

La sécurité des communications

- Objectifs
- Mécanismes et chiffrements
- Signature électronique
- Exemple de mise en œuvre
 - La Messagerie Sécurisée

Objectifs

Avec les techniques de cryptographie, on souhaite atteindre différents objectifs :

1. Confidentialité

→ Données exploitables par les personnes autorisées seulement

On doit pouvoir assurer qu'une personne non autorisée ne pourra avoir accès aux données

2. Intégrité

→ Non-altération des données transmises

On doit pouvoir assurer que les données reçues n'ont pas été modifiées en cours de transmission

Objectifs (suite)

3. Authentification

→ **Identité garantie de l'entité**

On doit pouvoir assurer que l'entité authentifiée (personne, application ou équipement) est bien celle qu'elle prétend être

4. Non répudiation

→ **Preuve que les données ont bien été émises**

On doit pouvoir confirmer que les données ont bien été transmises par cet émetteur qui ne pourra le nier

5. Autorisation

→ **Droits d'accès pour un utilisateur**

On doit pouvoir vérifier l'authenticité du privilège de l'utilisateur

Mécanismes de la sécurité

Afin d'atteindre les objectifs précédents, on met en œuvre les mécanismes et techniques suivants

- **Hachage**
- **Chiffrement symétrique**
- **Chiffrement asymétrique**
- **ICP ou PKI**
- **Certificats X509v3**

Hachage, résumé ou condensé

- Pour condenser ou résumer un document de longueur quelconque en une suite de caractères de longueur fixe
- **Utilisation:**
 - Intégrité de données
 - Signature électronique
- **Algorithmes performants, non réversibles**
- **Les algorithmes**
 - **MD5 (Message Digest 5) MIT**
 - → 128 bits
 - **SHA-1 (Secure Hash Alg.) NIST**
 - → 160 bits
 - **MAC (Message Authentication Code)**
 - Condensé avec clé secrète

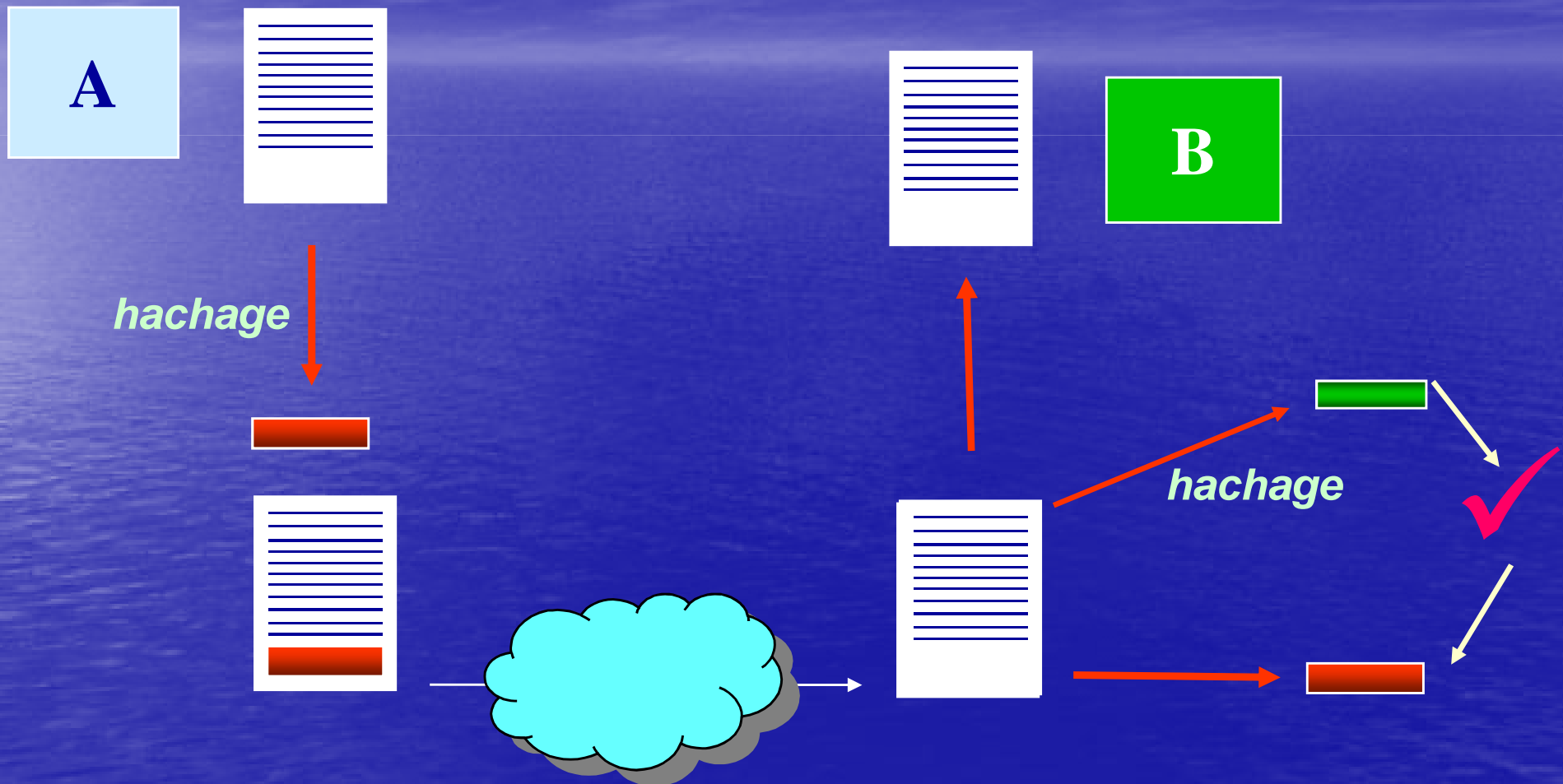
Chiffrement symétrique

- **Méthodes de chiffrement à clé secrète**
 - **Algorithmes**
 - **DES (Data Encryption Standard)** IBM 1968 *56 bits*
 - **3-DES Triple DES** *112 bits*
 - **RC4, RC5 (Rivest Code)** *128..256 bits*
 - **AES Advanced Encryption System** *Rijndael (B) 2000*
Clés de 128, 192 ou 256 bits
- + **Opérations simples performantes et rapides**
- **Administration des clés : Génération, Distribution, Partage et Stockage**

Chiffrement asymétrique

- Association d'une bi-clé : clé publique et clé privée
 - Algorithmes
 - **DH** (Diffie-Helman) (1975)
→ clés de 512...2048 bits
 - **RSA** (Rivest, Shamir, Adleman) 1977
→ clés de 512...2048 bits
 - **ECC** Elliptic Curve Crypto
→ clés de 57 ... 237 bits

2. Intégrité



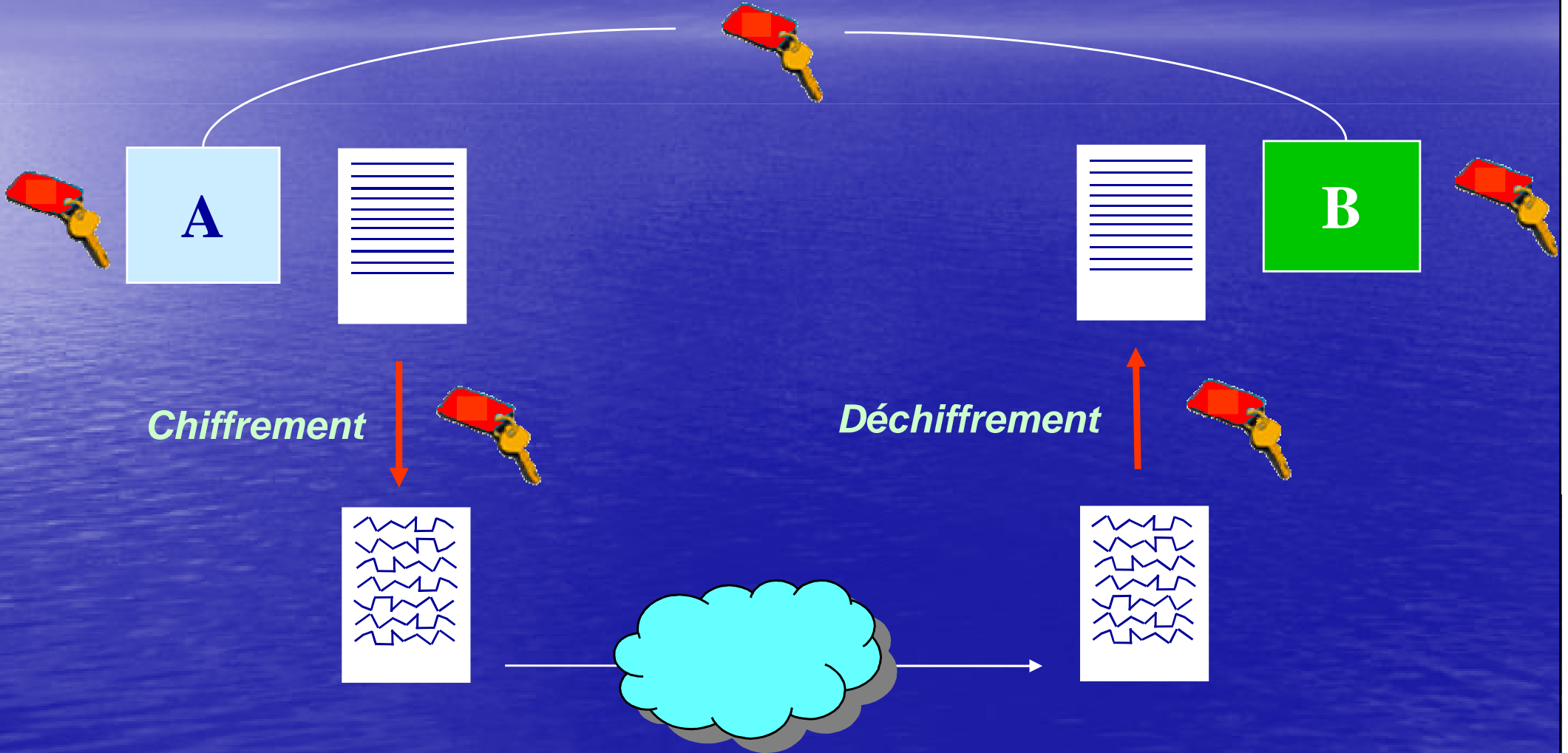
2. Intégrité

- **B reçoit un document de A et veut vérifier l'intégrité du document**
 - **A condense le document**
 - **A émet le document et son condensé vers B**
 - **B extrait le condensé**
 - **B condense le document et compare le résultat avec celui obtenu**
 - **Si identique, le document est intègre (car une altération du document durant la transmission implique un condensé différent)**

1 & 2. Confidentialité

et Intégrité

chiffrement symétrique



2. Confidentialité et Intégrité

avec chiffrement symétrique

- A choisit une clé secrète
- A communique cette clé à B (échange complexe à sécuriser)
- A chiffre le document avec cette clé et le transmet dans le réseau
- À réception, B déchiffre le document avec la clé secrète
 - Sans la clé secrète, le document ne peut être exploité

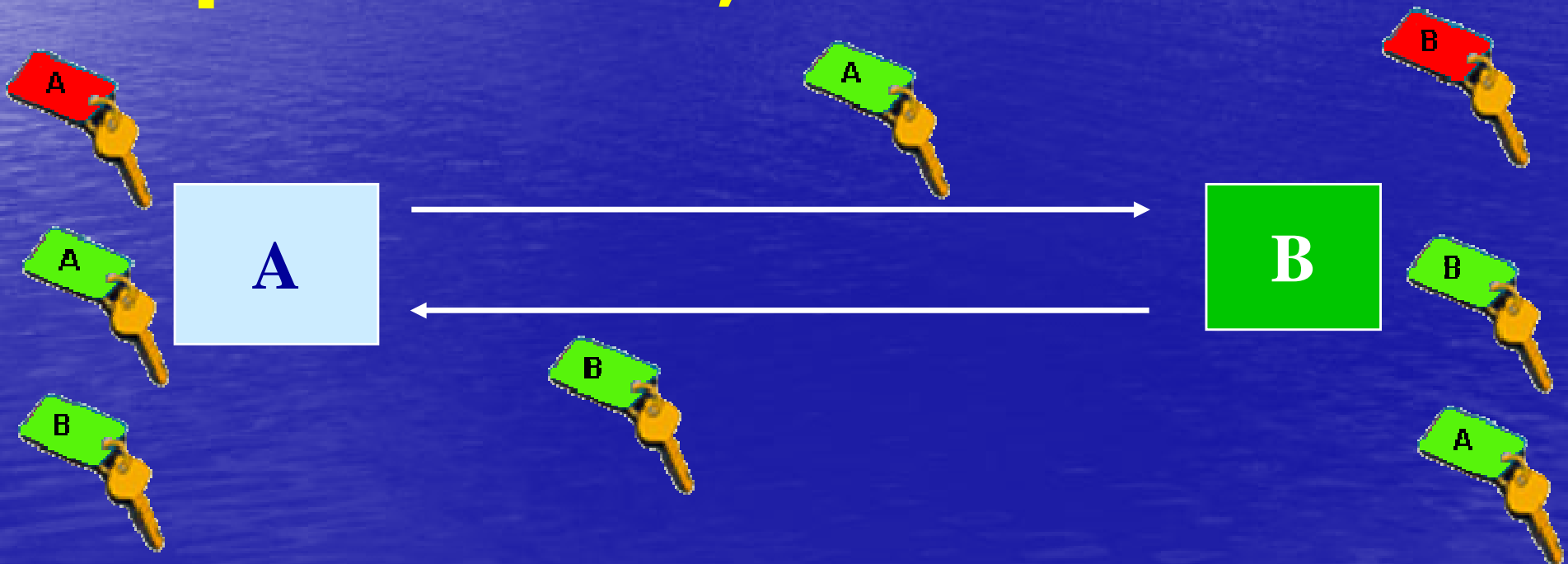
Chiffrement asymétrique préambule

- A et B génèrent chacun une bi-clé
 - Clé publique (verte)
 - Clé privée (rouge)
- A transmet sa clé publique à B
(pas de problème particulier de sécurité)
- B transmet sa clé publique à A
(pas de problème particulier de sécurité)

Chiffrement asymétrique

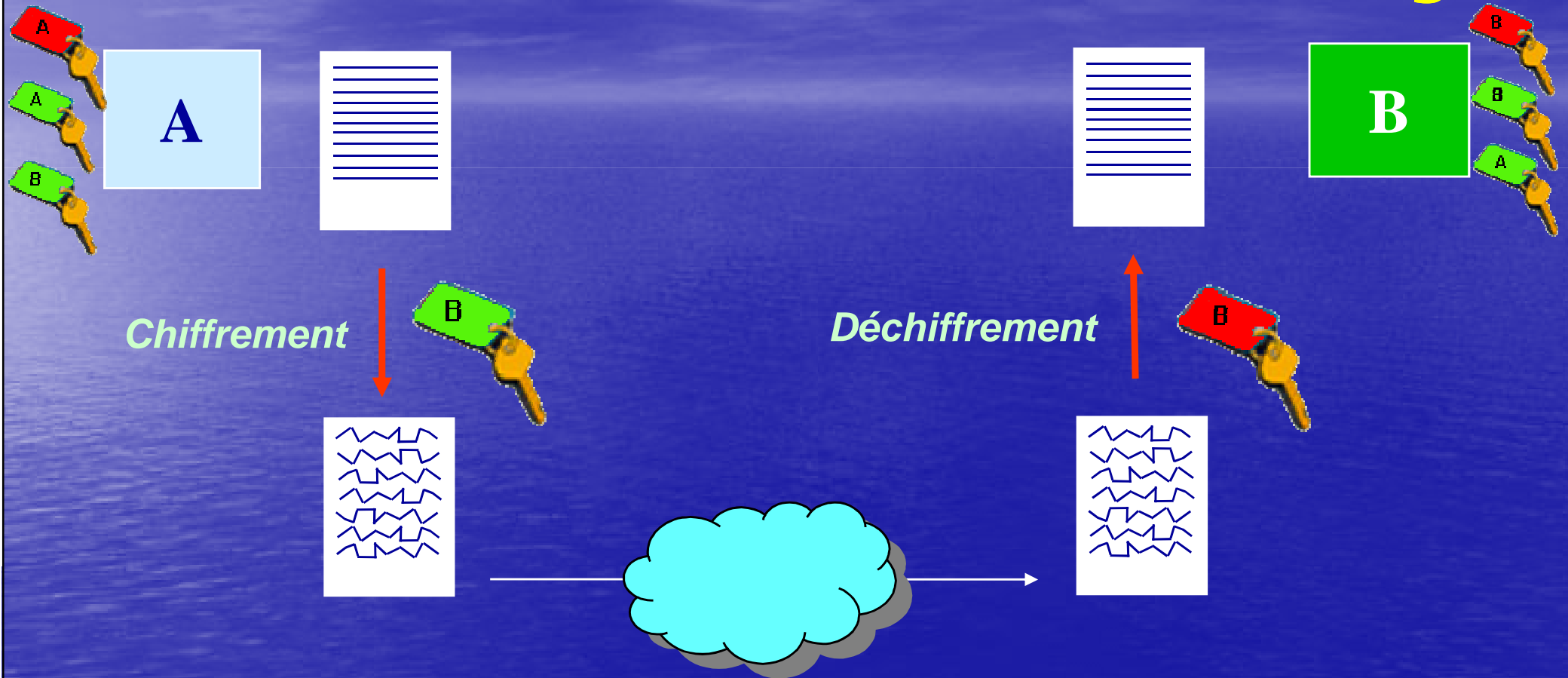
Échange des clés

- A possède **A**, **A** et **B**
- B possède **B**, **B** et **A**



1 & 2. Confidentialité

et Intégrité

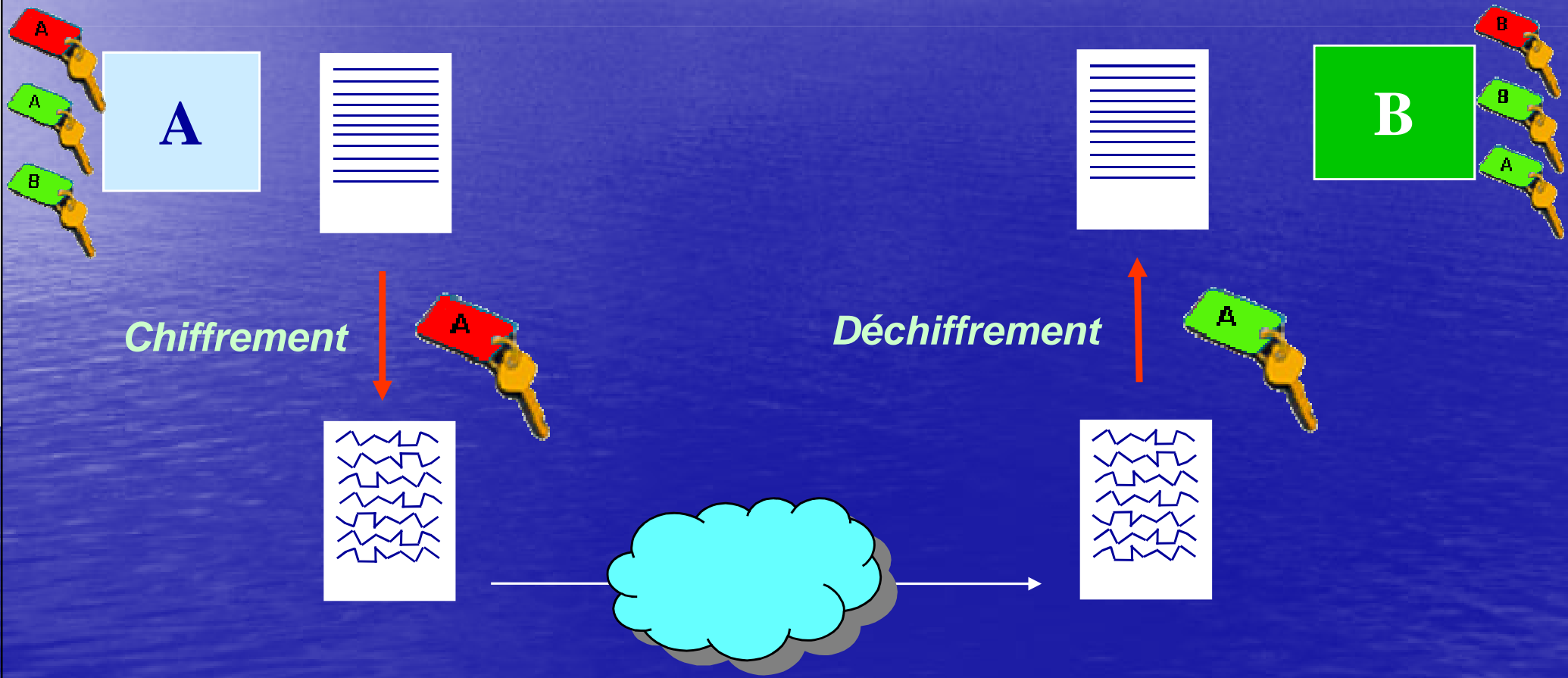


1 & 2. Confidentialité

et Intégrité

- **A veut transmettre un document confidentiel à B**
 - **A chiffre le document avec la clé publique de B**
 - **A émet le document à destination de B**
 - **À réception, B déchiffre le document avec sa clé privée**
 - **Seul B peut exploiter le document ; sans la clé privée de B, le document ne peut être exploité**

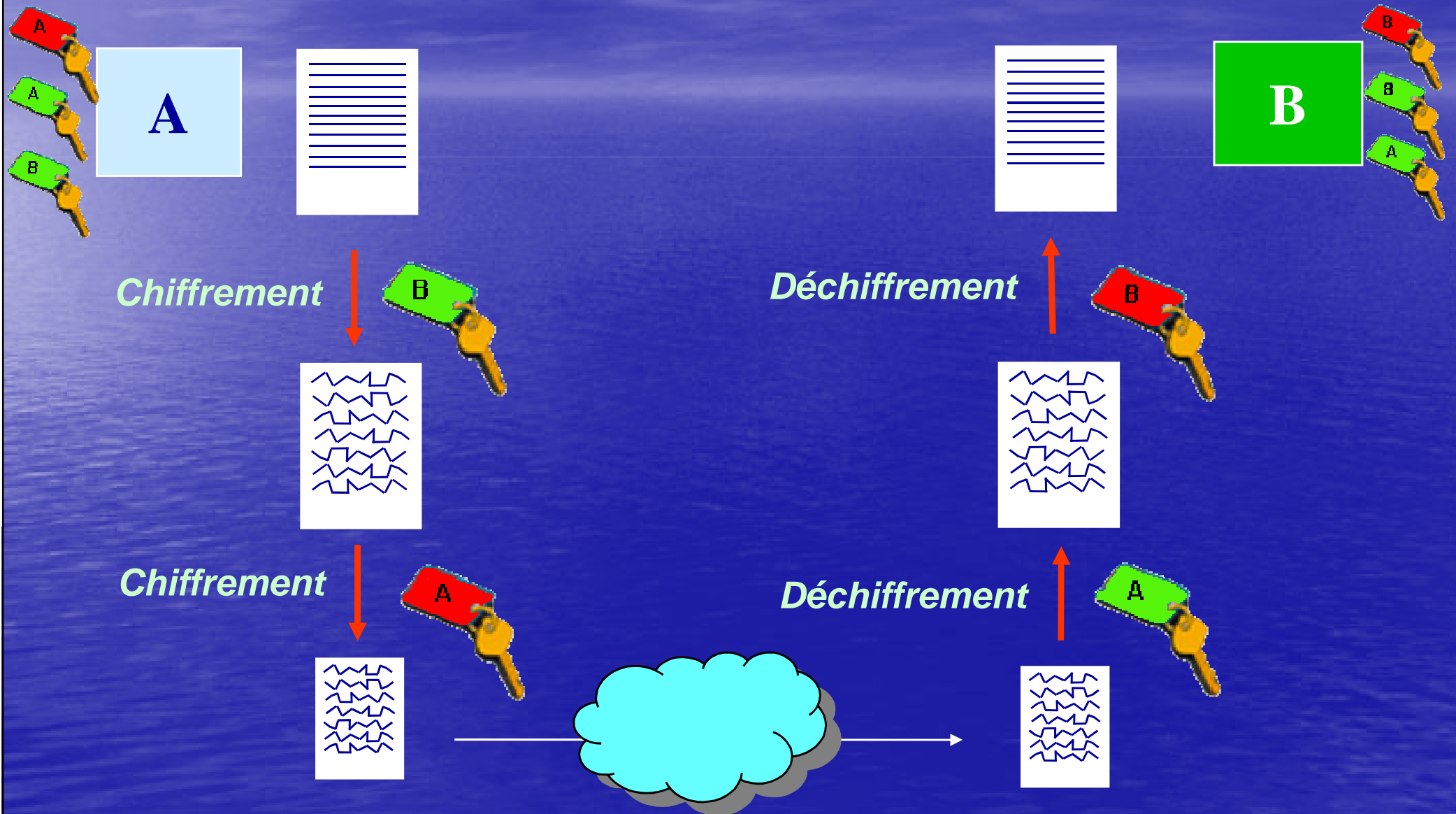
2 & 3 & 4. Authentification et non répudiation



3 & 4. Authentification et non répudiation

- **A veut transmettre un document authentifié à B**
 - **A chiffre le document avec sa clé privée**
 - **A émet le document à destination de B**
 - **B déchiffre le document avec la clé publique de A (*un tiers peut le faire également*)**
 - **Seul A a pu émettre ce document ; le document n'a pu être chiffré qu'avec la clé privée de A et de plus, A ne peut nier l'avoir émis**

1 & 3 . Authentication et confidentialité

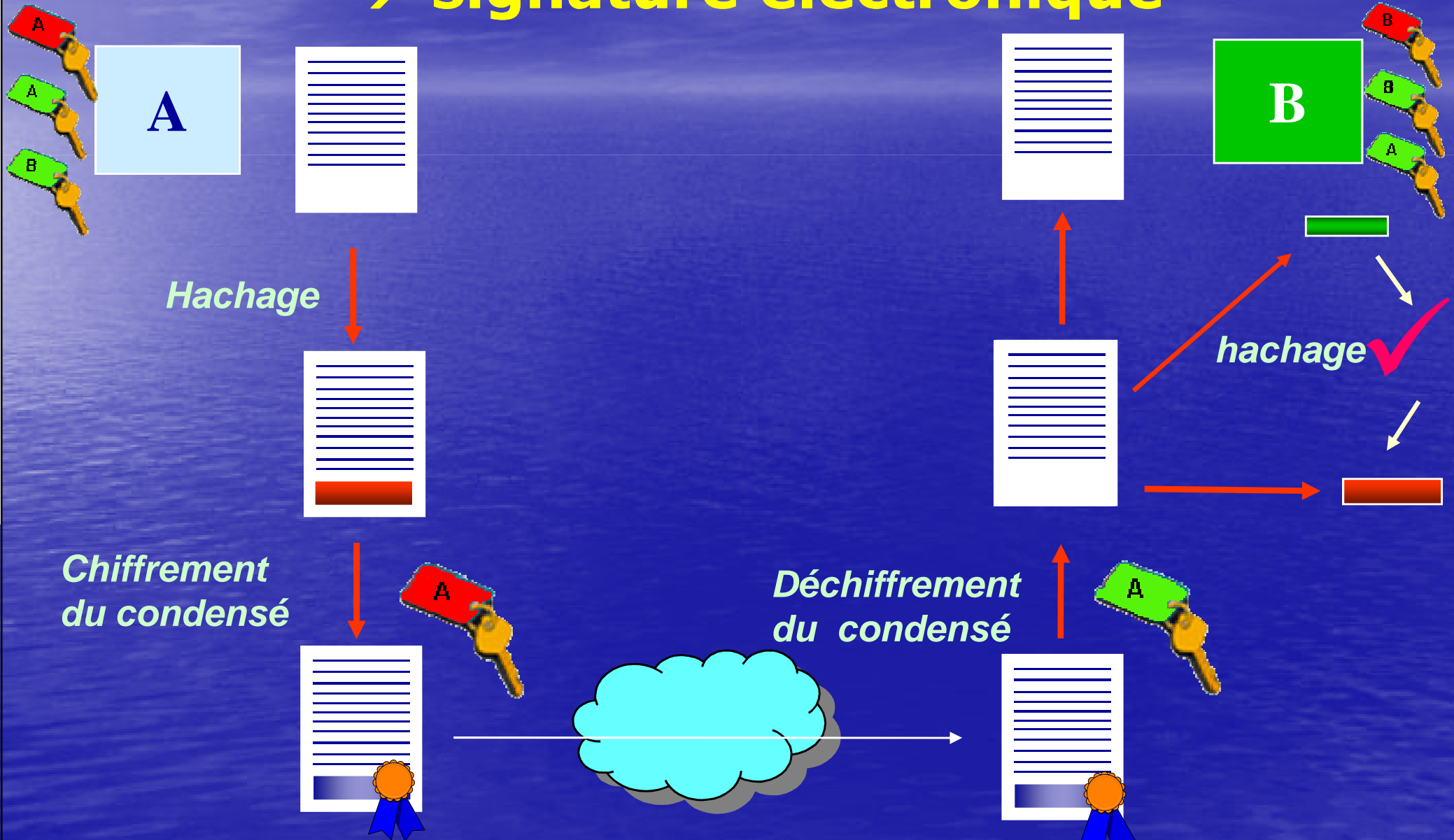


1 & 3 . Authentification et confidentialité

- A veut transmettre un document authentifié et chiffré à B
 - A chiffre successivement le document avec la clé publique de B, puis avec sa clé privée
 - A émet le document doublement chiffré à destination de B
 - À réception, B déchiffre successivement le document avec la clé publique de A, puis avec sa clé privée
 - Seul A a pu émettre ce document, le document n'ayant pu être chiffré qu'avec la clé privée de A
 - Seul B peut exploiter le document, car il a été chiffré avec sa clé publique
 - Le document est intègre si les 2 déchiffrements successifs n'échouent pas

2 & 3 . Authentification et intégrité

→ signature électronique

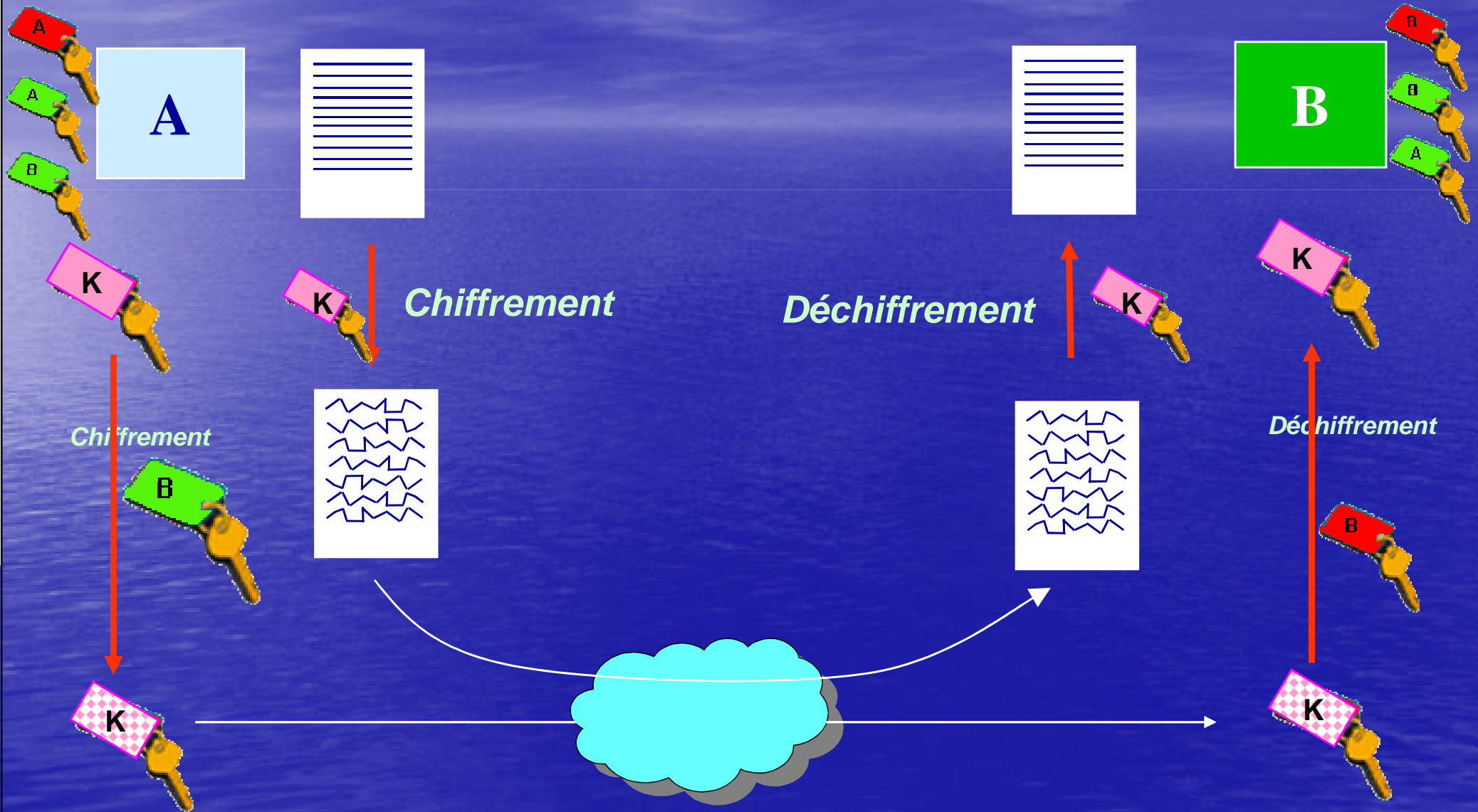


Authentification et intégrité

→ signature électronique

- A veut transmettre un document signé (authentifié et intègre) à B
 - A condense le document et chiffre le condensé avec sa clé privée → empreinte ou signature
 - A émet le document suivi de sa signature à destination de B
 - À réception, B déchiffre l'empreinte avec la clé publique de A, pour extraire le condensé
 - B condense également le document
 - B compare le condensé reçu avec celui obtenu
 - Si identique, le document est intègre et authentifié

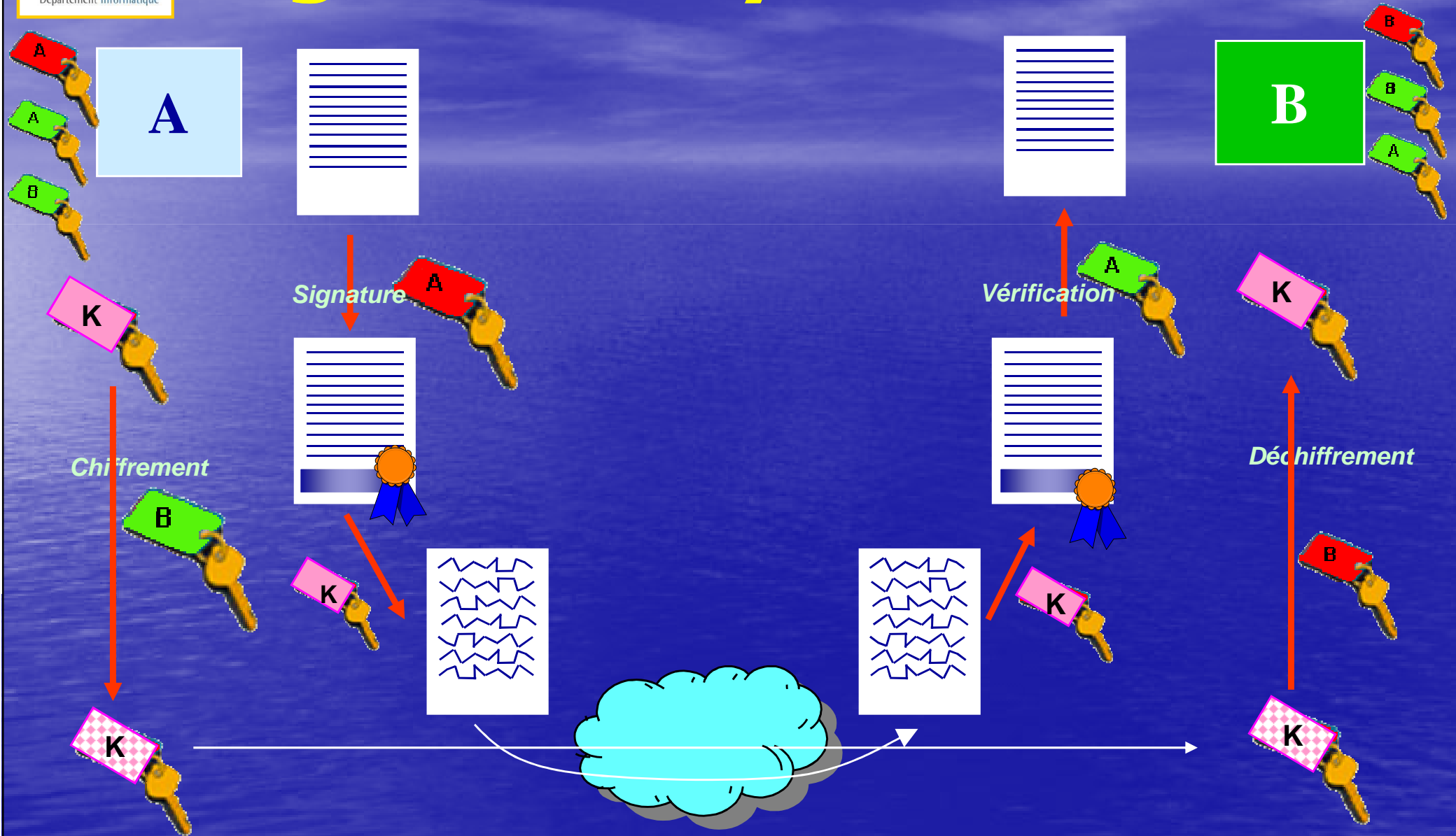
Chiffrement hybride



Chiffrement hybride

- A veut transmettre un document chiffré à B
 - A choisit une clé secrète K
 - A émet la clé secrète chiffrée avec la clé publique de B
 - A émet le message chiffré avec la clé secrète K
 - À réception, B déchiffre la clé secrète K avec sa clé privée
 - B déchiffre le message avec la clé secrète K
- → intérêt : la durée de vie de K est brève, elle peut changer à chaque message

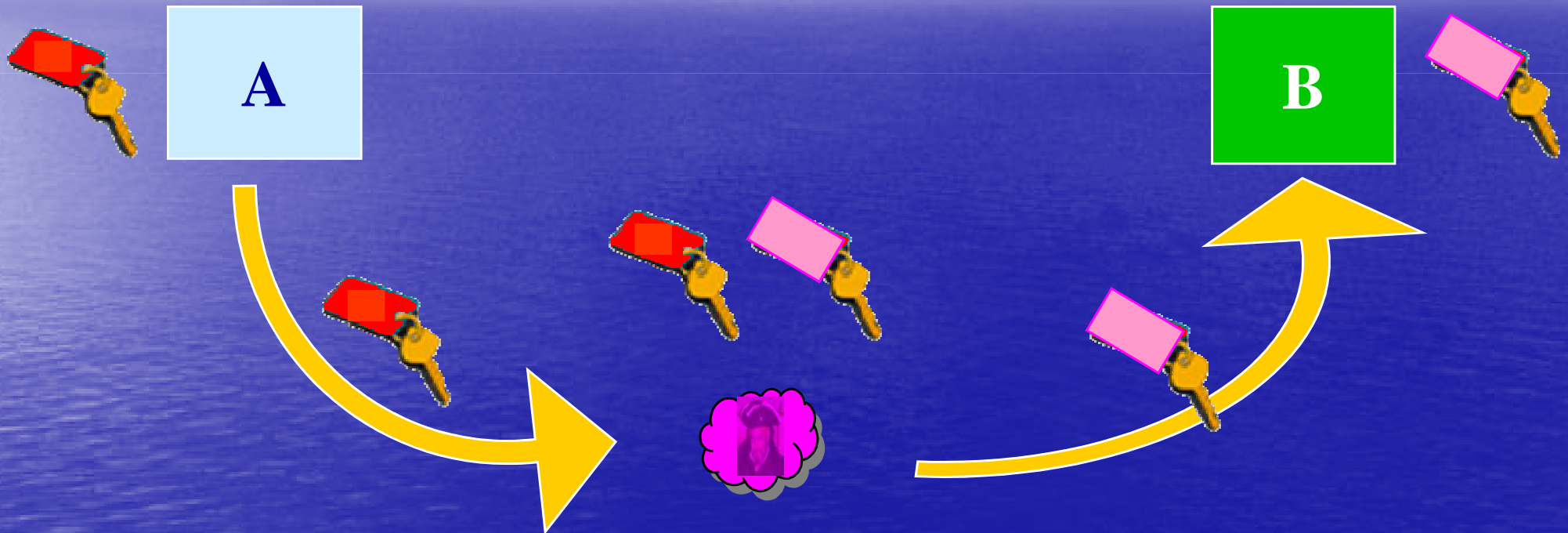
Signature hybride



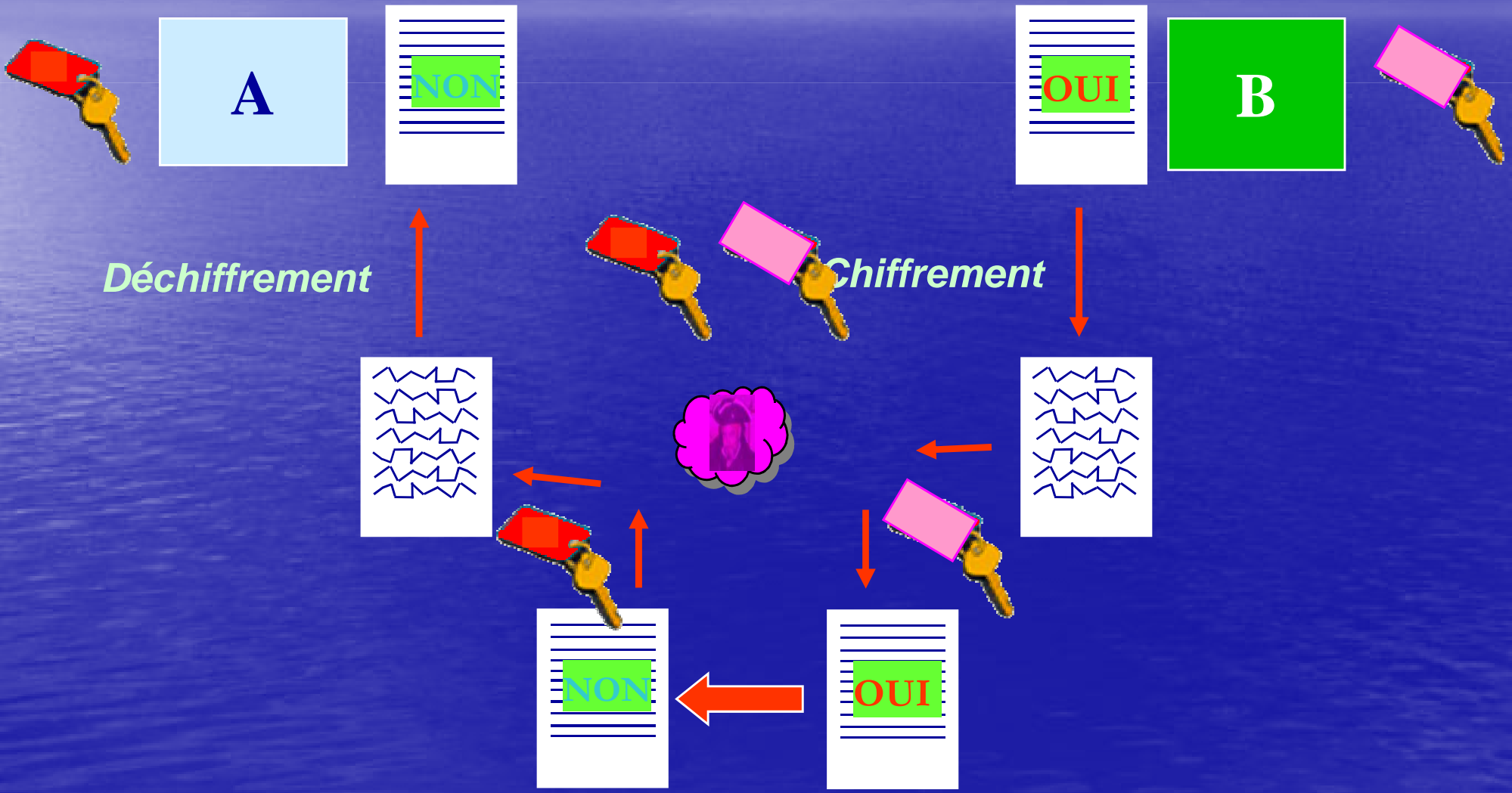
Signature hybride

- A veut transmettre un document signé et chiffré à B
 - A choisit une clé secrète K
 - A émet la clé secrète chiffrée avec la clé publique de B
 - À réception, B déchiffre la clé secrète K avec sa clé privée
 - A condense le document et chiffre le condensé avec sa clé privée → empreinte ou signature
 - A émet le document signé, chiffré avec la clé secrète K
 - À réception, B déchiffre le document avec la clé secrète K
 - B déchiffre l'empreinte avec la clé publique de A, pour extraire le condensé
 - B condense également le document
 - B compare le condensé reçu avec celui obtenu
 - Si identique, le document est confidentiel, intègre et authentifié
- → intérêt : la durée de vie de K est brève, elle peut changer à chaque message

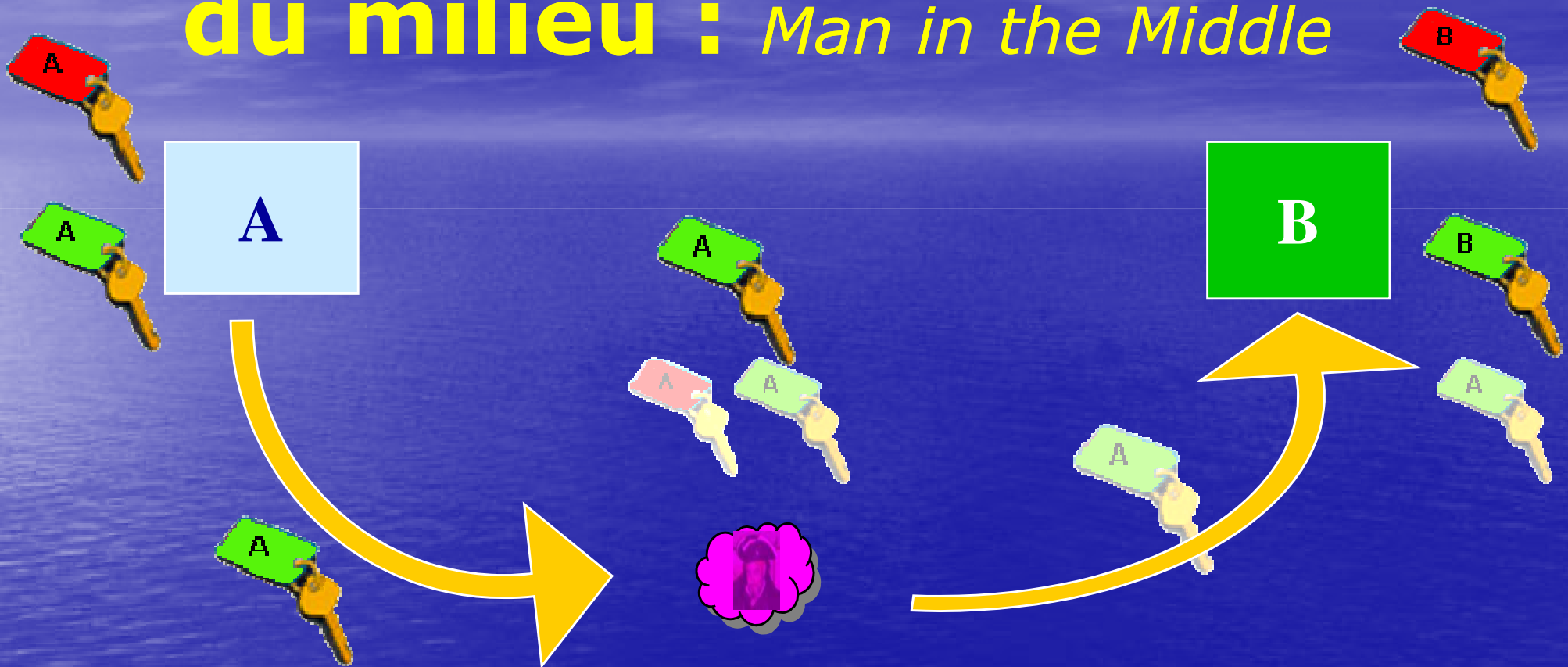
L'attaque de l'Homme du milieu : *Man in the Middle*



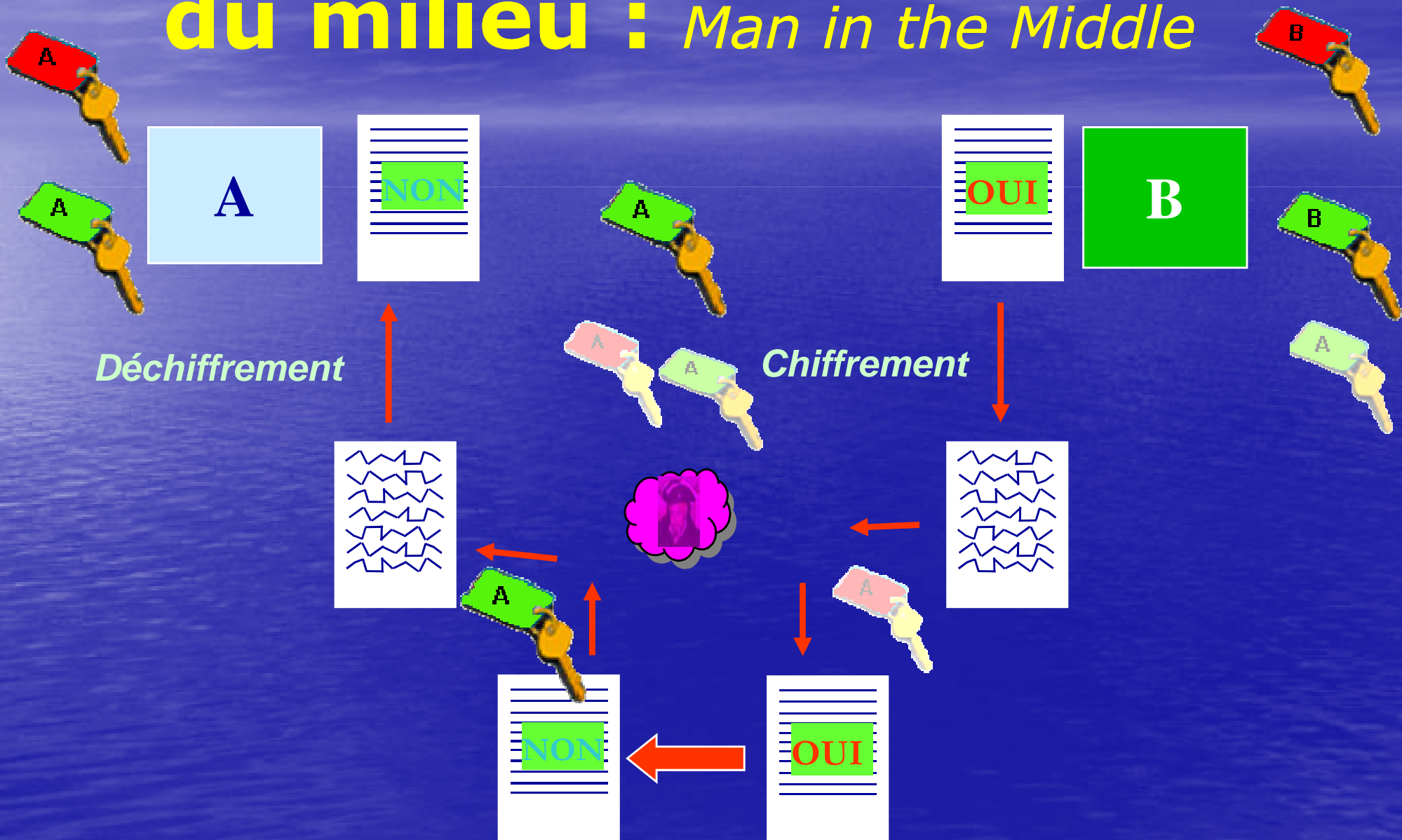
L'attaque de l'Homme du milieu : *Man in the Middle*



L'attaque de l'Homme du milieu : *Man in the Middle*



L'attaque de l'Homme du milieu : *Man in the Middle*



Difficultés liées aux clés

- Mécanismes à clé secrète
 - Difficulté de garantir la confidentialité lors du partage de la clé secrète
 - Ce mécanisme est souvent mis en œuvre en combinaison avec mécanisme à clé publique
- Mécanismes à clé publique
 - Le stockage de la clé privée pose un problème
 - Sécurité relative lorsque le stockage est assuré par le système d'exploitation
 - Cartes à puce pour permettre à l'utilisateur de ne pas «quitter» sa clé
 - Performances des algorithmes
 - Obtention des clés publiques/privées
 - Qui génère et distribue ces clés et qui en garantit l'authenticité ?
 - Que faire en cas de perte de la clé privée ?

Qu'est ce qu'un certificat ?

- Un certificat correspond à l'association :
 - D'une clé publique
 - De l'identité de son propriétaire
 - De l'usage qui peut être fait de la clé
- Cette association est assurée par une autorité de confiance, appelée
Autorité de Certification
ou Tiers de confiance
qui atteste de la véracité des informations contenues dans le certificat

Pourquoi un certificat ?

- Utilisation de certificats
 - Authentification : destiné à l'authentification forte d'entités (personnes, applications, équipements) stocké sur une machine ou une carte à puce
 - Signature de documents : utilisé pour la signature et la vérification de signature de documents
 - Chiffrement : utilisé pour le chiffrement de données et les clés de session
 - Signature de certificats : utilisé pour signer d'autres certificats

Où obtenir

un certificat ?

- Auprès d'une Autorité de Certification
 - Verisign
 - Thawte
 - Certplus
 - Certinomis
- Ces autorités reconnues délivrent, distribuent de manière sécurisée et révoquent les certificats.
- L'autorité de certification est représentée par un certificat racine

Tiers de Confiance et ICP

- ICP Infrastructure à Clé Publique
- PKI Public Key Infrastructure
- Architecture complexe avec
 - AC Autorité de Certification,
 - AE Autorité d'Enregistrement,
 - OC Opérateur de Certification
- U → OC identification et contrôle → AE validation → AC délivre le certificat → AE distribue → U

Certificats X509v3

- Un premier pas de normalisation a permis de définir le format d'un certificat X509v3
- Un utilisateur obtient une clé privée, la stocke et diffuse le certificat correspondant, qui comporte les champs suivants :

Version

Algorithmes utilisés

Nom de l'émetteur

Organisation

Adresse e-mail

Clé publique

Numéro de série

Validité

Période de Validité

Nom

Adresse

Signature de l'autorité de certification émettrice

La publication du certificat

- Une fois le certificat obtenu, la clé privée est stockée sur la machine du propriétaire et son certificat (identification + clé publique) doit être publié :
 - Par envoi d'un message signé
 - Par import sous forme de fichier PKCS#12
 - Par consultation du site web de l'AC
<https://digitalid.verisign.com/services/client/index.html>
 - Par consultation d'annuaires : X500, LDAP, Active Directory, DNS
 - IKE , mécanismes d'échange de certificats

La validation du certificat

- Avant d'être utilisée dans une application, le certificat doit être vérifié :
 - La **date de validité** est-elle correcte ? Le certificat n'a-t-il pas expiré ?
 - Quelle **confiance** accorde-t on à l'autorité de certification ayant signé le certificat ?
 - La **signature** de l'autorité de certification est-elle valable ?
 - Le certificat fait-il parti d'une liste de **révocation** ?
 - **Condition d'utilisation** du certificat

- Certificat **serveur**
 - Un certificat serveur est un certificat dont l'identité est associée à un service. Il permet à un client Web de s'assurer que le service concerné est bien celui qu'il prétend être
- Certificat **client**
 - Un certificat client est un certificat associé à un utilisateur final (identité de l'utilisateur ou adresse mail). Il permet au correspondant de s'assurer de l'authenticité de l'utilisateur et peut remplacer le *user/password*
- Certificat de **code**
 - Un certificat de code est utilisé afin de signer numériquement les exécutables. Il permet de fournir une garantie sur l'authenticité et l'intégrité des fichiers téléchargés (applets, ActiveX...)

Difficultés des PKIs

- Absence de standards de certification, pas de normalisation internationale
- Gestion permanente des clés à grande échelle
- Identification de l'utilisateur final (adresse email insuffisante)
- Validation, mise à jour et révocation des certificats dans les annuaires
- Contrôle de durée de validité et révocation
- Sécurité du stockage des clés

"Le plus compliqué, c'est d'organiser la confiance et la maintenir vingt-quatre heures sur vingt-quatre"

Richard Pirim, architecte système sécurité et réseau de la direction générale des impôts

Autres certificats ?

- Certificat auto-signé
 - l'utilisateur crée lui même et signe son propre certificat (*par ex. avec les outils openssl sous Unix*) auquel on peut accorder sa confiance ou non.
- PGP Pretty Good Privacy (Phil Zimmermann)
 - Outils et plug-ins disponibles gratuitement
 - Notion de réseau de confiance par chaînage
 - Chaque utilisateur crée et distribue sa clé
 - A envoie sa clé à B qui la valide
 - B envoie sa clé (et celle de A) à C qui la valide (si C accorde sa confiance à B, il l'accorde aussi à A)

Exemple de mise en œuvre de certificats : la messagerie

- La messagerie a été le premier service déployé à très grande échelle dans l'Internet
- Elle utilise toujours le protocole SMTP *SIMPLE MAIL TRANSFER PROTOCOL* RFC821 (1982)
- Le service ne permet aucune authentification, ne permet aucun contrôle, et les messages circulent en clair dans le réseau

La messagerie le problème

- Les messages ne peuvent contenir que des caractères ASCII sur 7 bits
- Afin de pouvoir attacher des fichiers binaires aux messages, le protocole a été étendu avec le format *MIME Multipurpose Internet Mail Extensions* (1996)
- MIME permet le codage de fichiers binaires quelconques (doc, xls, exe, zip, mp3, wav, mpg, ...) sous forme de suite de caractères ASCII sur 7 bits
- Toutes les informations **EN CLAIR** peuvent facilement être décodées et exploitées

Le message

```
Date: Sat, 18 Sep 2003 09:48:05 +0200
From: Robert.Nard@urs.u-strasbg.fr (NARD Robert)
Message-Id: <200310180748.h9I7m5h12438@urs.u-strasbg.fr>
To: Dominique.Grad@urs.u-strasbg.fr
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="-
    MQQ106646328578ef2ea4da92d58b0b5545b804fb6845"
X-Antivirus: scanned by sophos at u-strasbg.fr
```

En-tête

This message is in MIME format.

```
---MQQ106646328578ef2ea4da92d58b0b5545b804fb6845
Content-Type: text/plain
Content-Transfer-Encoding: 8bit
```

Corps du message

ci joint le compte rendu de la réunion de jeudi
Robert

Pièce
attachée
codée en
base64

```
---MQQ106646328578ef2ea4da92d58b0b5545b804fb6845
Content-Type: application/msword; name="Réunion du 16 octobre.doc"
Content-Transfer-Encoding: base64
Content-Disposition: inline; filename="Réunion du 16 octobre.doc"
```

```
0M8R4KGxGuEAAAAAAAAAAAAAAAAAAAAAAAAAPgADAP7/CQAGAAAAAAAAAAAAAAAAABAAAALgAAAAAA
piaUI94j0AAAAAAAAAAAAAAAAAAAAAAAAAMBBYALh4AAIBXAACAVwAAxgsAAAAAAAAArAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAD//w8AAAAAAAAAAD//w8AAAAAAAAAAD//w8AAAAAAAAAAAA
```

La messagerie la solution

- Le protocole **S/MIME** *Secure/MIME* est une extension du protocole MIME qui permet de sécuriser les messageries en assurant
 - Confidentialité
 - Intégrité
 - Authentification
 - Non-répudiation
- S/MIME va permettre de signer et/ou chiffrer les messages en utilisant les algorithmes de hachage, chiffrement et certificat X509
- S/MIME est supporté par les clients de messagerie Microsoft Outlook Express et Netscape Messenger

TP1 : La messagerie sécurisée

But du TP : Obtenir et utiliser un certificat lors d'échanges de messagerie sécurisée

