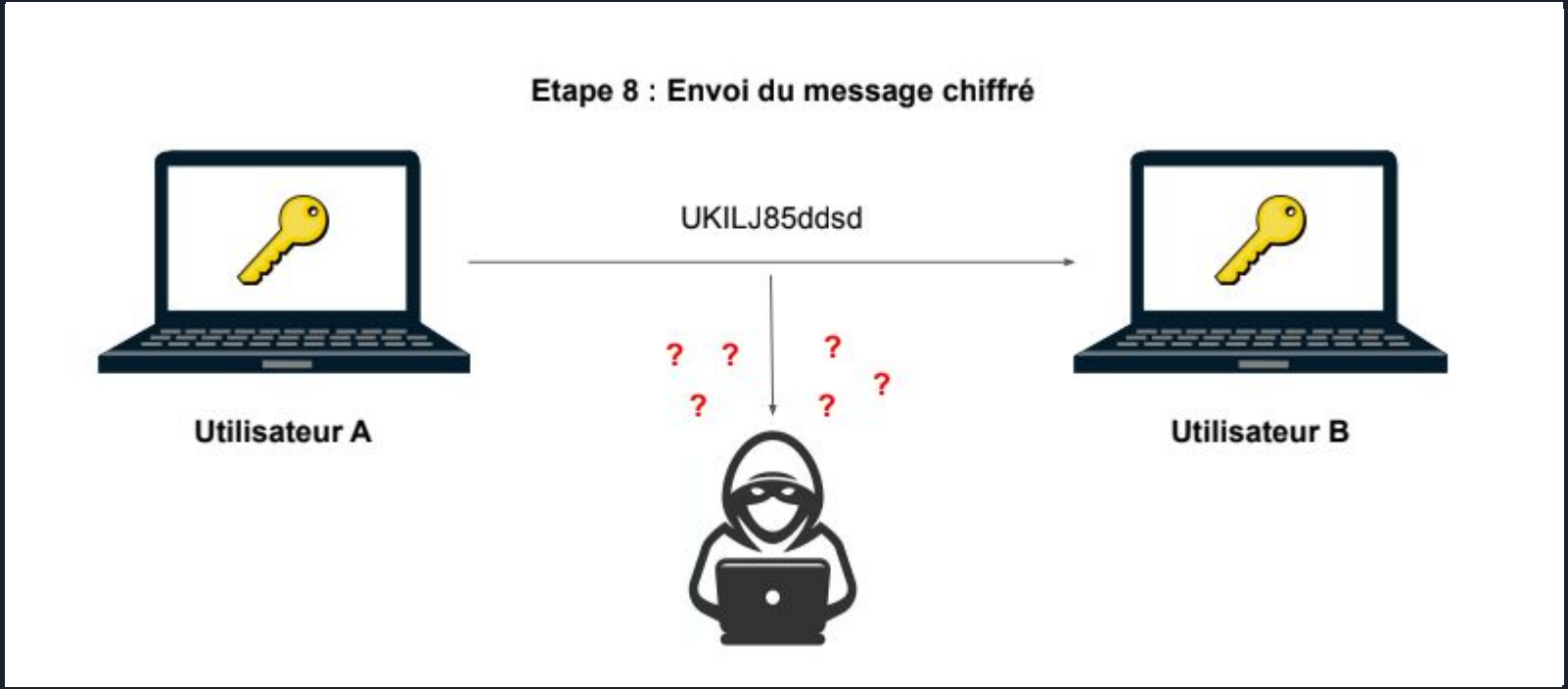
Etape 7 : Chiffrage avec la clé

Utilisateur A

Chiffre le message avec la clé “Bonjour” : “UKILJ85ddsd”

Utilisateur B

Chiffrement asymétrique + symétrique



Chiffrement asymétrique + symétrique

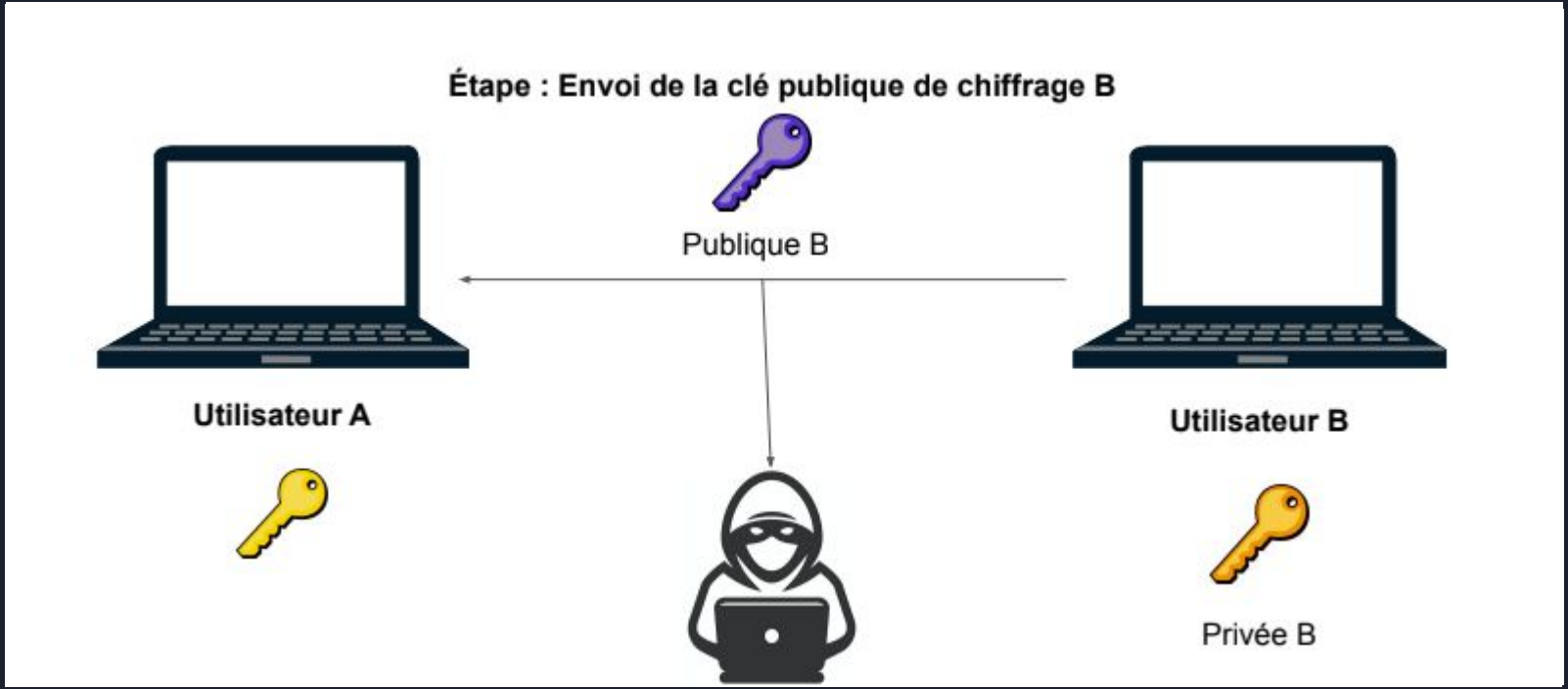
Etape 9 : Déchiffrage avec la clé

Utilisateur A

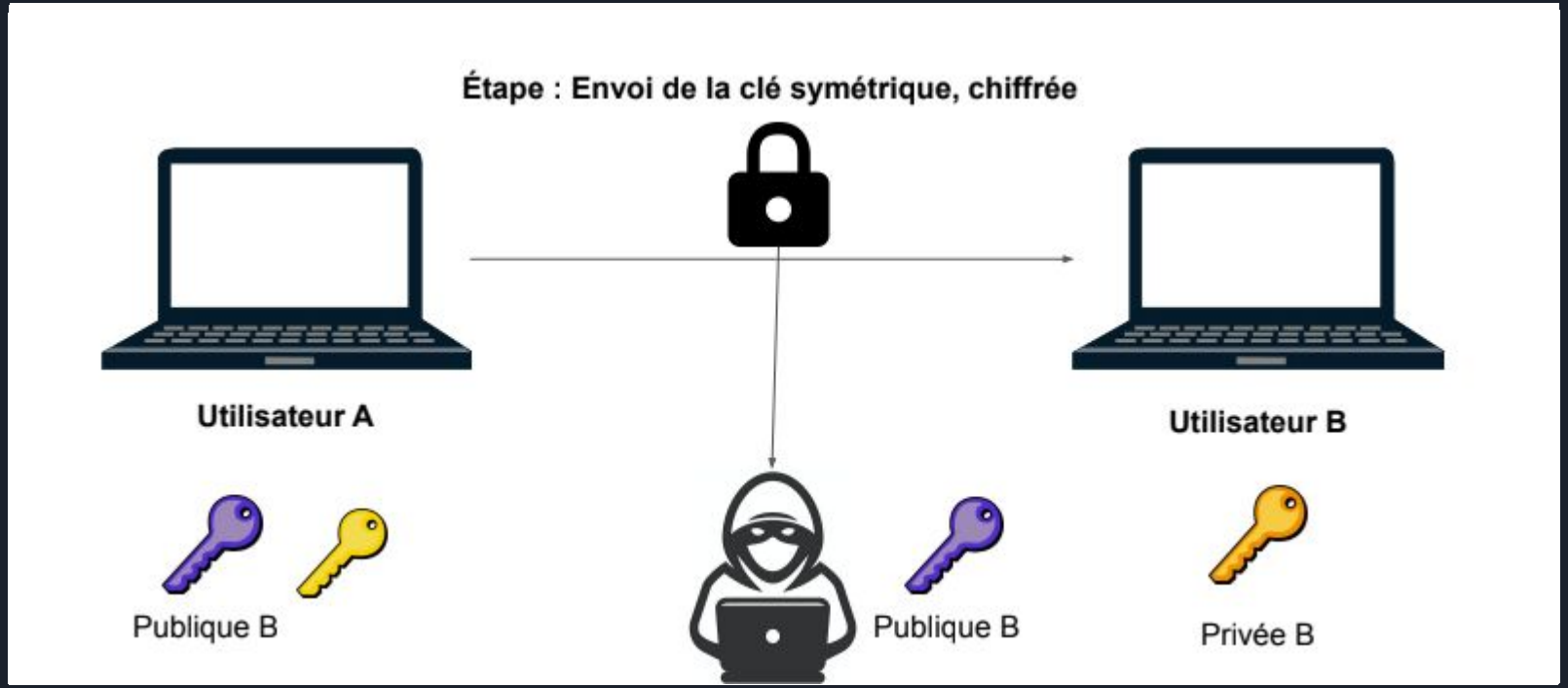
Utilisateur B

Déchiffre le message avec la clé “UKILJ85ddsd” : “Bonjour”

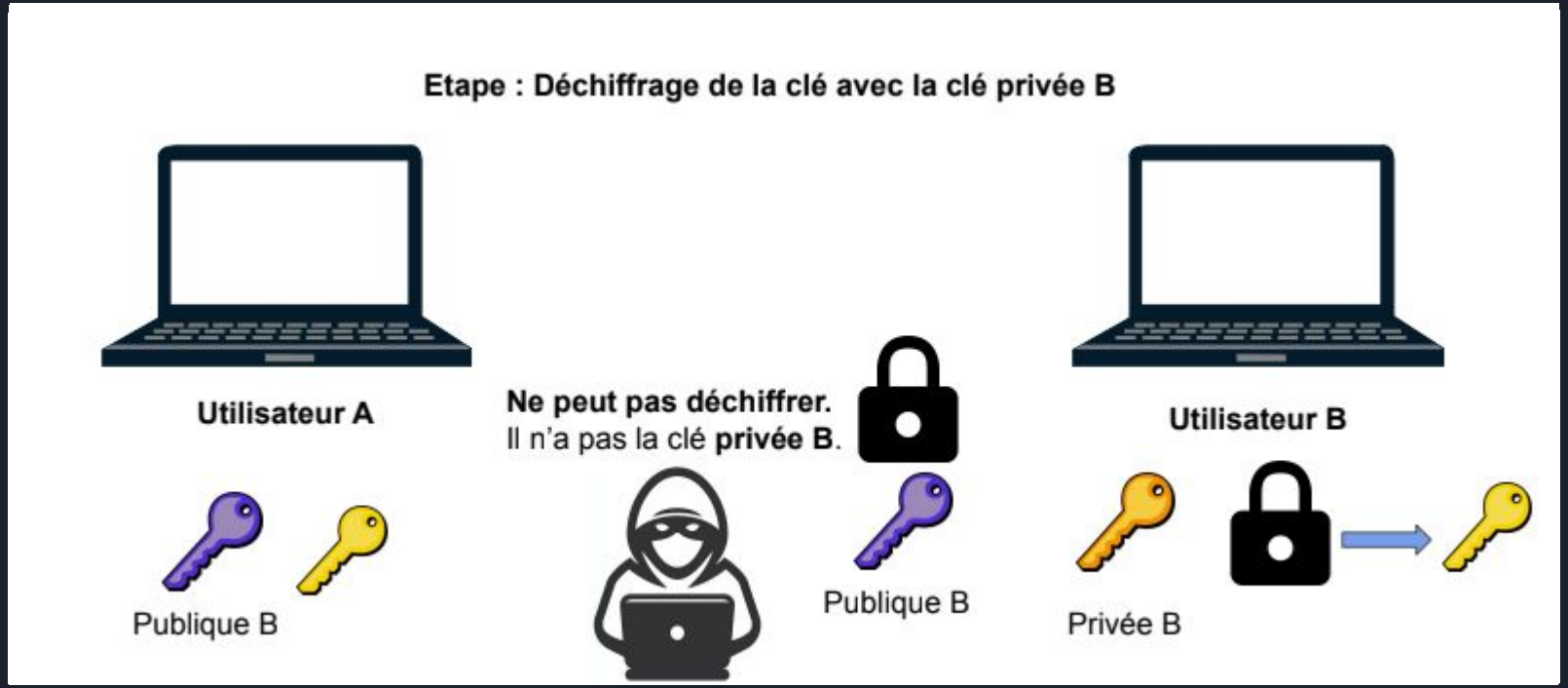
Chiffrement asymétrique + symétrique



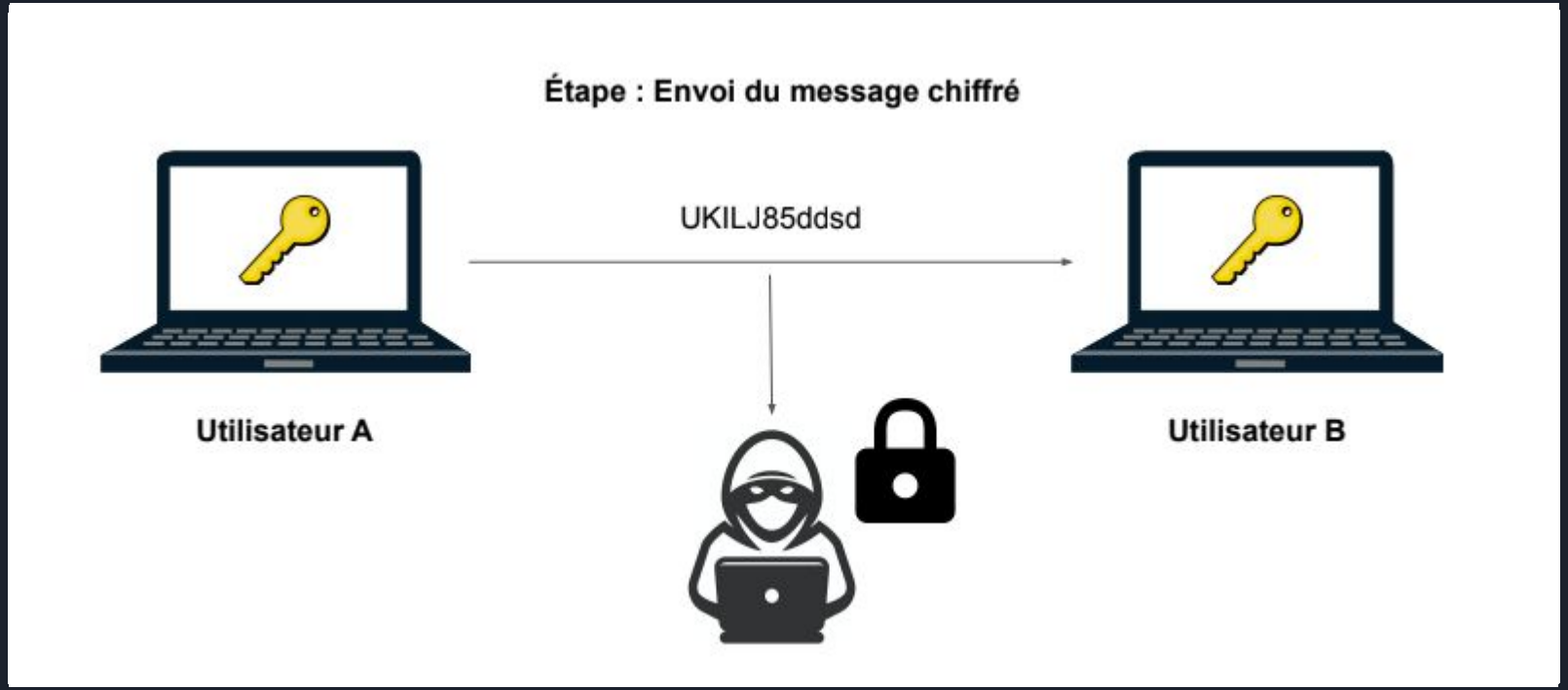
Chiffrement asymétrique + symétrique



Chiffrement asymétrique + symétrique



Chiffrement asymétrique + symétrique



Chiffrement asymétrique + symétrique

Etape : Déchiffrage avec la clé

Utilisateur A

Ne peut pas déchiffrer.
La clé qu'il a est toujours **chiffrée**.

Utilisateur B

Déchiffre le message avec la clé
"UKILJ85ddsd" : "Bonjour"

Protocole TLS

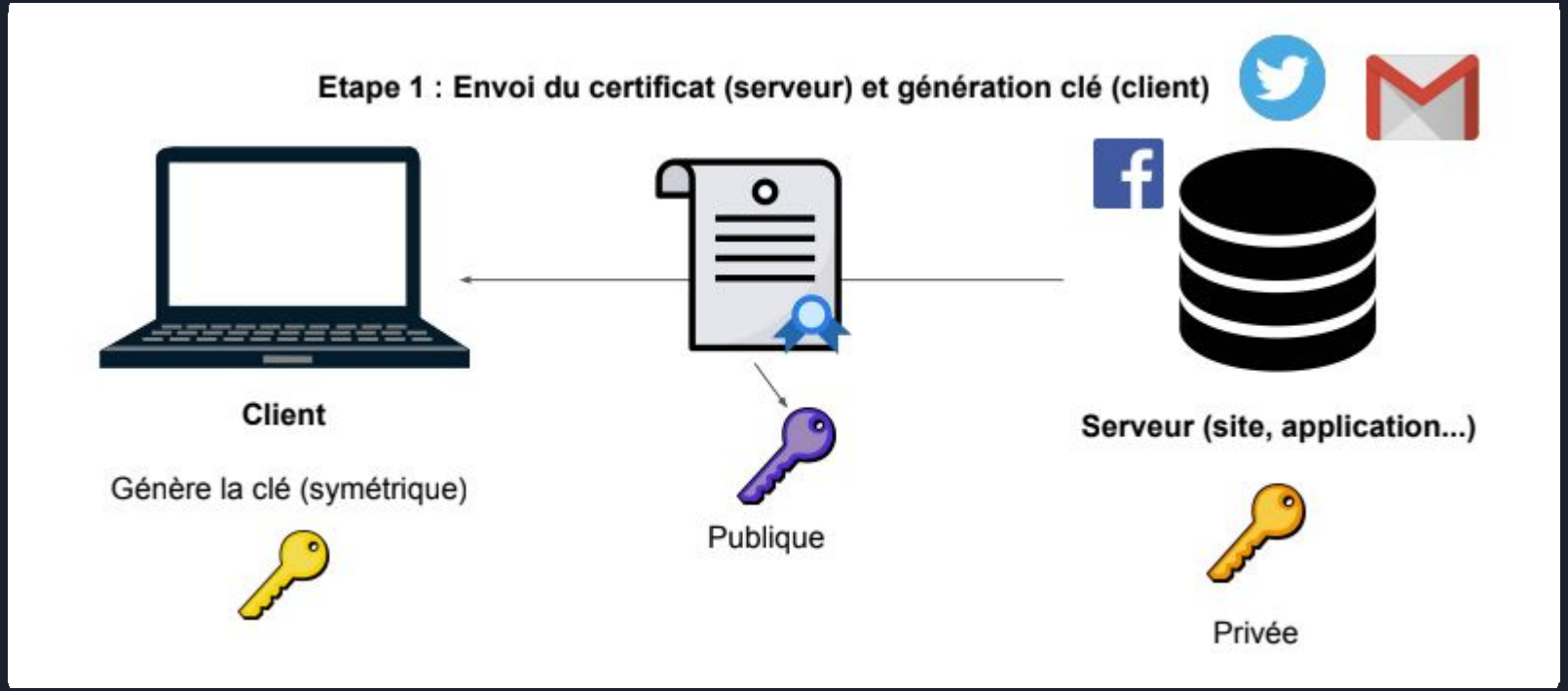
Le protocole TLS (Transfert Layer Security) implémente le fonctionnement que nous venons de voir (échange d'une clé symétrique via un procédé asymétrique). Il assure donc que les échanges de données via le protocole HTTPS soient sécurisés, c'est-à-dire qu'aucune personne susceptible de voir des trames émises au travers de ce protocole ne puissent déchiffrer les données.

Quand on essaye d'accéder à un site web, notre interlocuteur est donc un serveur web. Celui-ci possède un certificat qui est transmis au client (le navigateur) et qui contient sa clé publique.

Ce certificat possède une signature assurant l'authenticité du site web avec lequel on communique (cela assure qu'on n'est pas en train de dialoguer avec un site usurpant l'identité d'un autre site.)

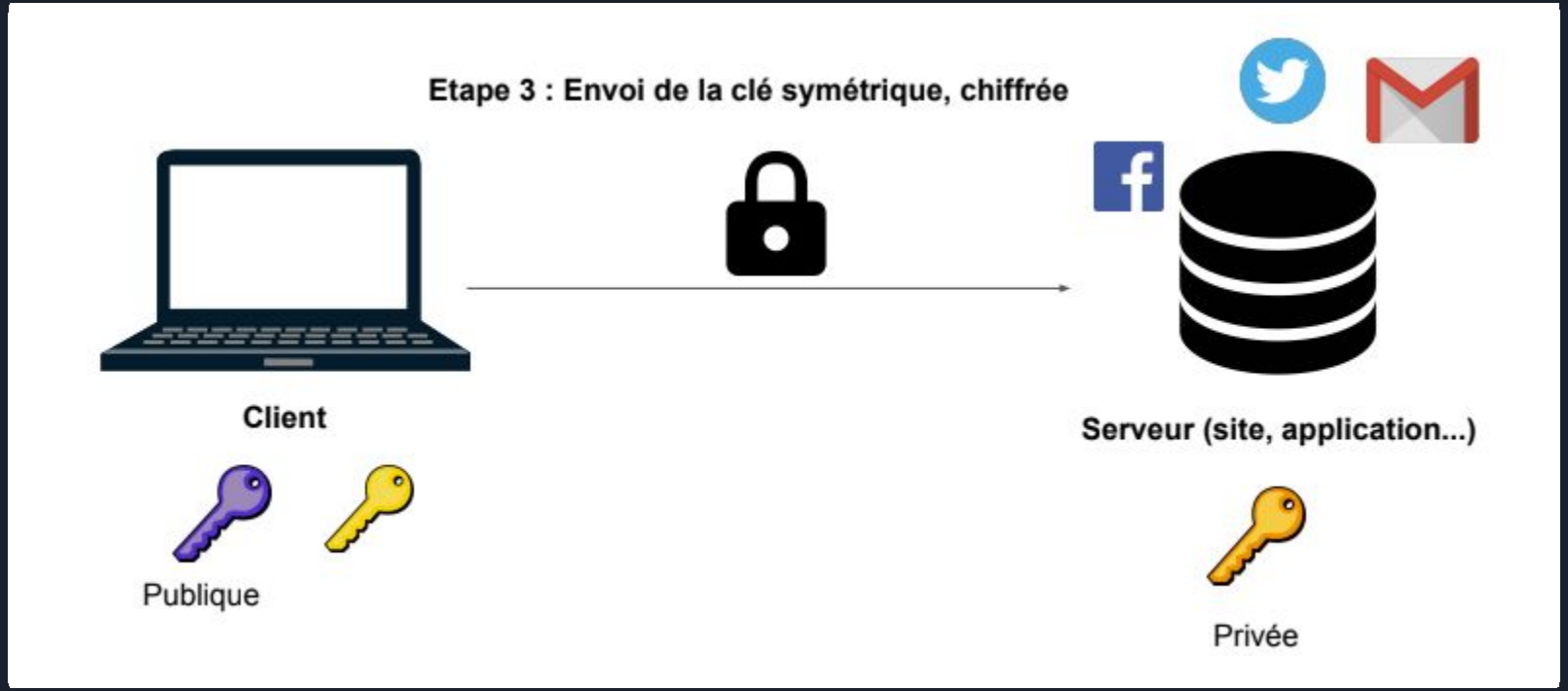
L'obtention de ce certificat est nécessaire pour pouvoir naviguer sur le site via le protocole HTTPS.

Protocole TLS



Protocole TLS

Protocole TLS



Protocole TLS

Protocole TLS

Etape 5 : Le client est prêt à dialoguer avec le serveur!
(Retour à un mode symétrique)

Client

Serveur (site, application...)

Protocole TLS

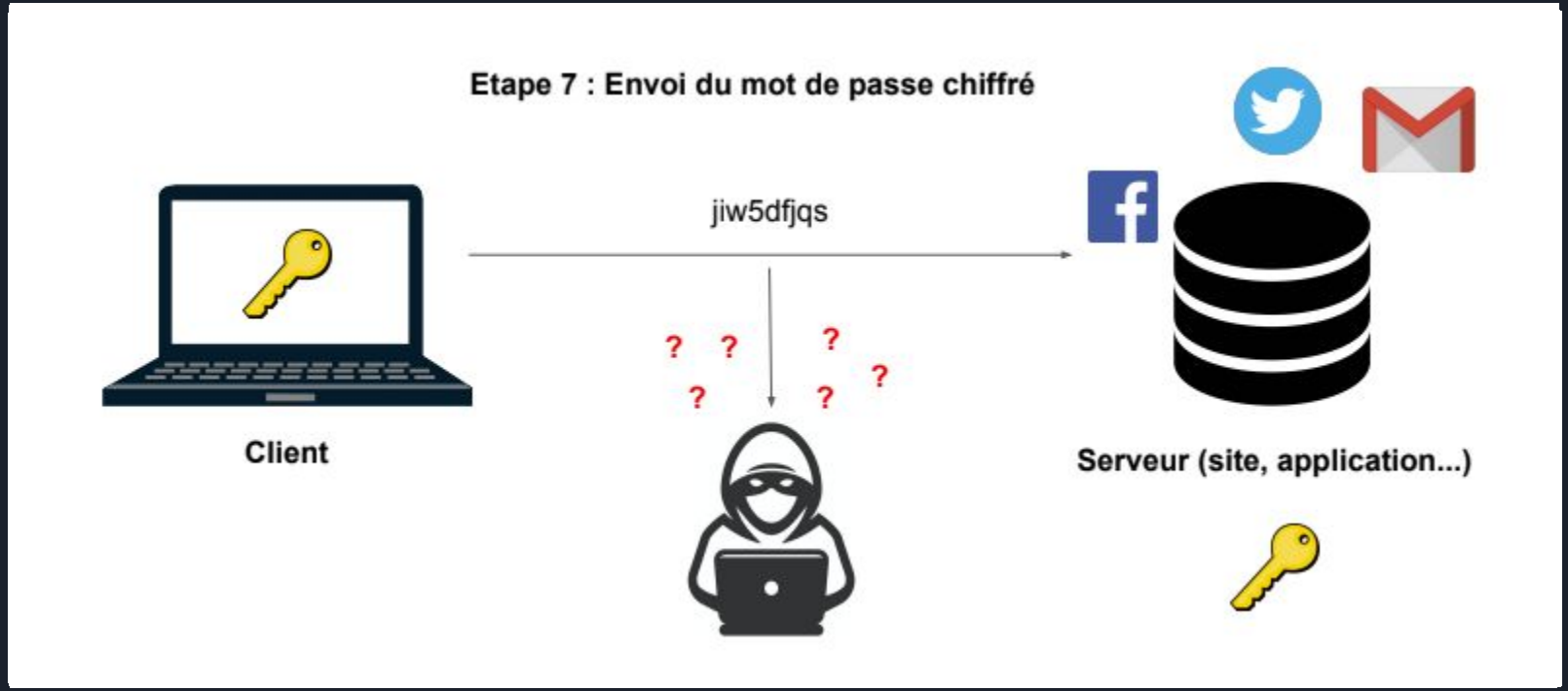
Etape 6 : Chiffrage avec la clé symétrique

Client

Chiffre le mot de passe avec la clé “banane” : “jiw5dfjqs”

Serveur (site, application...)

Protocole TLS



Protocole TLS

Etape 8 : Déchiffrage du mot de passe avec la clé

Client

Serveur (site, application...)

Déchiffre le mot de passe avec la clé
“jiw5dfjqs” : “banane”