



Application de la cryptographie & cryptosystèmes



par Stéphane LARCHER

Introduction & rappels

Algorithmes symétriques

AES (moderne, blocs de 128 bits, clés de 128/192/256 bits), DES/3DES (plus anciens).

Algorithmes asymétriques et fonctions de hachage

- **Algorithmes asymétriques** : RSA (basé sur la difficulté de factoriser de grands nombres), Diffie-Hellman (échange de clé), ECC (courbes elliptiques, plus efficaces pour de petites tailles de clé).
- **Fonctions de hachage** : SHA-2, SHA-3 (ex : SHA-256, SHA-512), MD5/SHA-1 (considérées obsolètes pour de nouvelles applications).
- **Mécanismes de signature** : Signature RSA, DSA, ECDSA, etc., qui permettent d'assurer l'authenticité et l'intégrité d'un document ou d'un message.

Principes de base de la sécurité informatique

- **Confidentialité** : S'assurer que seules les personnes autorisées peuvent accéder à l'information.
- **Intégrité** : Garantir que l'information n'a pas été altérée.
- **Authentification** : Vérifier l'identité d'une entité (personne, serveur).
- **Non-répudiation** : Empêcher une partie de nier avoir envoyé ou signé un document.

Objectifs de cette séance



Infrastructures à Clés Publiques (PKI)

Comprendre le rôle d'une PKI pour distribuer des certificats et instaurer un climat de confiance.



Protocoles sécurisés

Découvrir le fonctionnement de SSL/TLS, SSH, IPsec.



Sécurité des e-mails et du web

S/MIME, PGP, HTTPS.



Cryptographie quantique

Menaces potentielles et contre-mesures post-quantiques.



Mise en pratique

Installer une mini-PKI de test, générer des certificats, les manipuler avec OpenSSL.

Infrastructures à clés publiques (PKI)



Infrastructures à clés publiques (PKI)

Une PKI (Public Key Infrastructure) est un ensemble d'**organismes** et de **procédures** qui permettent de gérer les **clés publiques** et leur associer des **certificats**. Elle garantit :

- 1 L'identité des entités (individus, serveurs, services).
- 2 La mise en place d'un niveau de **confiance** validé par des autorités reconnues.

Les Autorités de certification (CA) ont pour **rôle principal** de « Signer » (valider) les certificats, c'est-à-dire attester qu'une clé publique correspond bien à une entité précise. Exemples : Let's Encrypt (gratuit, automatisé), GlobalSign, DigiCert, etc.



Gestion du cycle de vie des certificats



Génération de la paire de clés

Par l'entité elle-même ou la CA.



Création et signature du certificat

L'entité envoie une CSR (Certificate Signing Request) à la CA, qui vérifie l'identité et signe le certificat.



Distribution & utilisation

Le certificat est installé (ex. sur un serveur web).



Expiration, renouvellement, révocation

Les certificats ont une date de validité. En cas de compromis ou d'informations invalides, le certificat peut être révoqué (CRL – Certificate Revocation List, ou OCSP – Online Certificate Status Protocol).



Formats de certificats et standards

PKCS (Public-Key Cryptography Standards)

- **PKCS#7** : Format pour la signature et le chiffrement de données (certificats, chaînes de certification).
- **PKCS#10** : Format d'une CSR (utilisé pour demander la signature d'un certificat).
- **PKCS#12** : Format pour le stockage sécurisé de clés privées et de certificats (fichiers .p12 ou .pfx).

PEM (Privacy-Enhanced Mail)

- Format texte (ASCII Base64) encadré par -----BEGIN CERTIFICATE----- / -----END CERTIFICATE-----.
- Couramment utilisé pour les certificats SSL/TLS (Apache, Nginx, etc.).

Démonstrations / mini-exemples

- **Génération d'un CSR via OpenSSL** :
`openssl req -new -newkey rsa:2048 -nodes -keyout server.key -out server.csr`
- **Signature du certificat** par une mini-CA ou par une autorité publique.
- **Vérification** : `openssl x509 -in server.crt -text -noout`
- **Chaîne de certificats** : Schéma d'une root CA → CA intermédiaire → certificat serveur.

Protocoles sécurisés

SSL/TLS



Historique : SSL développé par Netscape, puis TLS (1.0 → 1.3). Versions anciennes obsolètes.

Fonctionnement : Handshake (échange d'informations cryptographiques), établissement de clé de session, authentification.

Points importants : Négociation des versions, vérification du nom de domaine, révocation.

Protocoles sécurisés

IPsec



Fonctionne au niveau réseau (couche 3),
contrairement à TLS (couche 4 / application).

Modes : Transport (chiffre seulement la charge
utile IP), Tunnel (encapsule tout le paquet IP).

ESP (Encapsulating Security Payload) et AH
(Authentication Header) :

ESP : assure confidentialité et/ou
authentification,

AH : assure uniquement l'authentification et
l'intégrité, pas la confidentialité.

Protocoles sécurisés

SSH (Secure Shell)

Authentification : Par mot de passe ou par **clé publique** (recommandé).

Établissement d'un canal chiffré : Pour exécuter des commandes sur un serveur distant (shell), ou pour transférer des fichiers (SCP, SFTP).

Attaque Man-in-the-middle : Risque si l'empreinte du serveur n'est pas vérifiée.



Sécurité des e-mails et du web

Sécurité des e-mails

S/MIME

- **Fondé sur** des certificats X.509.
- Permet de signer et/ou chiffrer les e-mails.
- Intégré dans des solutions telles que Microsoft Outlook, Apple Mail, etc.
- Nécessite une **PKI** (l'utilisateur doit obtenir un certificat personnel émis par une CA).

PGP / GnuPG

- **Modèle de "toile de confiance"** (web of trust) : Pas de hiérarchie centralisée ; chaque utilisateur peut signer la clé publique d'un autre.
- **Chiffrement asymétrique** (RSA, ECC) + fonctions de hachage pour la signature (SHA-2...).
- **Usage** : Principalement pour les e-mails chiffrés entre particuliers, organisations spécialisées ou communautés tech.

Sécurité des e-mails et du web

Sécurité du web : HTTPS

HTTPS = HTTP + TLS

- Permet la **confidentialité** (chiffrement) et l'**authentification** du serveur (via son certificat SSL/TLS).

Enjeux :

- Sécuriser les transactions bancaires (confidentialité),
- Éviter l'usurpation de site (authenticité),
- Protéger la navigation et empêcher l'interception des données (cookies, formulaires...).

Validation du certificat :

- Le navigateur vérifie que la chaîne de certificats est valide (jusqu'à une CA racine de confiance).
- Un **cadenas vert** ou icône de sécurité apparaît si tout est correct.

Impact de la cryptographie quantique



Ordinateurs quantiques

Exploitent les propriétés de la **superposition** et de l'**intrication** (entanglement) pour accélérer certains calculs.



Algorithmes majeurs

Algorithme de Shor : Permet de factoriser des nombres entiers en temps polynomial, ce qui menace RSA et ECC (logarithme discret).

Algorithme de Grover : Réduit le temps de recherche brute par racine carrée. Affecte les **clés symétriques** (AES), exigeant potentiellement de **doubler** la longueur des clés pour maintenir le même niveau de sécurité.

Impact de la cryptographie quantique



Algorithmes post-quantiques

Objectif : Développer des algorithmes résistants à un attaquant disposant d'un ordinateur quantique suffisamment puissant.

Exemples : **CRYSTALS-Kyber** (chiffrement), **CRYSTALS-Dilithium**, **Falcon** (signature).

NIST : En cours de standardisation (sélection d'un ensemble d'algorithmes post-quantiques).

Transition : De nombreuses entreprises et organismes se préparent à migrer d'ici les prochaines années. La compatibilité et la mise à jour des systèmes (PKI, TLS, etc.) sont des enjeux majeurs.

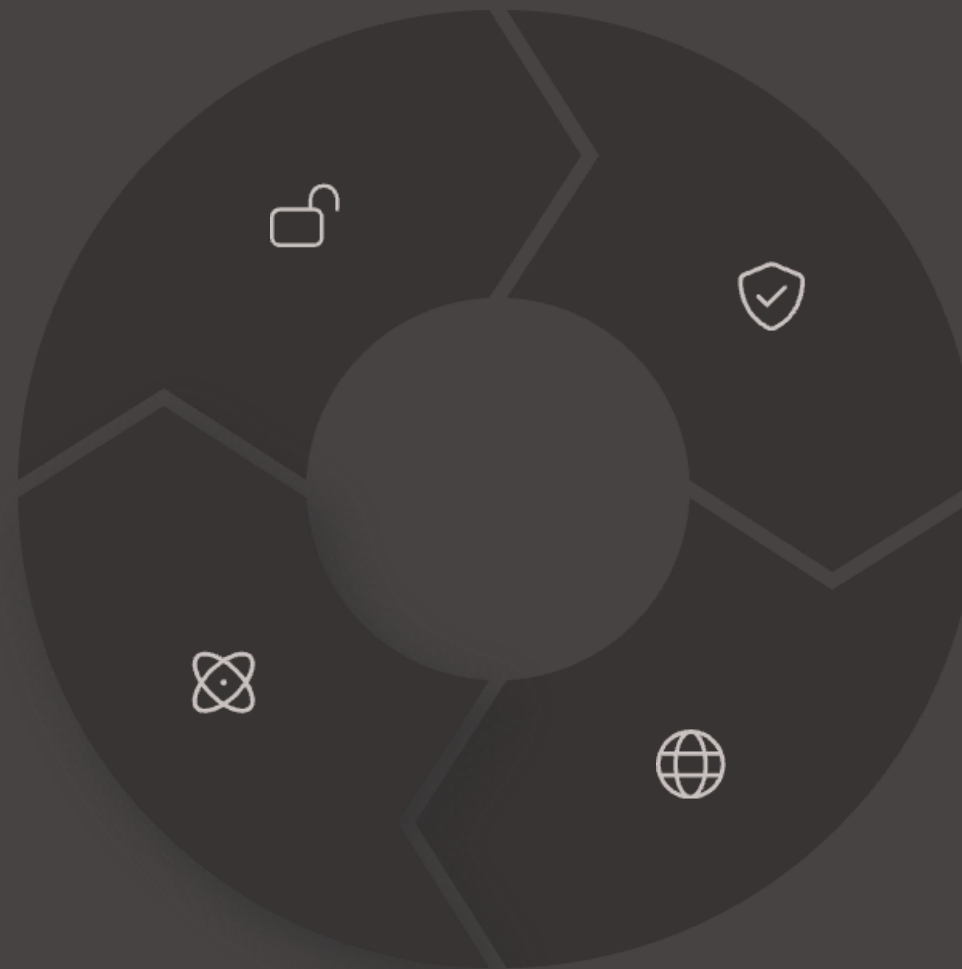
Remarques finales

PKI pour la gestion de la
confiance

Base de la sécurité moderne

Évolution vers la cryptographie
post-quantique

Repenser les briques fondamentales



Protocoles sécurisés

SSL/TLS, SSH, IPsec assurant
confidentialité, intégrité,
authentification

Sécurité des e-mails et du web

S/MIME, PGP, HTTPS indispensables
pour la communication quotidienne