Cryptographie

Notions de cryptographie

Sommaire

Au menu :

Introduction

Cryptographie à clé secrète

Cryptographie

Partie de la cryptologie s'attachant à rendre un message incompréhensible sauf pour son destinataire légitime.
S'oppose à la cryptanalyse.

- **Clé** : un secret
- **Chiffrer** : production d'un message chiffré à partir du message clair et d'une clé
- **Déchiffrer** : récupération du message clair à partir du message chiffré à l'aide de la clé
- **Décrypter** : récupération du message clair sans la clé de déchiffrement

Cryptologie

- Commence dans l'antiquité
- Lutte incessante cryptographie vs. cryptanalyse
- Basculement progressif de cryptosystèmes secrets => cryptosystèmes publics à clé secrète (Principe de Kerckhoffs)
- Automatisation des procédés (Ex : Enigma)
- Cryptographie moderne (~1950)
  - Cryptographie mathématique (Shannon)
  - Passage au numérique (Informations binaires)
  - Utilisation de l'ordinateur

Objectifs cryptographiques

Les principaux :

- **Confidentialité**
- **Authenticité**
- **Intégrité**

Mais aussi :

- Non répudiation/Déni plausible
- Confidentialité persistante (Forward Secrecy)
- Protection contre rejeu
- Anonymat
- Preuve à divulgation nulle de connaissance
- etc.

Cryptographie moderne

Cryptographie symétrique => Algorithmes à clé secrète

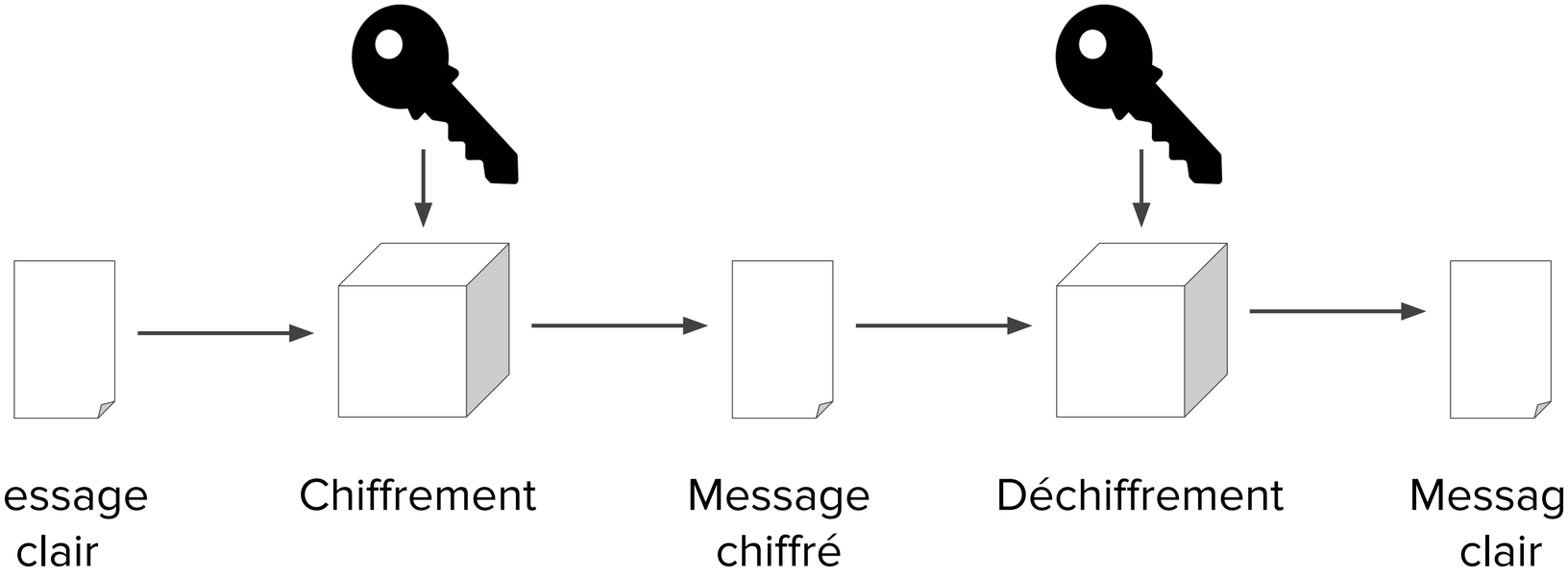
Cryptographie asymétrique => Alg. à clé publique et clé privée

Fonctions de hachage => Calcul d'empreinte/condensat/hash

Cryptographie symétrique

Cryptographie à clé secrète

Cryptographie symétrique



2 fonctions - 1 seule clé

Chiffrement
Entrée : Message clair + clé
Sortie : Message chiffré

Déchiffrement
Entrée : Message chiffré + clé
Sortie : Message clair d'origine

Que chiffre-t-on ?

Messages clairs :

- N'importe quelle séquence binaire (fichier, paquet réseau, tout...)
- **Chiffrement par blocs**
- **Chiffrement de flux**

Quel est le résultat ?

Messages chiffrés :

- Séquence binaire (taille équivalente)
- Ressemble à une séquence aléatoire
  - Confusion : lien clé ⟺ chiffré complexe
- Bonne diffusion :
  - 2 messages clairs proches engendrent des messages chiffrés très différents

Les clés

Une clé symétrique est :

- Séquence binaire d'une taille donnée
- **Aléatoire** : Imprévisible, impossible à deviner
- **Longue** : Rendre la force brute trop coûteuse
- **Secrete** : Partagée le moins possible (une clé par paire de correspondants)

3 problèmes :

- Générer les clés - Générateurs pseudo aléatoires
- Stockage sécurisé des clés - HSM
- Partage de clé

Chiffrement par blocs

Gestion de messages de taille quelconque par blocs de taille fixe

- Bourrage (*padding*) : Combler jusqu'à la *bonne* taille
- Modes opératoires - Lien entre les blocs :
  - ECB (*Electronic Cypher Block*) : Blocs chiffrés indépendamment - Non recommandé
  - CBC (*Cypher Block Chaining*) : Bloc chiffré dépend du bloc clair et du bloc chiffré précédent
  - Autres modes : CTR, CFB, OFB, CTS...

Vecteur d'initialisation (Bloc précédent le 1ère bloc) :

- Aléatoire mais pas secret (transmis/stocké avec le chiffré)
- Jamais réutilisé

Chiffrement de flux

Chiffrement du message clair bit par bit

- En général par XOR
- Nécessite un générateur pseudo aléatoire initialisé par :
  - La clé
  - Un nonce : nombre arbitraire utilisé une seule fois
- Même clé + même nonce = même séquence

Algorithmes symétriques

Chiffrement par bloc :

- **AES** (*Advanced Encryption Standard*) - Rijndael Daemen & Rijmen (clés de 128/196/256 bits)
- Également : Twofish, Serpent, MARS, RC6, ...
- Obsolète : Blowfish, DES, 3DES

Chiffrement par flux :

- **Chacha20** - Daniel J. Bernstein (Clé de 256 bits)
- Obsolète : RC4 (Arcfour)

Fonctions de dérivation de clés

Fonction de dérivation de clés :

- Calcul de clé(s) à partir d'un secret maître

Ce secret maître peut-être un mot de passe

- Ex. : PBKDF2 (Password-Based Key Derivation Function) - RFC 8018

Évidemment suppose l'utilisation de mots de passe (phrases de passe) robustes !

En pratique

Chiffrement de messages :

- Clés partagées
- Protocole d'échange de clés de **Diffie-Hellman**

=> Clés de session

- Clé pour une (partie) de communication
- Rapide (trafic réseau) et Sûr (jusqu'à preuve du contraire)

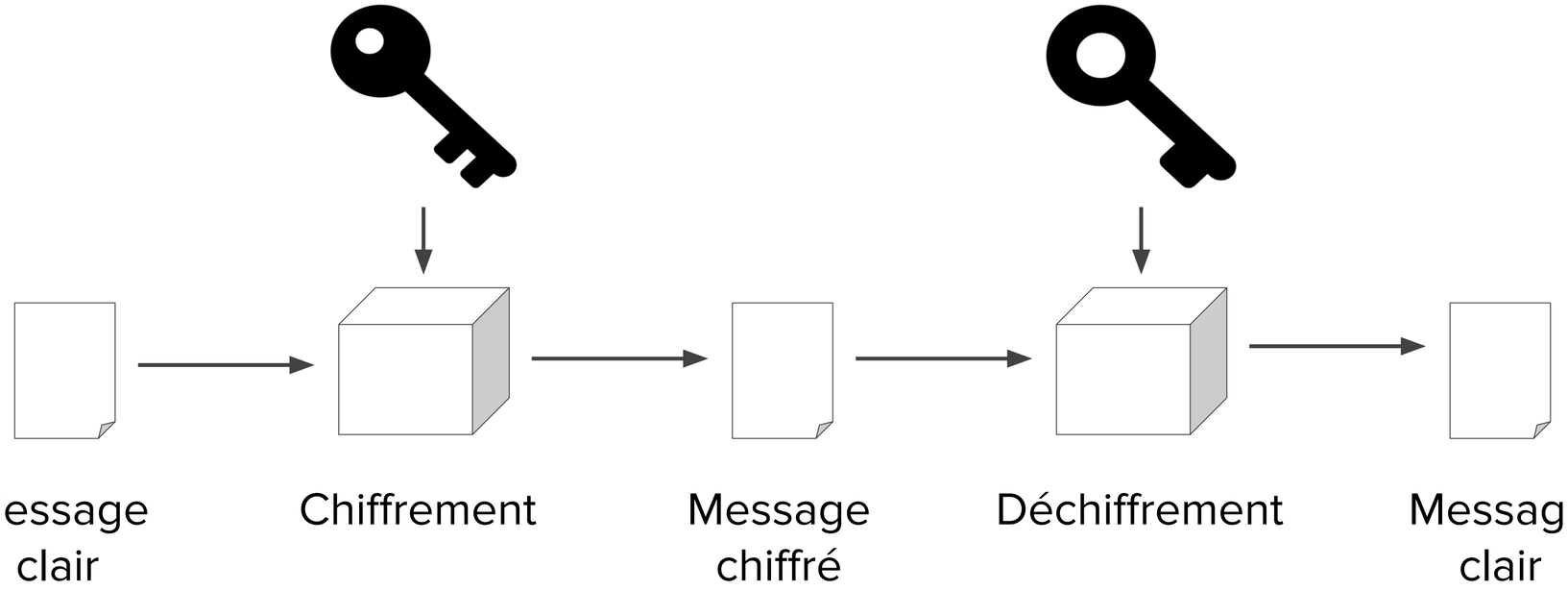
Chiffrement de fichiers :

- Coffre-fort logiciel

Cryptographie asymétrique

Cryptographie à clé publique

Cryptographie asymétrique



Clés générées par paire :

- 1 clé de chiffrement
- 1 clé de déchiffrement

Concept attribué à Whitfield Diffie et Martin Hellman

Les clés asymétriques

Une paire de clés asymétriques est :

- Construction mathématique impliquant de l'aléatoire
- Mais pas complètement aléatoire (ex. :produits de grands nombres premiers aléatoires)

Clé **publique** / Clé **privée** :

- Clé publique : distribuée aux correspondants (pas secrete)
- Clé privée : conservée secrètement (jamais partagée !)
- Une paire de clé par personne

Les clés asymétriques (suite)

Problème :

- Algorithmes lents
- Taille clé ≠ Puissance de chiffrement
- Vulnérabilité théorique => Cryptographie post-quantique

Les objectifs

Confidentialité : Publication de la clé de chiffrement

- Tout le monde peut chiffrer
- Seul le propriétaire de la clé peut déchiffrer
- Unidirectionnel => envoi au propriétaire de la clé

Authentification : Publication de la clé de déchiffrement

- Seul le propriétaire peut chiffrer
- Tout le monde peut déchiffrer (plus de confidentialité)
- Authentification de l'expéditeur : Signature numérique

Algorithmes asymétrique

RSA - Ronald Rivest, Adi Shamir et Leonard Adleman

- Clés de 1024 à 4096 bits (>= 3072 bits recommandé)
- Basé sur la difficulté de la décomposition en facteurs premiers

Cryptographie sur les courbes elliptiques (ECC)

- Clés de 256 à 512 bits
- Plusieurs courbes (Curve25519, Ed25519, NIST P-256...)

Cryptosystème de ElGamal

Sécurité et performance

Cryptographie asymétrique est :

- Sécurisée mais avec des clés plus longues
- Plus lente (chiffrement et déchiffrement coûteux)

Mais avec un calculateur quantique

- Il existe des résultats théoriques (Shor, Grover...)
- Casse les algorithmes asymétriques classiques
- Cryptographie post quantique

En pratique

Cryptographie hybride : symétrique + asymétrique :

- Avec la cryptographie symétrique : performance de chiffrement
  - Nécessite de partager une clé de manière sûre
- Avec la cryptographie asymétrique : permet d'échanger une clé de façon sécurisée => résout le problème du partage de clé

Clé de session : clé symétrique à usage unique, limité dans le tps.

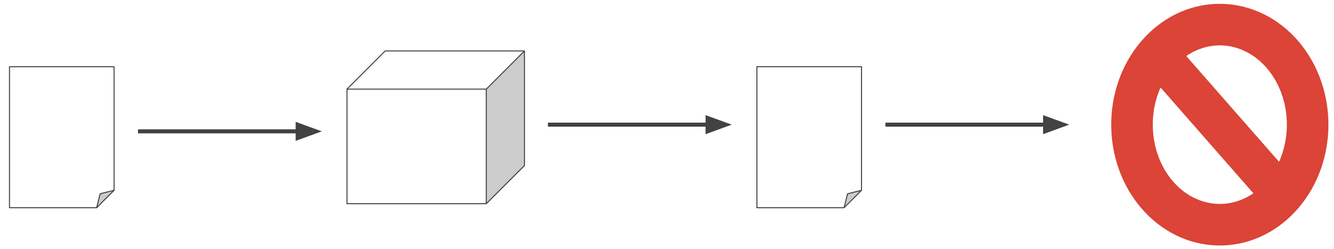
Échange de clés de Diffie-Hellman

- Permet la construction d'un secret partagé
- Communication possible même sur canal non sécurisé
- DH *classique* (logarithme discret) ou ECDH (courbes elliptiques)

Fonctions de hachage

Fonctions à sens unique

Fonctions de hachage



Entrée : Message quelconque
Sortie : Empreinte / Condensat / Haché (*hash*)

Calcul d'empreintes

En entrée :

- Séquence binaire de taille quelconque

En sortie :

- Empreinte de taille fixe et petite

Des propriétés particulières :

- Fonction à sens unique
- Calcul message ➜ empreinte : simple
- Calcul empreinte ➜ message : impossible (en pratique)

Quelques fonctions de hachage

SHA-2 (Secure Hash Algorithm 2) : Empreinte de 224, 256, 384 ou 512 bits.
Son remplaçant : SHA-3 (Keccak)
D'autres : Argon2, Bcrypt

Mécanismes obsolètes : MD5, SHA-1, RIPEMD-160, Whirlpool

Concept de collision :
2 messages différents produisent la même empreinte.

En pratique

Pister des empreintes

Contrôle d'intégrité de messages :

- Calcul de l'empreinte du message à envoyer
- Transmission message + empreinte
- À réception : calcul de l'empreinte du message reçu
  - Identique : message non modifié (erreur)
  - Différent : message modifié

Vérification de l'intégrité des fichiers :

- Comparer sans dupliquer/transmettre
- Calcul d'empreinte de 2 fichiers
  - Empreintes identique = Fichiers identiques

Empreintes de mots de passe

Pister des empreintes

Bonne pratique : stocker les empreintes et non les mots de passe

- Variante de l'attaque par dictionnaire : Rainbow table

Dans le cas des mots de passe :

- Salage : Mots de passe identiques => empreintes différentes
- Augmentation du coût : paramétrage du nombres d'itérations

Fonctions particulières :

- Bcrypt basée sur Blowfish
- Argon2 : vainqueur de la Password Hashing Competition
- Yescrypt : utilisée par défaut sur Debian

Authentification cryptographique

Utiliser la cryptographie pour vérifier à qui on parle

HMAC

Hachage + secret partagé

Keyed-hash message authentication code

- Code d'authentification de message (MAC)
- Basé sur un secret partagé (clé) + fonction de hachage
- Assure authentification et contrôle d'intégrité

Envoi :

- Mélange message + clé
- Calcul de l'empreinte (HMAC)
- Transmission message + HMAC

Réception :

- Mélange message reçu + clé
- Calcul de l'empreinte
- Comparaison avec HMAC reçu

Chiffrement authentifié

Cryptographie symétrique avec authentification

Objectifs : Confidentialité + Authentification + Intégrité

Mix : chiffrement symétrique + HMAC

Chiffrement authentifié (AE) + Associated Data (AD) - Intégrité

- Encrypt-then-MAC (EtM)
- Encrypt-and-MAC (E&M)
- MAC-then-Encrypt (MtE)
- CBC-MAC
- Galois/Counter Mode (GCM)
- Counter with CBC-MAC (CCM)

Note : clés différentes pour authentification et confidentialité

Authentification asymétrique

Hachage + cryptographie asymétrique

Signature numérique

- Clé publique de déchiffrement + fonction de hachage
- Assure authentification et contrôle d'intégrité

Envoi :

- Calcul empreinte message
- Chiffrement de l'empreinte (Signature)
- Transmission message + signature

Réception :

- Calcul empreinte message
- Déchiffrement de la signature
- Comparaison

Mécanismes de signature

Quelques candidats

RSA - Mécanisme de signatures basé sur RSA

ECDSA (Elliptic Curve Digital Signature Algorithm) - Courbes NIST

EdDSA (Edwards-curve Digital Signature Algorithm) - Ed25519

Mécanisme obsolète : DSA

Les certificats

Le passport numérique

Limite des signatures

Association clé publique <-> identité

Signatures numériques permettent de vérifier que l'expéditeur possède bien la clé privée correspondant à la clé publique que je possède.

Besoin : authentifier une personne / un hôte

Comment s'assurer que la clé publique correspond à la bonne identité ?

Les certificats

Le principe

Certificat éléctronique :

- Des clés publiques
- Des informations d'identification (nom)
- Dates de validités et autres métadonnées
- Signatures numérique de Tiers de confiance (certificateur)

2 formats :

- X.509
- OpenPGP

Créer un certificat

Faire sa demande

La demande Certificate Signing Request (CSR)

- Créer une/des paire(s) de clé(s)
- Conserver précieusement les clés privées
- CSR : information d'identité + clés publiques + métadonnées

La certification

- Envoi à un tiers de confiance
- Satisfaire à sa procédure de vérification d'identité
- Récupérer le certificat signé par le tiers.

Vérifier

Faire sa demande

À réception d'un certificat inconnu

- Vérifier que l'identité présente dans le certificat correspond
- Vérifier la signature présente dans le certificat
- => nécessite la clé publique associée au tiers qui a signé
- Récupération du certificat du tiers de confiance...
  - Chaine de certification

Problème circulaire

- Des certificats de départ (racines) sont nécessaires !

Tiers de confiance

Qui certifie

Tiers de confiance : entité qui s'occupe pour nous de faire les vérifications d'identité et qui signe des certificats

Approche X.509 : Autorité de certification
- Organisme supposé connu de tous et de confiance
- Souvent : entreprises spécialisées

Approche OpenPGP : Toile de confiance
- Approche décentralisée : tout le monde peut certifier

Révocation

Parce que tout ne se passe pas toujours comme prévu

En cas de perte/compromission de clés privées

- Révocation de certificat

Diffusion du certificat révoqué :

- Liste de révocations (CRL) gérée par les CA
- Vérification en direct (OCSP Online Certificate Status Protocol)
- Agrafage OCSP

X.509

Parce que tout ne se passe pas toujours comme prévu

Norme UIT pour certificats numérique

- Formats de certificats
- Principe d'autorité de certification
- Suppose des certificats racines connus (auto-signé)

Un exemple d'autorité de certification : let's encrypt

Public Key Infrastructure

Infrastructure d'émission de certificats

Infrastructure à clés publiques (PKI)

- Ensemble d'éléments permettant d'émettre des certificats
- Matériel - éventuellement Hardware Security Module
- Logiciel
- Procédure

Mise en place d'une autorité de certification

ACME

L'approche let's encrypt

Protocole d'automatisation de certification

- Automated Certificate Management Environment
- Facilite le déploiement de certificats (notamment HTTPS)
- Permet d'utiliser des durées de validité courtes
- Évite les problèmes de renouvellement

Implémentation : certbot

OpenPGP

L'alternative

Protocole IETF (RFC 4880)

- Version ouverte du PGP (Pretty Good Privacy) de Philip Zimmermann
- Concept de Toile de confiance (Web of Trust)
  - Serveurs de clés (distributions)
  - Key signing party (démarrage)

Implémentation : GNU Privacy Guard (GPG)

En pratique

Les certificats dans la vraie vie

Web - HTTPS

- Protocole TLS
- Confidentialité - Authentification (du serveur) - Intégrité

Email

- S/MIME Secure/Multipurpose Internet Mail Extensions (X.509)
- PGP/MIME : OpenPGP

VPN

- OpenVPN (X.509)
- IPsec

...

MERCI

Des questions ?
Des remarques ?