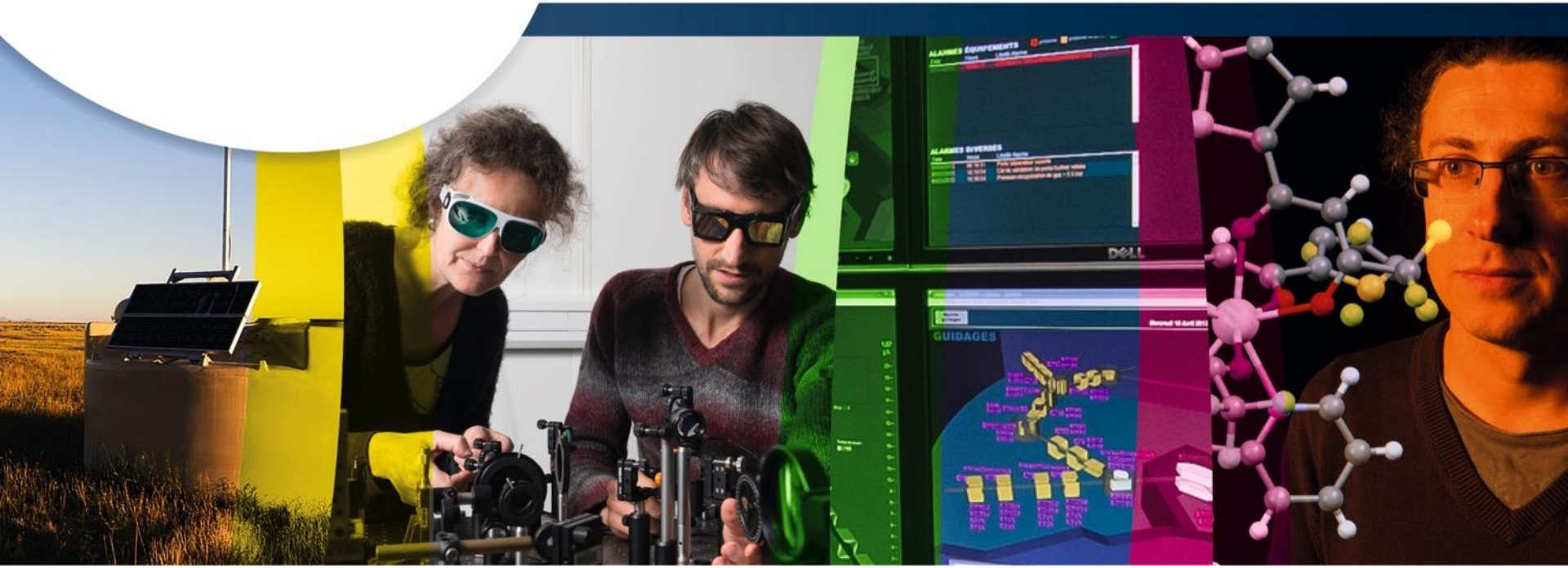
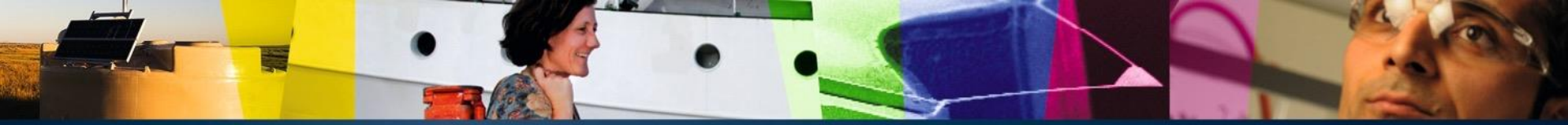
www.cnrs.fr

Confiance et cryptographie



www.cnrs.fr

A quoi sert la cryptographie ?



Vocabulaire

www.cnrs.fr

- Chiffrer, chiffrement
  - Clé de chiffrement
- Déchiffrer, déchiffrement
  - Clé de déchiffrement
- Décrypter, décryptage
  - On ignore la clé, existe-t-elle ?
- ~~Cryptor~~, ~~cryptage~~
- Cryptologie, étymologiquement la science du secret
  - Cryptographie : s'attache à la protection des messages, assure confidentialité, authenticité et intégrité
  - Cryptanalyse
    - L'art de rendre clair un message secret
    - Attaquer un système cryptographique

Des limites de la cryptographie

www.cnrs.fr

- Bruce Schneier, un cryptographe renommé
  - *Applied Cryptography (1994)* : l'utopie mathématique ou la cryptographie est la réponse à tout
    - *It is insufficient to protect ourselves with laws; we need to protect ourselves with mathematics.*
  - *Secrets & Lies (2000) :* aucun système n'est parfait, la technologie n'est pas la réponse
    - *If you think technology can solve your security problems, then you don't understand the problems and you don't understand the technology.*
  - *Liars and Outliers (2012) :* traite de la confiance

De la nécessité de la cryptographie

www.cnrs.fr

- Très difficile d'assurer la confidentialité sans chiffrement
  - Surveillance constante de l'accès à l'information par des personnes de confiance ➔ irréaliste
- Comment détecter qu'un document a été modifié sans avoir un moyen de vérification ?
  - Empreinte (*hash*), signature
- Comment prouver qu'un document a bien été écrit par tel personne sans qu'elle puisse le réfuter ?
  - Cryptographie vient à l'aide

www.cnrs.fr

Rappels de cryptographie

Hachage

www.cnrs.fr

- Fonction de hachage (*hash*)
  - Associe à un grand ensemble de données un ensemble beaucoup plus petit
  - Message m → haché, condensat ou empreinte h(m), entier de n bits
- Propriétés
  - Facile et rapide à calculer
  - Difficile à inverser (parcours exhaustif)
  - Bien réparti
  - Résistance aux attaques

Fonctions de hachage

www.cnrs.fr

- Quelques fonctions de hachage
  - MD5 définitivement cassée
  - SHA-1 sur le point d'être cassée → ne plus utiliser
  - SHA-2 : 224, 256, 384, 512 bits, construction de Merkle-Damgård
  - SHA-3 (Keccak) nouveau, fonction éponge
- Recommandations RGS annexe B1 : **utiliser SHA-2**
  - *Le mécanisme de hachage SHA-1 n'est donc pas conforme au référentiel*
  - *Le mécanisme de hachage **SHA-256** défini dans le FIPS 180-2 est conforme au référentiel*
  - *Dans l'attente de la publication de nouveaux standards, dont par exemple SHA-3, et de leur mise à l'épreuve, la famille SHA-2 reste utilisable*

Mots de passe

www.cnrs.fr

- Stockage en clair sur le serveur → catastrophique en cas de fuite
- Stockage du haché h(p)
  - Utilisé par Windows dans NTLM (MD4) et bien des sites
  - Attaque par dictionnaire
  - Attaque par force brute
  - Rainbow table : pré-calcul pour accélérer la recherche (compromis temps/mémoire)
- Utilisation de diversifiant (*salt*) h(salt || password)
  - Obstacle aux *Rainbow tables*
  - Vulnérable à des attaques hors ligne avec de puissants moyens de calcul (GPU)
- Itérations d'une fonction pour ralentir le calcul → **ce qu'il faut faire**
  - Améliore la robustesse d'un mot de passe court
  - bcrypt
  - sha256crypt,
  - scrypt
  - PBKDF2
  - argon2

Intégrité

- Fonction de hachage pour le contrôle d'intégrité
  - La probabilité que 2 messages aient le même haché est très faible
    - $2^{-128} \sim 3 \times 10^{-39}$ pour SHA-256 (paradoxe des anniversaires)
    - On considère que cela n'arrivera jamais
  - Si un message a été modifié son haché est différent
  - Si on connait de façon sûre le haché d'un message on s'assure de son intégrité

HMAC

- HMAC (Keyed-Hash Message Authentication Code)
  - hmac(k, m) = h((k ⊕ ipad) || h(k ⊕ opad) || m)
- Intégrité + authenticité : signature
  - Hachage : intégrité
  - HMAC ajoute l'authenticité
- Exemple d'utilisation : client – serveur
  - Serveur → client : d1, s= hmac(k, d1), d2
    - d1, d2, d3 : données 1, 2, 3
    - k: clé
    - s : signature
  - Client → serveur : d1, s, d3
  - Le client ne peut modifier d1 (utile si cela contient le prix dans le panier)

Cryptographie à clé secrète

www.cnrs.fr

- Appelé aussi chiffrement symétrique
- La confidentialité est basée sur l'utilisation d'un secret commun.
  - Cette méthode consiste à utiliser une clé identique pour chiffrer et déchiffrer le message.
  - Il est appelé ainsi car la clé (unique) ne doit être connue que des personnes devant pouvoir accéder au secret.
- Un même algorithme est utilisé pour le chiffrement et le déchiffrement
- Système rapide, et facile à mettre en œuvre
  - Fonctions symétriques :
  - $M' = F_k(M)$ et donc $F_k^{-1}(M') = F_k^{-1}(F_k(M)) = M$
  - Algorithme basé sur des opérations mathématiques simples (substitutions, permutations).

Types de chiffrement

www.cnrs.fr

- Masque jetable (One Time Pad) : m' = m ⊕ k
  - Clé aléatoire
  - Taille clé = taille du message
  - **Ne jamais réutiliser la clé**
  - Sécurité prouvée (Shannon)
  - Difficilement praticable
- **Par bloc**
  - Le plus répandue,
  - Découpe les données en blocs de même taille et les chiffre ensuite les uns après les autres
  - L'opération de chiffrement s'effectue sur des blocs de taille prédéfinie.
  - Bourrage
- Par flot
  - Chiffre les données bit par bit quelle que soit la longueur du message à coder sans besoin de les découper.
  - Pas de bons algorithmes, quasiment tous cassés (RC4, RC5). *Il est recommandé d'employer des primitives de chiffrement par bloc et non des algorithmes de chiffrement par flot dédiés (RGS).*

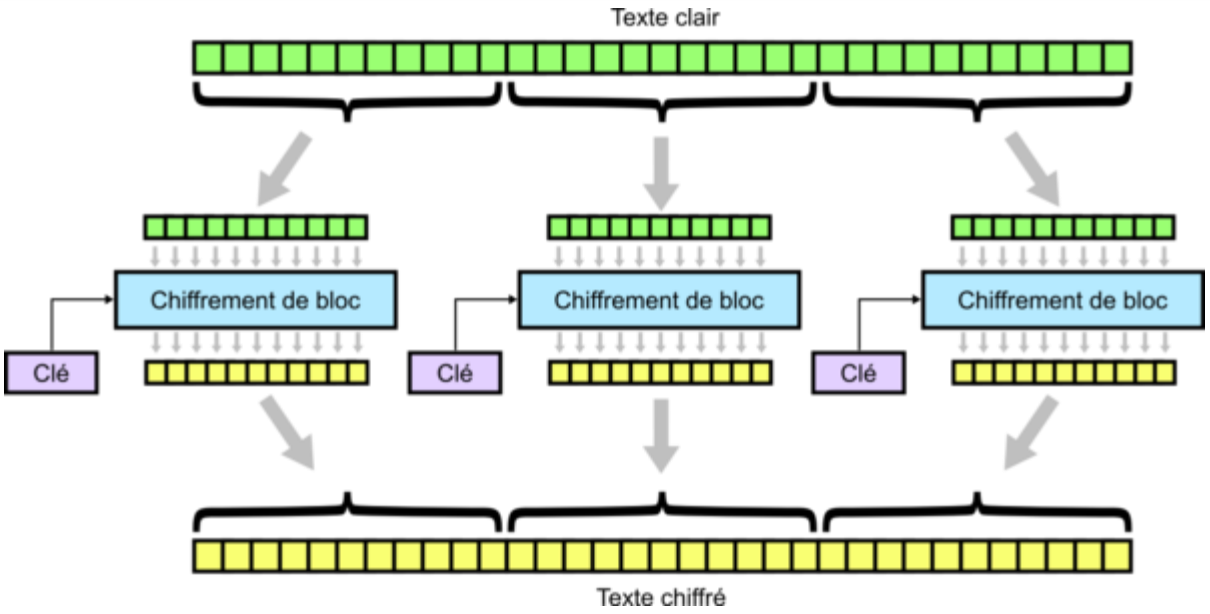
Modes de chiffrement

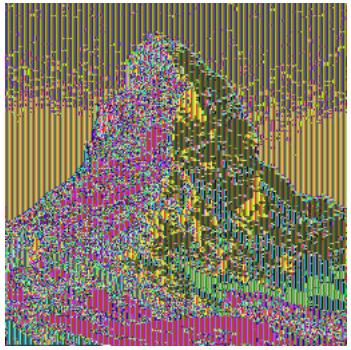
www.cnrs.fr

- ECB (Electronic Code Book)
  - Mauvais
  - Utilisé généralement par ceux qui se vantent d'un chiffrement « militaire » 256 bits et qui stockent parfois le mot de passe en clair
- CBC (Cipher Block Chaining)
- XTS (XEX-based tweaked-codebook mode with ciphertext stealing)
- GCM (Galois Counter Mode)
  - Chiffrement + contrôle d'intégrité
- Etc.

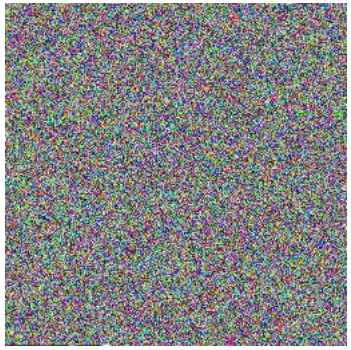
ECB (Electronic Code Book)

www.cnrs.fr





ECB



CBC

Générateurs de nombres aléatoires

www.cnrs.fr

- Générateur aléatoire
  - Phénomène physique : radioactivité, bruit thermique
  - Instructions dans les processeurs Intel récents (bruit thermique)
- Générateur pseudo-aléatoire
  - Lorsqu'une personne génère une clé de chiffrement, elle doit faire intervenir le hasard de façon à ajouter de la complexité.
  - De même, certains protocoles cryptographiques nécessitent, pour éviter la re-jouabilité, l'utilisation d'aléas imprévisibles.
  - Mais il est impossible de produire des suites aléatoires uniquement à l'aide d'un ordinateur à cause de sa nature propre.
    - Un générateur sera toujours périodique, donc prévisible
    - On les appelle alors des générateurs pseudo-aléatoires :
      - C'est un algorithme qui génère une séquence de nombres présentant certaines propriétés du hasard.
- Source d'entropie
  - Variations de temps : processus de boot, interruptions
  - Sert à initialiser le générateur pseudo-aléatoire

Générateurs pseudo-aléatoires

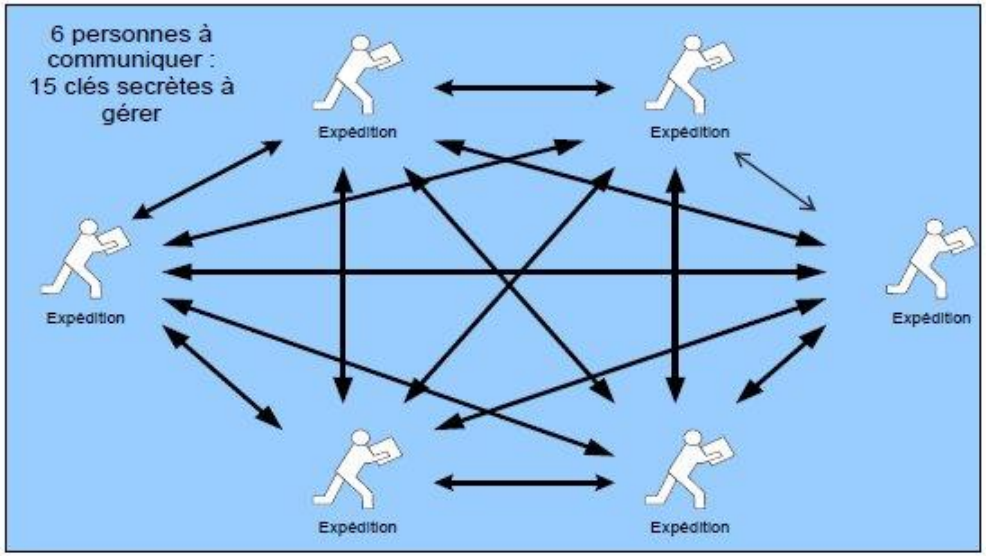
www.cnrs.fr

- Dual_EC_DRBG basé sur les courbes elliptiques
  - Peu performant
  - Intense lobbying de la NSA
  - Porte dérobée : choix de paramètres et de points particuliers
  - Pare-feu Juniper
    - Utilise Dual_EC_DRBG
    - Découverte d'une porte dérobée qui utilise d'autres paramètres
    - Une porte dérobée peut être utilisée par quelqu'un d'autre
- Confiance limitée dans le générateur matériel des processeurs Intel
  - A utiliser comme source additionnelle d'entropie

Le problème de la distribution des clés

www.cnrs.fr

- Il faut pouvoir les transmettre d'une manière sûre.
  - Grand nombre de clés à partager deux à deux entre de nombreuses personnes
  - Nombre de clés à gérer : N(N-1)/2 (N=nombre de personnes)



Cryptographie à clé publique

www.cnrs.fr

- Appelée aussi cryptographie asymétrique
  - Elle utilise 3 algorithmes :
    - Algorithme de génération des clés
    - Algorithme de chiffrement
    - Algorithme de déchiffrement
- Objectif : résoudre le problème de la distribution des clés
  - Établissement préalable d'un canal pour la transmission de la clef
  - Le principe sera d'utiliser deux clés, une par algorithme (chiffrement/déchiffrement).
  - Et ces deux clés ont la propriété d'être liées l'une à l'autre par un algorithme.

Cryptographie à clé publique

www.cnrs.fr

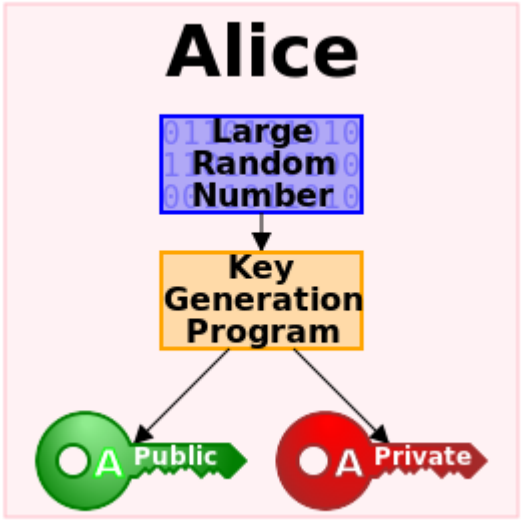
- Algorithmes basés sur des problèmes mathématiques difficiles à résoudre
  - Logarithme discret (Ex : Elgamal)
    - le calcul des logarithmes discrets s'avère difficile, tandis que le problème inverse de l'exponentiation discrète ne l'est pas. C'est une fonction à sens unique.

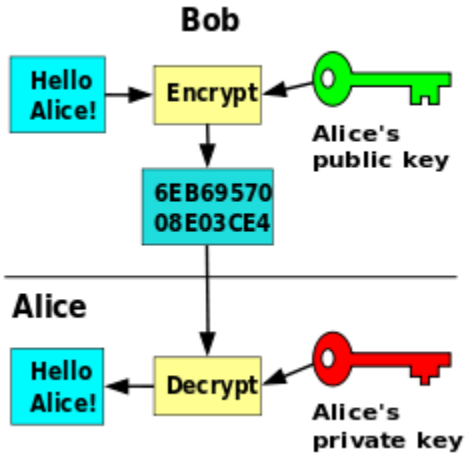
      Soient g un générateur d'un groupe G d'ordre n (groupe cyclique), et y un élément du groupe G. Le problème du logarithme discret dans G est de trouver $x \in Zn$, tel que $g^x = y$. La valeur x ainsi obtenue est appelée le logarithme discret de y.
  - Factorisation de grands nombres (ex : RSA)
    - Utilisation de fonctions à sens unique;
    - Il est facile de multiplier deux nombres premier pour obtenir un produit, mais difficile de factoriser ce produit afin de retrouver les deux nombres premiers
    - Exemple : quels sont les 2 nombres premiers p et q, dont p x q = 437 ?
  - Courbes elliptiques
    - Plus récent
    - Moins coûteux, clé plus courtes que RSA à robustesse égale
    - NSA et le futur des ordinateurs quantiques

Cryptographie à clé publique

www.cnrs.fr

- Fonctionne avec une paire de clés uniques (bi-clés)
  - Une clé privée : connue que du propriétaire de la paire de clé
  - Une clé publique : connue de tous, souvent publié dans un annuaire





Signature

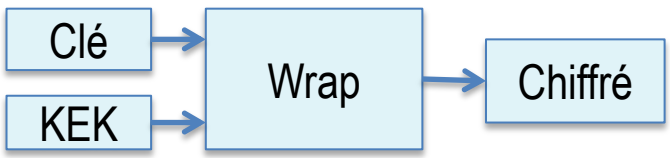
www.cnrs.fr

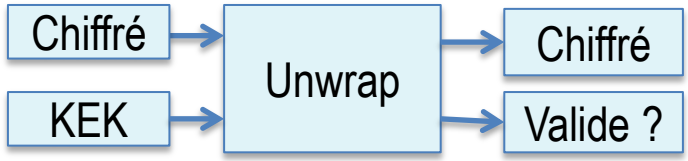
- Alice calcule l'empreinte du document
- Alice chiffre cette empreinte avec sa clé privée → signature
- Alice transmet à Bob le document avec sa signature
- Bob récupère la clé publique d'Alice
- Bob déchiffre la signature à l'aide de cette publique
- Bob calcule l'empreinte du document et la compare à celle issue du déchiffrement de la signature
- Si identiques le document est intègre et a bien été signé par Alice

AES Key Wrap

www.cnrs.fr

- Encapsule en utilisant AES une clé en clair de façon sûre avec contrôle d'intégrité
  - Stockage sur un support non sûr
  - Transfert
- Le taille de la clé encapsulée peut être différente de la taille du bloc
- RFC 3394, spécifications du NIST
- Key encryption key (KEK)
- Exemple : protection des clés de chiffrement de disque





PFS (*Perfect Forward Secrecy*)

- PFS ou confidentialité persistante
  - Propriété qui garantit que la découverte par un adversaire de la clé privée d'un correspondant (secret à long terme) ne compromet pas la confidentialité des communications passées
  - Échanges de clés éphémères par Diffie-Hellmann
- Diffie-Hellmann
  - Dans un groupe fini Alice et Bob se mettent d'accord sur g et p
  - Alice choisit a calcule et envoie à Bob $g^a \mod p$
  - Bob choisit b et calcule et envoie à Alice $g^b \mod p$
  - Bob reçoit g^a et calcule $(g^a)^b \mod p = g^{ab}$
  - Alice reçoit g^b et calcule $(g^b)^a \mod p = g^{ba} = g^{ab}$
  - g^{ab} est utilisé comme clé secrète pour les échanges
  - Vulnérable aux attaques MitM (Man in the Middle)
    - Contremesure : authentification
- A mettre en œuvre systématiquement

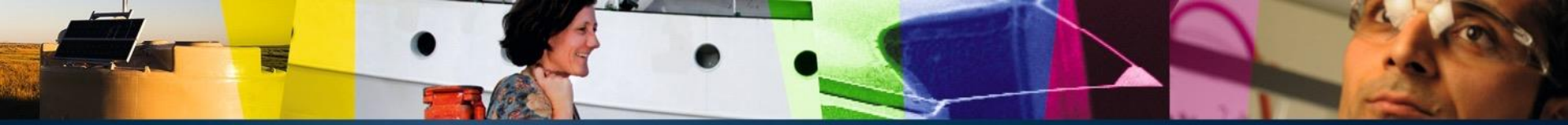
Certificats

www.cnrs.fr

- X509
  - Identité
  - Clé publique
  - Divers attributs
  - Le tout signé par l'autorité de certification
- IGC
  - Hiérarchie d'autorités de certification
    - AC racine ➔ AC 1 ➔ AC 2 ➔ … ➔ utilisateur
    - Transitivité de la confiance
    - Si on fait confiance à la racine on fait confiance au certificat de l'utilisateur
    - Peut-on toujours faire confiance à l'AC racine ? Non

www.cnrs.fr

Applications



Cryptographie dans les applications

www.cnrs.fr

- État des lieux ➔ catastrophique
  - Analyse des 1000 applications Android les plus téléchargées
    - 73% de celles qui communiquent avec un serveur ne vérifient pas son certificat
    - 8% ne vérifient pas le nom de l'hôte
    - 77% de celles qui utilisent Webkit ignorent les erreurs SSL
    - 68% ont au moins une de ces 3 failles SSL/TLS
    - Permet des attaques Man in the Middle
- On trouve tout ce qu'il ne faut pas faire
  - Top 10 OWASP
  - Mot de passe en clair
  - Non vérification des certificats
  - Mauvaise utilisation d'un jeton
  - Etc.

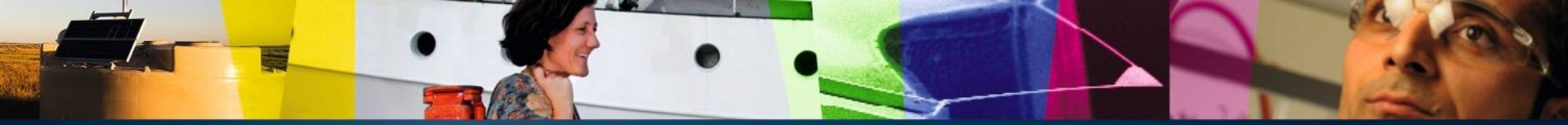
Les raisons d'un échec

www.cnrs.fr

- La cryptographie et sa mise en œuvre sont intrinsèquement compliquées
- Les API sont complexes, manquent de cohérence, sujettes à erreur
- Les développeurs récupèrent n'importe quel code sur Internet à l'aide d'un moteur de recherche
- Les développeur suppriment les contrôles pour la mise au point et ne les rétablissent pas pour la production
- *Time to market* ➔ impasse sur la sécurité

www.cnrs.fr

Perspectives



Ouverture vers l'Internet des objets

www.cnrs.fr

- On ne peut ignorer les objets connectés
  - De plus en plus présents
  - On en utilise
  - On en produit
- Contraintes spécifiques
  - Puissance de calcul limitée
  - Consommation électrique (batterie)
  - Réseau à faible débit
  - Maintenance, mises à jour, fin de vie
- Besoins de sécurité
  - D'abord intégrité et authenticité
  - Avant confidentialité
  - Risques humains
- Un défi qu'il faut relever

Futur de la cryptographie

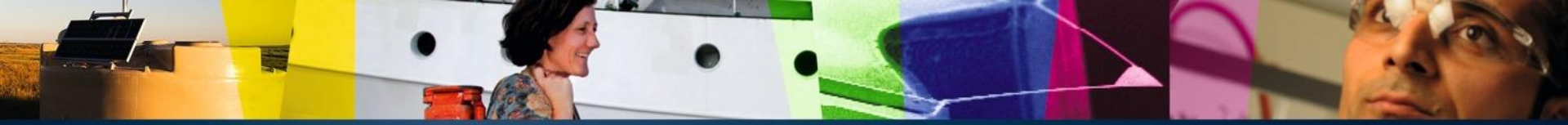
www.cnrs.fr

- Chiffrement homomorphe
  - Permet d'effectuer des opérations sur les données chiffrées
  - Encore très peu performant (CPU, mémoire)
- Cryptographie post ordinateurs quantiques
  - Logarithme discret, RSA, courbes elliptiques ne résistent pas : algorithme de Shor
  - Mise en garde NSA
  - AES : taille de la clé divisée par 2
  - Algorithmes à l'étude

Bonnes pratiques de développement

www.cnrs.fr

- Ne jamais chercher à développer un nouvel algorithme ou protocole cryptographique ➔ laisser cela aux spécialistes
- Ne pas réinventer la roue ➔ utiliser des outils, des bibliothèques éprouvées
- Les systèmes offrent des fonctionnalités intéressantes en matière de sécurité ➔ les utiliser
  - Classes de chiffrement dans iOS
  - Utiliser les mécanismes de gestions des secrets fournis par tous les OS (Trousseau de clés, magasins, etc.)
  - Ne pas se contenter des valeurs par défaut
- Mettre en œuvre les recommandations du RGS
  - Taille de clé
  - Algorithmes
  - Gestion des secrets d'authentification



Focus sur quelques points

www.cnrs.fr

- Mots de passe
  - Systématiquement hacher le mot de passe
  - Toujours utiliser un diversifiant (*sal*t)
  - Utiliser une fonction qui ralentit le calcul
- Génération de nombres aléatoires
  - Initialiser le générateur avec suffisamment d'entropie
- Chiffrement symétrique
  - AES
  - Clé bien aléatoire
    - PFS pour le chiffrement des échanges
  - Mode de chiffrement sûr (fonction de l'usage)
    - Avec contrôle d'intégrité si possible
  - Vecteur d'initialisation aléatoire
  - Bourrage (padding)

Focus sur quelques points

www.cnrs.fr

- Systématiquement vérifier la validité du certificat présenté
  - Signé par une autorité de certification reconnue
  - Dates de validité
  - Non révoqué
  - Usage (signature, chiffrement, authentification, etc.)
  - Seule parade au MITM

www.cnrs.fr

Questions ?