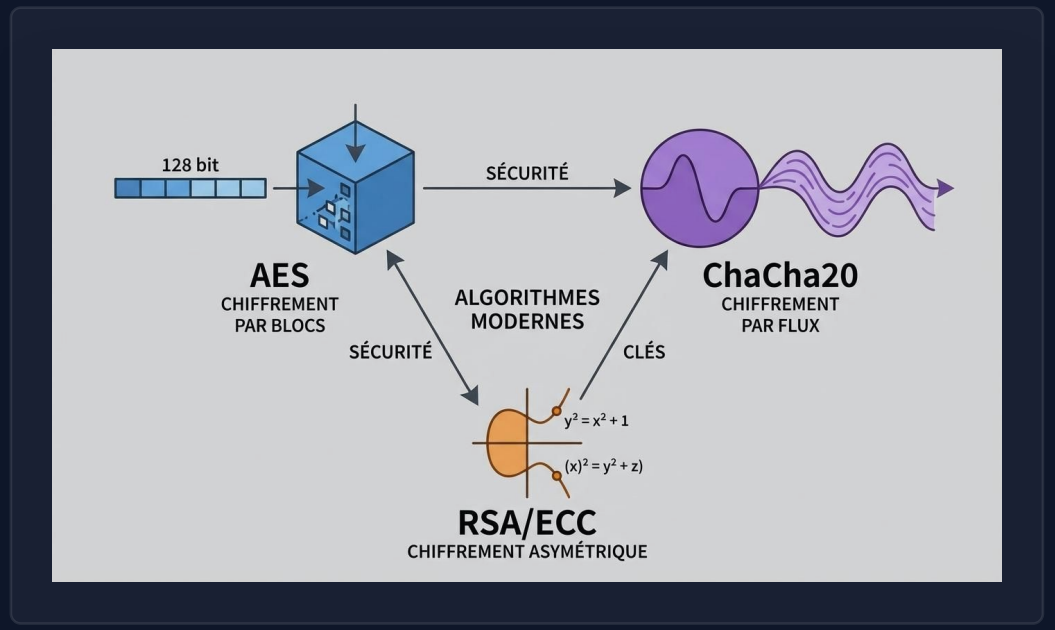
Chiffrement Appliqué au Réseau

De la théorie à la pratique : maîtrisez les concepts fondamentaux (César, AES, RSA) et découvrez leur implémentation moderne dans TLS.



SOMMAIRE

Plan de la présentation

01

Pourquoi chiffrer ?

Objectifs de sécurité fondamentaux (CIA : Confidentialité, Intégrité, Authenticité) et panorama des menaces réseaux.

02

Intro & Code César

Concepts de base avec un algorithme historique simple. Comprendre la substitution et ses limites.

03

Chiffrement Symétrique

Principe de la clé unique partagée. Focus sur les standards modernes (AES, ChaCha20) et leur rapidité.

04

Chiffrement Asymétrique

La révolution de la clé publique/privée. RSA, courbes elliptiques (ECC) et mécanismes de signature.

05

Protocole TLS

Application pratique : fonctionnement du Handshake TLS 1.3, certificats X.509 et sécurisation du web.

06

Bonnes Pratiques

Recommandations actuelles (ANSSI/NIST), gestion du cycle de vie des clés et erreurs à éviter.

FONDAMENTAUX

Pourquoi chiffrer sur les réseaux ?

OBJECTIFS DE SÉCURITÉ

Confidentialité
Données illisibles pour les tiers non autorisés.

Intégrité
Détection de toute modification des données.

Authenticité
Garantie de l'identité de l'émetteur.

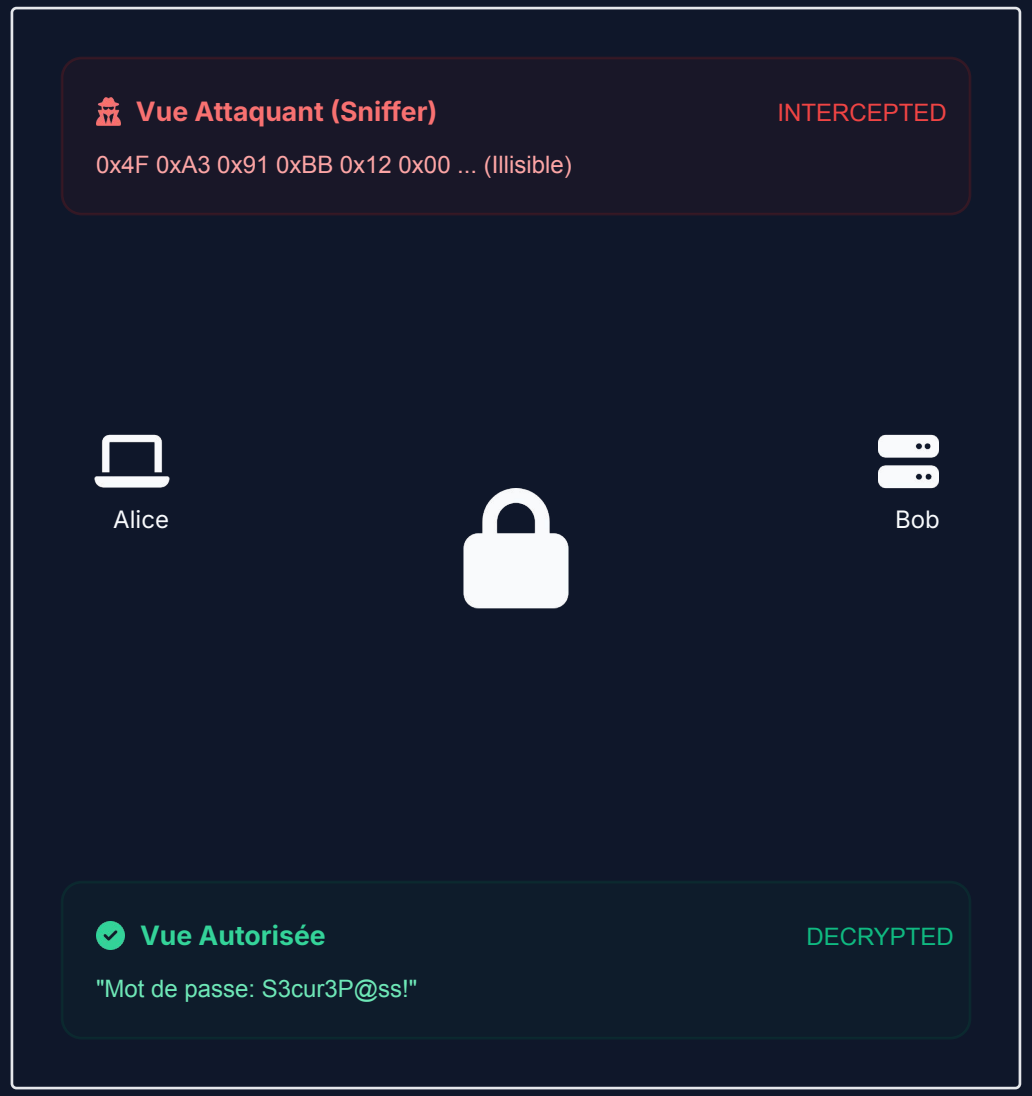
Non-répudiation
Impossibilité de nier l'envoi d'un message.

MENACES PRINCIPALES

- Écoute (Sniffing)
- Man-in-the-Middle (MITM)
- Usurpation (Spoofing)
- Injection de paquets

TERMES CLÉS

Clair (Plaintext) • Chiffré (Ciphertext) • Clé (Key) • Entropie



INTRODUCTION ALGORITHME HISTORIQUE

Le Code de César

Shift Cipher (k=3)

Principe

C'est un chiffrement par substitution monoalphabétique. Chaque lettre du texte clair est remplacée par une lettre située à un nombre fixe de positions plus loin dans l'alphabet.

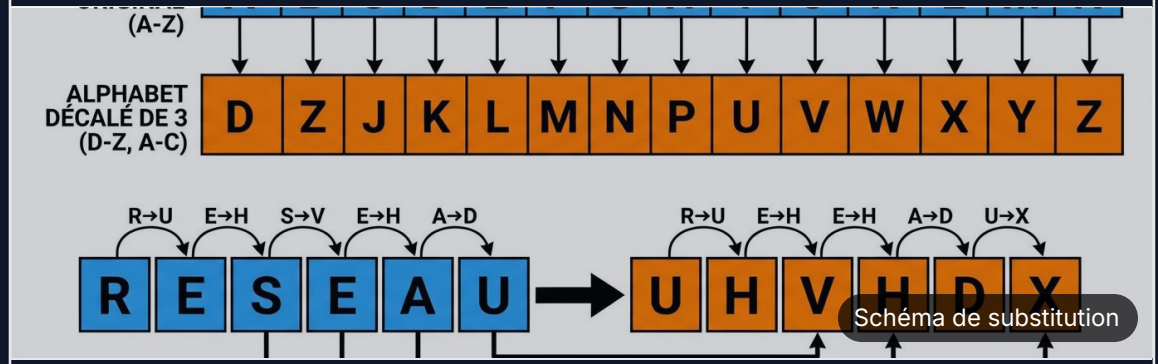
Formules Mathématiques

Chiffrement	$E(x) = (x + k) \bmod 26$
Déchiffrement	$D(x) = (x - k) \bmod 26$

Où x est la position de la lettre (A=0, B=1...) et k la clé.

Exemple Concret

"RESEAU" → k=3



TRANSFORMATION PAS À PAS

R	17 + 3 = 20 →	U
E	04 + 3 = 07 →	H
S	18 + 3 = 21 →	V

RÉSULTAT : **UHVHDX**

CONCEPTS FONDAMENTAUX > TRANSITION

De César à la cryptographie moderne

L'ère de l'informatique

Principe de Kerckhoffs

Auguste Kerckhoffs, 1883

"La sécurité d'un cryptosystème ne doit reposer que sur le secret de la clé, jamais sur le secret de l'algorithme."

- **Open Design**

  Les algorithmes (AES, RSA) sont publics et audités par le monde entier. Pas de "sécurité par l'obscurité".

- **Secret de la Clé**

  Si la clé est compromise, on la change. Si l'algo est compromis, tout le système s'effondre.

Sécurité Moderne

Mesure et robustesse

Taille de Clé & Entropie

Plus l'espace de clés est grand, plus l'attaque par force brute est difficile.

`128 bits (AES)` `2048 bits (RSA)`

Sécurité Computationnelle

Un système est sûr s'il faudrait plus de ressources (temps/énergie) pour le casser que la valeur des données protégées.

Force Brute (128 bits)
> Âge de l'univers pour tester

ARCHITECTURE CRYPTOGRAPHIQUE

Chiffrement Symétrique : Concepts

Clé Secrète Unique

La même clé K est utilisée pour chiffrer et déchiffrer.

Chiffrement : E(Message, K) = Chiffré
Déchiffrement : D(Chiffré, K) = Message

ARCHITECTURES & MODES

Par Blocs (Block)

Découpe en blocs fixes (ex: 128 bits).

AES | CBC

Par Flux (Stream)

Chiffrement bit à bit ou octet par octet.

ChaCha20 | RC4 (Obs)

ATOUTS

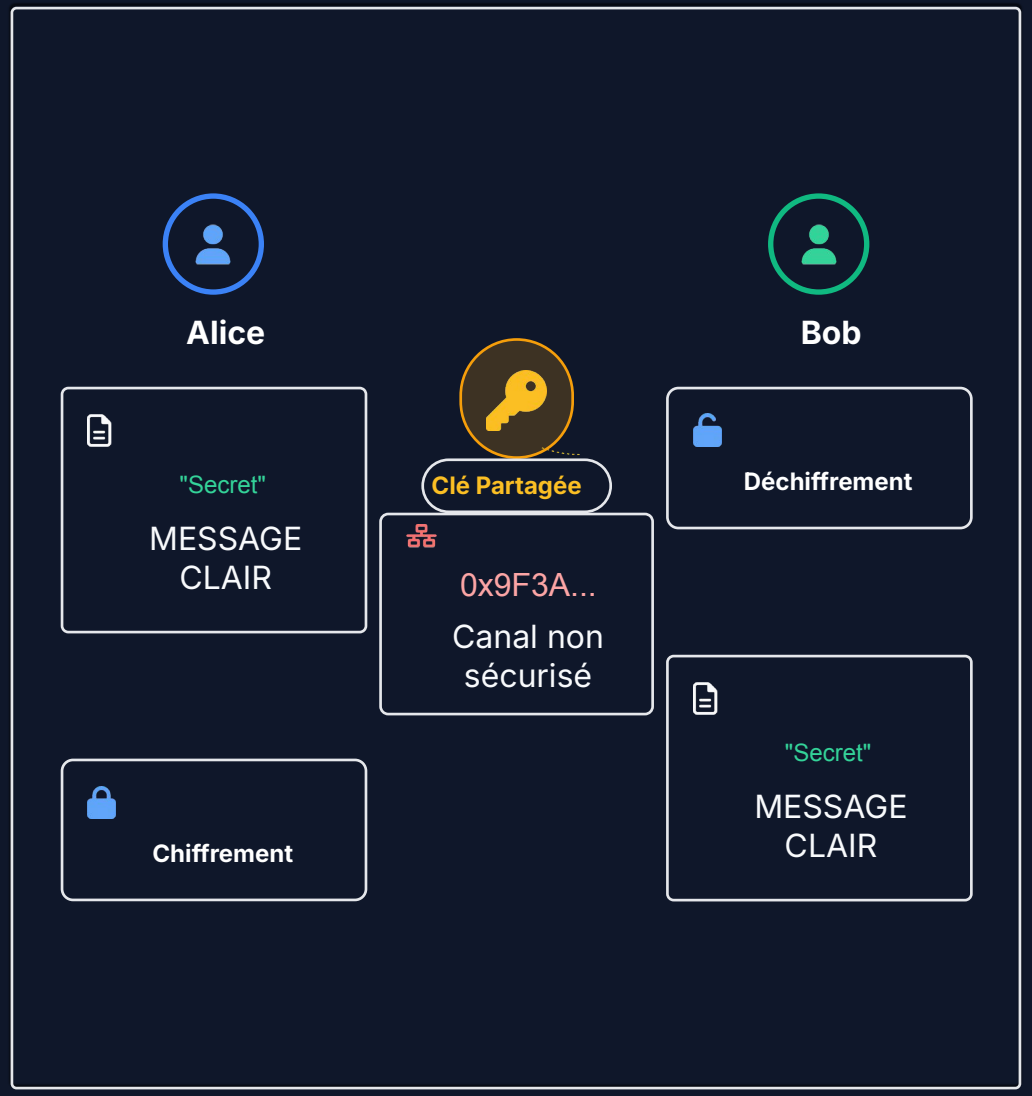
- Extrêmement rapide
- Faible latence
- Efficace pour gros volumes

LIMITES

- Distribution de la clé
- Gestion de n(n-1)/2 clés
- Non-répudiation difficile

PARAMÈTRES CRITIQUES

IV (Vecteur d'Init) : Doit être unique • Padding : Remplissage des blocs • AEAD : Chiffrement + Auth



Algorithmes de Chiffrement

État de l'art actuel

Recommandés

Standards actuels

AES

STANDARD

`128/256 bits` `GCM/CTR`

Le standard mondial (NIST). Privilégier le mode GCM (AEAD) pour garantir l'intégrité en plus de la confidentialité.

ChaCha20

MODERNE

`Stream Cipher` `Poly1305`

Alternative robuste à AES. Souvent couplé avec Poly1305 pour l'authentification (RFC 8439).

Performance

Hardware vs Software

- **AES-NI (Hardware)**

  Les processeurs modernes (Intel/AMD/ARMv8) intègrent des instructions dédiées pour AES.

  ⚡ Ultra Rapide (~GB/s)

- **ChaCha20 (Software)**

  Conçu pour être rapide en logiciel pur. Idéal pour les appareils mobiles ou IoT sans accélération AES matérielle.

"Google a popularisé ChaCha20 sur Android pour sécuriser le trafic mobile sans vider la batterie."

À Éviter (Danger)

Obsolètes ou Cassés

DES & 3DES

`Sweet32`

Blocs de 64 bits trop petits, vulnérables aux collisions. Très lents.

RC4

`Cassé`

Biais statistiques majeurs. Interdit dans TLS depuis des années.

Mode ECB

`Pattern`

Ne masque pas les motifs des données (ex: Pingouin Linux visible).

Ne jamais implémenter soi-même la crypto ("Don't roll your own crypto").

CRYPTOGRAPHIE À CLÉ PUBLIQUE

Chiffrement Asymétrique : Concepts

LE PRINCIPE DES PAIRES DE CLÉS

Clé Publique
Diffusée à tous. Sert à chiffrer des messages ou vérifier une signature.

Clé Privée
Gardée secrète. Sert à déchiffrer des messages ou signer.

Approche Hybride
L'asymétrique sécurise l'échange d'une clé symétrique (pour la rapidité).

PFS (Forward Secrecy)
Les anciennes sessions restent sûres même si la clé privée fuite (via ECDH).

USAGES PRINCIPAUX

- Échange de clés (TLS)
- Signature Numérique
- Authentification (SSH)

LIMITATIONS

- Lenteur (Calcul lourd)
- Grandes clés (>2048 bits)

ALGORITHMES & STANDARDS

RSA • ECC (Elliptic Curve) • Diffie-Hellman (DH) • Certificats X.509

SÉCURITÉ RÉSEAU > PROTOCOLES

TLS : Vue d'ensemble

Transport Layer Security (ex-SSL)

Versions & Objectifs

Évolution du protocole

"TLS garantit Confidentialité, Intégrité et Authenticité au-dessus de TCP."

- **TLS 1.2 vs 1.3**

  TLS 1.3 est plus rapide (1 RTT vs 2 RTT) et supprime les algos obsolètes (pas de RSA key exchange, pas de CBC).

  `PFS Obligatoire` `0-RTT`

- **Protection**

  Empêche l'écoute (sniffing) et l'altération (MITM) des données entre client et serveur.

Identité & Confiance

PKI & Certificats X.509

Certificat Numérique

Lie une Clé Publique à une Identité (Nom de domaine). Format standard X.509.

Chaîne de Confiance

Le navigateur fait confiance aux Autorités de Certification (AC) Racines.

Suites & Extensions

La mécanique interne

Cipher Suites

Combinaison d'algos négociée.

`TLS_AES_128_GCM_SHA256`

SNI (Server Name)

`Ext`

Le client indique quel site il veut (en clair dans le ClientHello). Indispensable pour l'hébergement mutualisé.

ALPN

`Ext`

Négociation du protocole applicatif (h2 vs http/1.1) durant le handshake TLS.