Mesures techniques : Limites du contrôle d'accès et chiffrement

M231 — Appliquer la protection et la sécurité des données

Jérôme Frossard

EPAI

14 mai 2024

Plan

1. Introduction
2. Attaque de l'intérieur
3. Usurpation d'identité
4. Accès physique non autorisé
5. Chiffrement des données
6. Chiffrement et authentification

Plan

1. Introduction
2. Attaque de l'intérieur
3. Usurpation d'identité
4. Accès physique non autorisé
5. Chiffrement des données
6. Chiffrement et authentification

Introduction

Dans un système d'exploitation moderne, le contrôle d'accès est solide. Il est difficile à attaquer directement, et protège efficacement les données.

Mais cette protection n'est pas absolue. Parmi les principaux risques, on peut mentionner :

- **Attaque de l'intérieur** : p. ex., vol de données par une personne de l'organisation.
- **Usurpation d'identité** : p. ex., en utilisant des données personnelles obtenues grâce à une cyberattaque (ingénierie sociale, fuite de données, etc.)
- **Accès physique non autorisé** : p. ex., vol d'ordinateurs ou de dispositifs de stockage (disque, SSD, clé USB, etc.), accès illicite à des locaux, etc.

Dans cette présentation, nous commençons par décrire brièvement ces risques et les principales mesures de mitigation, avant d'aborder les différentes formes de chiffrement de données (cryptographie) et leurs applications.

Plan

1. Introduction
2. Attaque de l'intérieur
3. Usurpation d'identité
4. Accès physique non autorisé
5. Chiffrement des données
6. Chiffrement et authentification

Attaque de l'intérieur

Une attaque de l'intérieur peut être le fait d'une **personne au sein de l'organisation**, ou d'une personne extérieure parvenue à **percer la sécurité du périmètre** (accès VPN, usurpation d'identité, malware, escalade de privilèges, pénétration des locaux, etc.).

Les principales mesures de protection sont :

- **Politiques* de sécurité strictes :** *zero trust*, application du principe de moindre privilège, contrôle d'accès au réseau (IEEE 802.1X), surveillance des locaux, etc.
- **Analyse des journaux de sécurité :** permet de détecter des comportements suspects ; peut être facilitée par la mise en œuvre d'un système de gestion des événements et des informations de sécurité (SIEM), etc.
- **Chiffrement des données :** permet de renforcer l'application des politiques de sécurité, indépendamment du contrôle d'accès.

***** Le terme politique (*policy*) désigne un ensemble de normes, de règles et de pratiques définies dans un but précis, dont l'application peut être rendue obligatoire par des mesures techniques et organisationnelles.

Plan

1. Introduction
2. Attaque de l'intérieur
3. Usurpation d'identité
4. Accès physique non autorisé
5. Chiffrement des données
6. Chiffrement et authentification

Qu’est-ce que l’usurpation d’identité

De manière générale, on peut définir l’usurpation d’identité comme ***l’utilisation de données personnelles propres à identifier une personne sans son accord.***

Dans le domaine des TIC, l’usurpation d’identité est généralement liée au vol de **données de connexion** de l’un des comptes de la personne, par exemple :

- Un compte de messagerie.
- Un compte d’utilisateur·rice sur un poste de travail.
- Un compte d’utilisateur·rice dans un réseau d’entreprise.

Dans tous les cas, une usurpation d’identité traduit un **manquement dans la protection des données**, mais il ne s’agit que très rarement d’une faille logicielle (même si cela n’est bien sûr pas exclu).

Méthodes d'obtention d'informations

Dans la plupart des cas, le vol de données personnelles conduisant à une usurpation d'identité repose en grande partie sur les techniques suivantes :

- **OSINT (Open Source Intelligence)** : Utilisation de sources publiques pour collecter des informations sur des individus.
- **Ingénierie sociale** : Manipulation psychologique visant à amener les personnes à divulguer volontairement des informations confidentielles. Mets en œuvre de diverses techniques dont le *phishing*, dans le but d'abuser la confiance ou exploiter l'ignorance de la victime.
- **Fuites de données** : Accès non autorisé à des bases de données, souvent résultant d'attaques exploitant les techniques ci-dessus, mais pouvant également survenir à la suite d'un acte malveillant ou d'une négligence au sein même de l'organisation.

Prévenir les attaques utilisant ces techniques est un aspect important de la **cybersécurité**.

Mesures de protection

Le risque d'usurpation d'identité peut être fortement mitigé par la mise en œuvre d'une **authentification multifactorielle**.

Toutefois, la prévention relève en grande partie du domaine organisationnel :

- **Politiques de sécurité strictes** : définition et application de normes, de règles, et de pratiques pour la création et le renouvellement des mots de passe, pour la détection et la réponse aux incidents de sécurité, etc.
- **Formation continue** : sensibilisation du personnel sur les risques de sécurité, les méthodes de prévention comme le repérage de tentatives de phishing, et renforcement de la culture de sécurité au sein de l'organisation.

Plan

1. Introduction
2. Attaque de l'intérieur
3. Usurpation d'identité
4. Accès physique non autorisé
5. Chiffrement des données
6. Chiffrement et authentification

Accès physique non autorisé

Lorsque l'on parle d'accès physique, cela peut être un accès :

- À la console (clavier, écran, souris, etc.)
- Aux ports d'entrées-sorties (USB, Thunderbolt, Ethernet, etc.)
- À l'alimentation électrique (bouton d'alimentation, prise électrique)
- À l'interface du firmware (BIOS/UEFI) ou à l'intérieur du boîtier

Un accès physique non autorisé présente un **risque sérieux de prise de contrôle** du système, notamment :

- accès à l'ensemble des données du système,
- installation de malwares : mouchard (*spyware*), porte dérobée (*backdoor*), etc.
- installation de matériel malveillant : *key logger*, *key stroke injection*, implants, etc.

Accès physique non autorisé — Mesures de protection

Les principales mesures de protection sont :

- **Mesures de sécurité physique :** verrouillage des locaux, portique d'entrée, surveillance vidéo, journalisation des accès (badge de sécurité), serrure biométrique, câbles et verrous pour attacher les appareils au bureau, etc.
- **Politiques de sécurité strict :** définition et application de politiques claires sur qui peut accéder à quoi, quand, et dans quelles conditions.
- **Sécurisation du firmware (BIOS/UEFI) :** avec des mots de passe forts, pour prévenir les modifications non autorisées des paramètres de démarrage.
- **Chiffrement des disques :** avec LUKS ou BitLocker pour empêcher le montage des disques dans un autre système pour contourner le contrôle d'accès.
- **Mise en œuvre de TPM :** pour créer et stocker une empreinte du matériel et du logiciel installés de manière à prévenir tout changement.

Plan

1. Introduction
2. Attaque de l'intérieur
3. Usurpation d'identité
4. Accès physique non autorisé
5. Chiffrement des données
6. Chiffrement et authentification

Qu’est-ce que le chiffrement des données

Le chiffrement est une mesure technique utilisée pour renforcer la confidentialité et l’intégrité des données, ainsi que pour en assurer l’authenticité (signature, HMAC).

Il existe différents types d’algorithmes de chiffrement, chacun adapté à des usages spécifiques.

- **Chiffrement symétrique (clé secrète)** : Utilisé pour le chiffrement rapide et sûr des données en transit (dans un canal de communication) et des données au repos (dans un dispositif de stockage).
- **Chiffrement asymétrique (clé publique/clé privée)** : Utilisé pour l’échange de clés secrètes, la signature numérique, ainsi que pour l’authentification à clé publique (ssh).
- **Chiffrement à sens unique* :** : Utilisé pour la protection des mots de passe et la signature électronique.

***** par chiffrement à sens unique, on entend le hachage, la dérivation de clé, et le chiffrement de mots de passe sur lesquels nous revenons plus loin.

Principe du chiffrement

Une clé est un nombre de longueur fixe :

- Pour le chiffrement symétrique, le standard AES définit trois longueurs de clé : 128, 192 et 256 bits.
- Pour le chiffrement asymétrique, la taille varie selon l'algorithme : ≥ 2048 bits pour RSA, 256 bits pour ED25519.

Pour une entrée et une clé données, un algorithme de chiffrement **produit toujours la même sortie** (données chiffrées) qui devrait être indiscernable de données aléatoires.

Les données chiffrées ne devraient contenir **aucune information exploitable**. En pratique, c'est le travail de la cryptanalyse de vérifier que c'est bien le cas et de disqualifier les algorithmes lorsque cela n'est plus vrai (p. ex. DES et RC4).

À l'exception du chiffrement à sens unique, le chiffrement est réversible à condition de disposer de la clé appropriée.

Chiffrement à sens unique

Un algorithme de chiffrement à sens unique transforme une entrée de longueur quelconque en une **empreinte** (*hash*) unique et de taille fixe, typiquement comprise entre 128 et 512 bits selon les algorithmes.

Principales caractéristiques :

- **Fonction mathématique :** une même entrée produit toujours la même empreinte.
- **Irréversible :** impossible de retrouver l'entrée à partir de l'empreinte.
- **Résistance aux collisions :** très faible probabilité de produire une même empreinte pour deux entrées différentes.
- **Effet avalanche :** deux entrées presque identiques devraient produire des empreintes très différentes.

Chiffrement à sens unique — fonction de hachage

Le but d'une fonction de hachage est de produire des empreintes le plus rapidement possible et en utilisant le moins de ressources possible.

Ces fonctions sont utilisées pour :

- Vérifier l'intégrité de données
- Vérifier l'authenticité de données : signature électronique, HMAC
- Réaliser des fonctions de dérivation de clés ou de chiffrement de mot de passe.

Les fonctions de hachage de qualité cryptographique les plus courantes sont :

- SHA-256 et SHA-512 (empreintes de 256 et 512 bits)
- RIPEMD-160 (empreintes de 160 bits)
- Whirlpool (empreintes de 512 bits)

Remarque : Les algorithmes MD4, MD5, RIPEMD et SHA-1 ne devraient plus être utilisés.

Chiffrement à sens unique — fonction de dérivation de clé

Une fonction de dérivation de clé (KDF) permet de produire une ou plusieurs clés à partir d'un secret maître, typiquement une phrase secrète (*passphrase*).

Principales caractéristiques :

- **Délibérément lente :** répète un grand nombre de fois l'application d'une fonction de hachage, pour augmenter le coût d'une attaque par force brute.
- **Utilisation d'un sel :** un sel est une donnée aléatoire ajoutée à la phrase secrète, pour garantir des empreintes uniques même pour des phrases identiques.

Pour produire chaque fois les mêmes clés à partir de la phrase secrète, le sel et le coût (nombre d'itérations) doivent être stockés avec les données chiffrées.

L'un des principaux algorithmes est :

- PBKDF2 (typiquement avec SHA-256, SHA-512, RIPEMD-160, ou Whirlpool)

Chiffrement à sens unique — chiffrement de mot de passe

Pour stocker un mot de passe sans le divulguer, on utilise une fonction réalisée à partir d'une KDF ou une fonction de hachage. Cette fonction renvoie le mot de passe chiffrer sous la forme d'une chaîne de caractère qui contient :

- Un préfixe identifiant l'algorithme (ex : **y** pour yescrypt)
- Les paramètres utilisés, typiquement le sel et le coût (nombre d'itérations).
- L'empreinte de la concaténation du mot de passe et du sel

Le plus souvent, une fonction associée permet de valider un mot de passe avec un temps d'exécution constant pour contrer les attaques temporelles.

Quelques-uns des principaux algorithmes :

- **scrypt** et **yescrypt**
- **argon2**
- **bcrypt**

Chiffrement symétrique

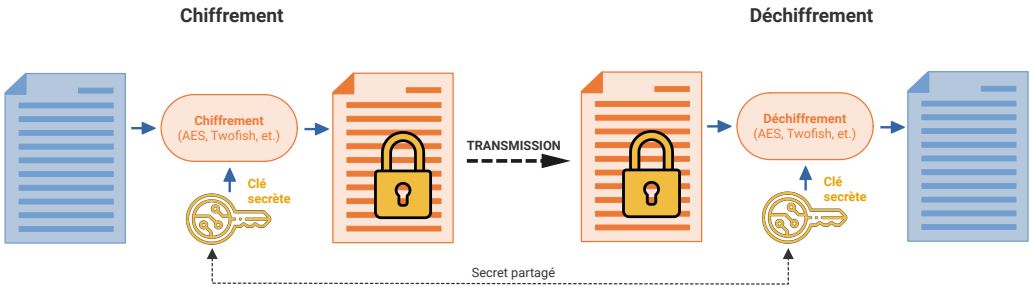
Le chiffrement symétrique ou à clé secrète utilise la **même clé pour chiffrer et déchiffrer** les données. Ces algorithmes sont conçus pour être :

- **Rapide et efficace :** ils doivent permettre de chiffrer de grandes quantités de données, et de chiffrer et déchiffrer les données d'un canal de communication en temps réel.
- **Résistant à la cryptanalyse :** si des données chiffrées sont volées, il doit être **extrêmement difficile** de décrypter l'information, c.-à-d. casser le code sans la clé.

Algoritmes courants :

- **AES** (standard étasunien et international), **Camellia** (standard européen et japonais)
- **Serpent** (domaine public, license libre)
- **Blowfish** (licence libre)
- **Twofish** (domaine public)

Chiffrement symétrique — Chiffrement et déchiffrement



Chiffrement symétrique — Chiffrement de disque entier

Le chiffrement de disque entier (*full disk encryption*) permet d'interdire le montage d'un disque chiffré dans un autre système, à moins d'en posséder la clé ou la phrase secrète.

Chaque système d'exploitation possède sa propre technologie :

- **LUKS (Linux) :** standard pour le chiffrement de disques sous Linux ; supporte les principaux algorithmes de chiffrement symétrique (AES, Serpent, etc.).
- **BitLocker (Windows) :** dépend de TPM 2.0 (*Trusted Platform Module*) pour la génération et le stockage des clés ; utilise AES avec des clés de 128 ou 256 bits.
- **FileVault2 (macOS) :** utilise AES avec des clés de 128 et 256 bit, la phrase secrète est le mot de passe de l'utiliateur·rice.

Les système de chiffrement de disques entiers sont des système de chiffrement de **données au repos** (*data at rest*) par opposition au **données en transit** (*data in transit*) lorsqu'elle cisculent dans un canal de communication

Chiffrement symétrique — Chiffrement de données en transit

La sécurité au niveau de la couche de transport (TLS) utilise principalement un chiffrement symétrique pour protéger les données en transit via un canal de communication TCP.

L'algorithme de chiffrement est typiquement AES avec une clé de 256 bits.

- **Problème :** Pour chiffrer les communications avec AES, les deux parties doivent partager la même clé secrète. Mais, comment échanger cette clé de manière sécurisée sans qu'elle ne soit interceptée ? Si la clé est transmise en clair, elle n'est plus secète.

- **Solution :** Utiliser un algorithme qui permet d'échanger la clé secrète de manière sécurisée sans avoir à partager de secret : un algorithme de chiffrement asymétrique.

Chiffrement asymétrique (1/2)

Par rapport au chiffrement symétrique, le chiffrement asymétriques est **lent** et ne permet de **chiffrer que de petites quantités de données**, mais il **ne nécessite pas de secret partagé**.

Au lieu de cela, le chiffrement asymétrique utilise une **paire de clés** :

- **Une clé publique** qui peut être librement distribuée
- **Une clé privée** qui ne doit en aucun cas être divulguée

Des données chiffrées avec l'une des clés de la paire peuvent être déchiffrées avec l'autre.

Selon qu'on les chiffrent avec la clé publique ou la clé privée, ces données sont :

- Déchiffrables exclusivement par le détenteur de la clé privée (**échange de clés**)
- Déchiffrables par tout le monde : (**signature numérique**)

L'échange de clés (*key exchange*) et la signature numérique (*digital signature*) sont les principales application du chiffrement asymétrique.

Chiffrement asymétrique (2/2)

Le principe du chiffrement asymétrique repose sur l'existence de problèmes (p. ex. la factorisation de grand nombre) pour lesquels, il est :

- très facile d'effectuer un calcul dans un sens (multiplier deux grands nombres premiers)
- très difficile (pratiquement impossible) de le faire dans l'autre sens (factoriser un grand nombre)

La clé publique et la clé privée ne sont pas indépendantes. Il est pratiquement impossible de trouver la clé privée à partir de la clé publique, mais l'inverse est assez facile.

Principaux algorithmes :

- Echange de clé : **RSA**, **Diffie-Hellman**, **ECDH**
- Signature numérique : **DSA**, **ECDSA**, **EdDSA** (Ed25519, Ed448)

Chiffrement asymétrique – Échange de clé

Principe de l'énchange de clé (*key exchange*) :

- Le client **chiffre** la clé secrète avec la **clé publique du serveur**, et l'envoie sur le canal de communication non sécurisé.
- Comme la clé secrète ne peut être **déchiffrée** qu'avec la **clé privée du serveur**, la clé secrète ne risque pas d'être divulguée.

L'échange de clé est parfaitement sûr, tant que le client utilise effectivement la clé publique du serveur. Mais comment en être certain ?

Comme nous allons le voir, le chiffrement peut également être utiliser pour authentifier des données.

Plan

1. Introduction
2. Attaque de l'intérieur
3. Usurpation d'identité
4. Accès physique non autorisé
5. Chiffrement des données
6. Chiffrement et authentification

Chiffrement et authentification (1/2)

Comme nous l'avons vu, le chiffrement assure la confidentialité et l'intégrité des données, en interdisant leur consultation et leur modification à moins de posséder la clé.

Le chiffrement peut également être utilisé pour authentifier des données et s'assurer qu'elles n'ont pas été trafiquées (*tampered*)

Les deux principaux mécanismes :

- **HMAC** pour le chiffrement symétrique
- **Signature numérique** pour le chiffrement asymétrique

Chiffrement et authentification — Principe du HMAC

Un **HMAC** (*hash-based message authentication code*) est essentiellement une empreinte que l'on obtient en concaténant un message et une **clé secrète**.

Comme la clé secrète doit être **partagée entre les parties**, cela implique qu'elles se font mutuellement **confiance**.

Création du HMAC :

- Calculer le HMAC avec le message et le **secret partagé**.
- Joindre le HMAC au message.

Vérification :

- Calculer le HMAC avec le message et le **secret partagé**.
- Vérifier que l'empreinte calculée et l'empreinte déchiffrée correspondent.

Chiffrement et authentification — Principe de la signature numérique

Pour la **signature numérique**, l'utilisation d'un chiffrement asymétrique permet d'éviter le partage d'un secret.

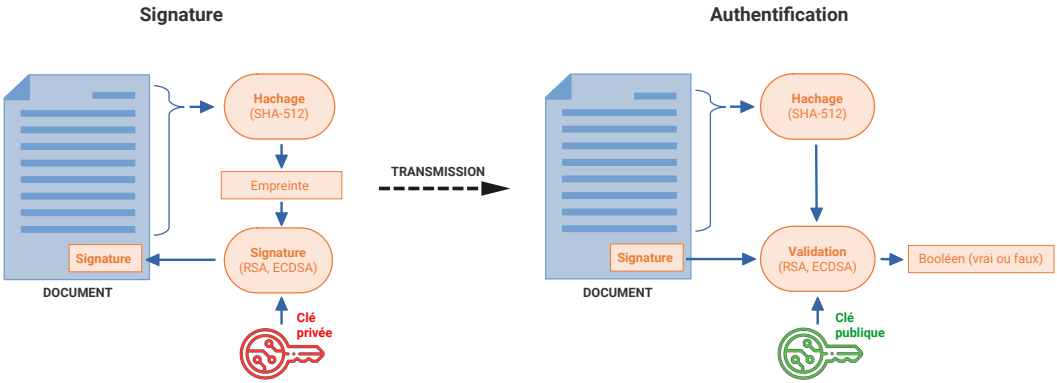
Création de la signature :

- Calculer l'empreinte du message,
- **Chiffrer** cette empreinte avec la **clé privée**,
- Joindre la signature au message.

Authentification :

- Calculer l'empreinte du message,
- **Déchiffrer** cette empreinte avec la **clé publique**,
- Vérifier que l'empreinte calculée et l'empreinte déchiffrée correspondent.

Chiffrement et authentification — Signature et authentification



Chiffrement et authentification — Communiquer la clé publique

Problème :

Dans le cas de la signature, il n'est pas nécessaire de partager un secret, mais il faut quand même obtenir la clé publique d'une manière ou d'une autre.

Si la clé publique est distribuée avec le message, qu'est-ce qui empêche un attaquant de :

- Intercepter et modifier le message,
- Signer le message avec sa clé privée
- Renvoyer le message signé avec sa clé publique ?

Solutions :

- Une première solution de vérifier personnellement que la clé publique est bien celle de la personne ou de la machine à l'origine du message (solution adoptée dans ssh).
- Une deuxième solution est l'utilisation d'un **certificat de clé publique**.

Chiffrement et authentification — Certificat de clé publique

Un **certificat de clé publique** (*public key certificate*) est un document **signé** et délivré par un **tiers de confiance** que l'on appelle **autorité de certification**.

Ce document contient :

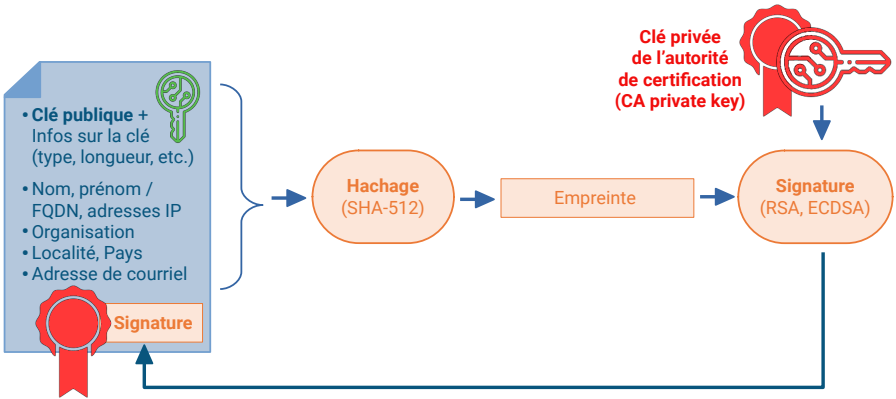
- Des informations d'identification (données de contact, FQDN d'une machine, etc.),
- La clé publique de la personne ou de la machine.

Pour vérifier le certificat, on utilise le certificat de clé publique de l'autorité de certification, que l'on appel **certificat racine** (*root certificate*)

Le certificat racine est signée par l'autorité de certification elle-même. Il est donc **auto-signé** (*self-signed*) ; on utilise la clé publique du certificat pour en vérifier la signature.

Pour que le système puisse utiliser un certificat racine, celui-ci doit être placé dans le dépôt des **certificats de confiance** (*trusted certificates*).

Chiffrement et authentification — Certificat de clé publique



Chiffrement et authentification — Validation de certificat