Identification, Authentification

INF8085

L’authentification par possession d’un objet unique

L’authentification par possession d’un objet unique est une méthode où l’accès est accordé à un utilisateur parce qu’il détient un objet physique ou électronique considéré comme sûr et difficile à reproduire. C’est le principe du « quelque chose que je possède ».

1. Cartes physiques
    - Carte à puce (Smart Card) : utilisée avec un lecteur, souvent protégée par un code PIN.
    - Badge RFID ou NFC : pour le contrôle d’accès aux bâtiments ou systèmes.
    - Token matériel : petit dispositif qui génère des codes temporaires (ex. RSA SecurID).
2. Clés de sécurité électroniques
    - Clés USB de sécurité (YubiKey, Google Titan Key) : conformes aux standards comme FIDO2 / U2F.
    - Elles doivent être insérées ou rapprochées du terminal pour prouver la possession.
3. Codes à usage unique (OTP) générés par un objet
    - Token matériel (afficheur LCD donnant un code renouvelé toutes les 30–60 secondes).
    - Applications mobiles (soft tokens) comme Google Authenticator ou Authy (attention : ce n’est plus un objet strictement physique, mais un objet détenu).

RFID

La technologie RFID repose sur la communication par ondes radio entre deux éléments :

- Un tag RFID (ou badge) → c'est l'objet unique que possède l'utilisateur. Il contient une puce électronique avec un identifiant (UID) et parfois des données supplémentaires.
- Un lecteur RFID → installé à l'entrée d'un bâtiment, sur une machine ou un terminal. Il émet un champ électromagnétique qui alimente le tag (passif) ou communique avec lui (actif).

Processus d'authentification

- L'utilisateur présente son badge RFID devant le lecteur.
- Le lecteur envoie un signal radio qui active la puce.
- Le tag transmet son identifiant ou un code chiffré.
- Le système vérifie cet identifiant dans une base de données ou via un serveur d'authentification.
- Si l'identifiant est valide → l'accès est accordé (ou l'action autorisée).

Types de méthodes RFID en authentification

- Authentification simple : le tag envoie juste son identifiant (peut être cloné facilement si non protégé).
- Authentification mutuelle : le lecteur et le tag échangent des challenges chiffrés pour s'assurer de leur légitimité.
- RFID avec chiffrement (MIFARE DESFire, iCLASS, etc.) : utilisation d'algorithmes cryptographiques (AES, 3DES) pour protéger les échanges.

RFID (Pas pour l'Intra)

1. Structure d'une carte RFID
    - Une carte RFID contient une puce électronique avec :
        - Un identifiant unique (UID) attribué en usine → souvent non modifiable (ROM).
        - Des blocs ou secteurs mémoire (selon le type de carte, ex. MIFARE Classic, DESFire…) → qui peuvent stocker des données personnalisées (numéro d'utilisateur, droits d'accès, etc.).
2. Écriture de l'identifiant
    - UID en lecture seule
        - Gravé par le fabricant, il ne peut pas être modifié.
        - Utilisé comme identifiant principal dans de nombreux systèmes de contrôle d'accès simples.
    - Mémoire réinscriptible
        - Les cartes avec mémoire (MIFARE, iCLASS, etc.) permettent d'écrire un identifiant ou des données personnalisées.
        - L'écriture se fait à l'aide d'un lecteur/enregistreur RFID relié à un logiciel qui envoie une commande d'écriture.
3. Exemple pratique (MIFARE Classic 1K)
    - La carte est organisée en 16 secteurs, chaque secteur contenant 4 blocs. Chaque bloc fait 16 octets.
    - On peut écrire un identifiant utilisateur dans un bloc de données, par exemple sous forme de texte (12345) ou binaire (0x39 0x30 0x31).
    - L'écriture nécessite un lecteur RFID compatible et l'utilisation d'une clé d'accès (A ou B) pour déverrouiller le secteur mémoire.

NFC

Le NFC est une technologie de communication sans fil à courte portée (quelques centimètres, typiquement moins de 10 cm). Il s'agit d'une extension de la technologie RFID haute fréquence (13,56 MHz), mais optimisée pour des usages sécurisés et interactifs entre appareils.

- Communication
    - Deux appareils NFC (par ex. un smartphone et un terminal de paiement) se rapprochent l'un de l'autre.
    - Une communication s'établit par induction électromagnétique via une petite antenne intégrée.
    - Les échanges se font en mode actif (les deux émettent) ou mode passif (un seul émet, l'autre se contente de répondre).
- Modes de fonctionnement du NFC
    - Mode lecteur/écriture
        - Un appareil lit ou écrit des données sur une étiquette NFC (tag).
        - Exemple : scanner un badge ou lire une affiche interactive.
    - Mode peer-to-peer
        - Deux appareils échangent directement des informations.
        - Exemple : partage de fichiers ou contacts entre deux smartphones.
    - Mode émulation de carte
        - Un smartphone se comporte comme une carte à puce sans contact.
        - Exemple : paiement mobile (Google Pay, Apple Pay) ou carte de transport.

Applications principales

- Paiement sans contact (NFC + carte bancaire ou smartphone).
- Titres de transport (cartes de métro/bus, passes électroniques).
- Contrôle d'accès (badges NFC pour entrer dans un bâtiment).
- Partage rapide de données (cartes de visite virtuelles, fichiers).
- Objets connectés (tags NFC pour automatiser des actions sur smartphone).

RFID vs NFC

1. Technologie de base
    - RFID : technologie plus ancienne et générique (radio-identification), utilisée dans le contrôle d'accès, la logistique, les inventaires.
    - NFC : extension de RFID HF (13,56 MHz), conçue pour des usages interactifs et sécurisés entre appareils.

2. Distance de lecture
    - RFID : varie selon le type (LF, HF, UHF) → de quelques centimètres à plusieurs mètres.
    - NFC : portée très courte (≤ 10 cm), ce qui réduit le risque d'interception.

3. Modes de fonctionnement
    - RFID : le tag répond passivement au lecteur.
    - NFC : trois modes possibles :
        - Lecture/écriture (comme RFID),
        - Peer-to-peer (deux appareils échangent),
        - Émulation de carte (un smartphone agit comme badge).

4. Sécurité pour l'authentification
    - RFID :
        - Les systèmes simples (ex. MIFARE Classic) envoient juste un identifiant unique (UID), parfois clonable → vulnérable.
        - Les versions avancées (DESFire, iCLASS) intègrent du chiffrement (AES, 3DES).
    - NFC :
        - Intègre par défaut des mécanismes de sécurité renforcés.
        - Supporte des standards modernes (ex. FIDO2, paiements sans contact).
        - Plus adapté à l'authentification forte (ex. smartphone + biométrie).

5. Usages typiques
    - RFID : contrôle d'accès (portes, badges), inventaire, transport de marchandises.
    - NFC : paiement mobile, transport public, badges sécurisés, authentification multi-facteur (smartphone + empreinte digitale).

L’authentification biométrique statique

L’authentification biométrique statique est une méthode de reconnaissance basée sur des caractéristiques physiques immuables d’un individu. Elle fait partie de la catégorie “quelque chose que je suis” (en opposition à “quelque chose que je sais” – mot de passe, ou “quelque chose que je possède” – badge).

- Principe
  - On capture une caractéristique biologique unique et stable (empreinte digitale, visage, iris, rétine, voix, etc.).
  - On extrait ses caractéristiques distinctives (vecteurs ou gabarits biométriques).
  - On les stocke dans une base de données sécurisée ou sur un support (ex. carte à puce).
  - Lors de l’authentification, on compare l’échantillon capturé en temps réel avec le modèle de référence.
- Exemples de biométrie statique
  - Empreinte digitale : la plus répandue, utilisée dans les smartphones et contrôles d’accès.
  - Reconnaissance faciale : caméras analysent la géométrie du visage.
  - Iris / rétine : très précis, utilisé dans les environnements à haute sécurité.
  - Reconnaissance de la main : forme de la main, géométrie des doigts.
- Avantages
  - Difficile à falsifier (unique et stable).
  - Pratique (pas besoin de retenir un mot de passe ni de posséder un objet).
  - Rapide et de plus en plus intégré dans les appareils du quotidien.
- Limites
  - Non révocable : si une empreinte digitale est copiée, on ne peut pas “changer d’empreinte”.
  - Vie privée : stockage centralisé peut poser problème (risques de fuite).
  - Conditions environnementales : reconnaissance faciale peut échouer en faible luminosité, empreinte digitale si le doigt est humide ou abîmé.
  - Attaques : usurpation via photos, masques, ou empreintes reproduites

L’authentification biométrique statique

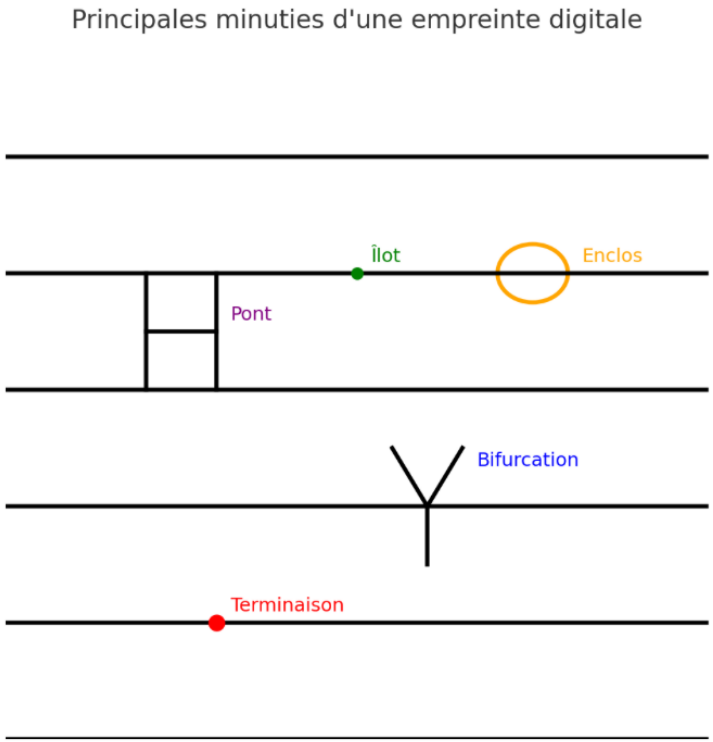
Les caractéristiques d’une empreinte digitale correspondent aux éléments distinctifs utilisés pour identifier une personne. Elles se regroupent en trois grandes catégories :

1. Caractéristiques globales (niveau macroscopique)
    - Ces caractéristiques décrivent la forme générale des crêtes :
        - Boucles (loops) : les crêtes forment une boucle.
        - Arches (arches) : les crêtes passent d’un côté à l’autre sans revenir.
        - Verticilles (whorls) : les crêtes forment des cercles ou spirales.
    - Ces motifs globaux permettent de classifier les empreintes en grandes familles.
2. Caractéristiques locales (minuties)
    - Les minuties sont les points particuliers qui rendent chaque empreinte unique :
        - Terminaison de crête : fin d’une ligne de crête.
        - Bifurcation : une crête se sépare en deux.
        - Îlot (ou point) : petite crête isolée.
        - Pont (bridge) : connexion entre deux crêtes.
        - Crochet (hook) : crête qui se termine en boucle courte.
        - Enclos (enclosure) : une crête forme un cercle autour d’un espace vide.
    - Ce sont ces minuties (20 à 40 en moyenne par empreinte) qui sont extraites et comparées lors d’une authentification.
3. Caractéristiques quantitatives
    - Densité des crêtes : nombre de lignes de crête par millimètre.
    - Position des minuties : coordonnées spatiales uniques.
    - Orientation des crêtes : inclinaison locale des lignes.

L’authentification biométrique statique (Pas pour l’intra)

Ces minuties sont les **caractéristiques locales uniques** utilisées dans les systèmes d’authentification par empreinte digitale

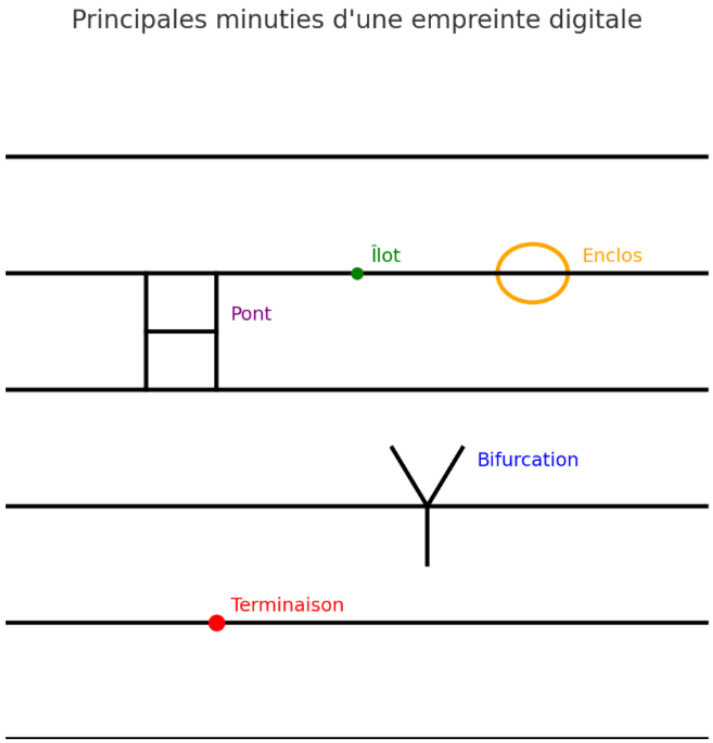
- **Terminaison** : fin d’une crête.
- **Bifurcation** : une crête qui se divise en deux.
- **Îlot** : petit point isolé.
- **Pont** : connexion entre deux crêtes.
- **Enclos** : boucle fermée autour d’un espace.



L’authentification biométrique statique (Pas pour l’intra)

Ces minuties sont les **caractéristiques locales uniques** utilisées dans les systèmes d’authentification par empreinte digitale

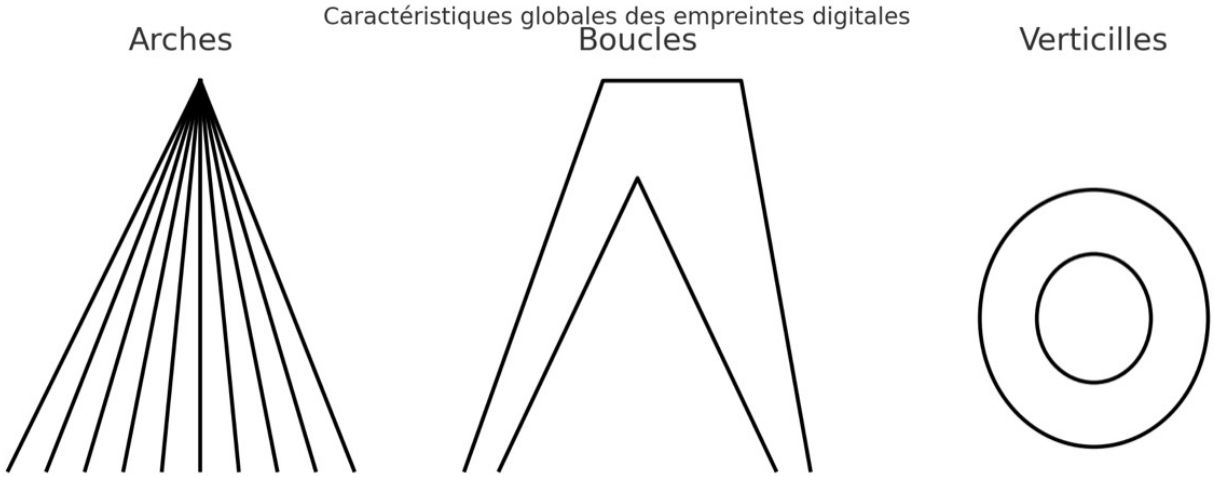
- **Terminaison** : fin d’une crête.
- **Bifurcation** : une crête qui se divise en deux.
- **Îlot** : petit point isolé.
- **Pont** : connexion entre deux crêtes.
- **Enclos** : boucle fermée autour d’un espace.



L’authentification biométrique statique (Pas pour l’intra)

Ces motifs sont utilisés pour classifier les empreintes en grandes familles avant d’analyser les minuties locales(terminaisons, bifurcations, îlots, etc.)

- Arches : les crêtes traversent le doigt d’un côté à l’autre sans former de boucle.
- Boucles : les crêtes s’incurvent et reviennent en arrière pour former une boucle.
- Verticilles : les crêtes s’enroulent en cercles ou spirales concentriques.



L’authentification biométrique statique (Pas pour l’intra)

Exemple d'empreinte digitale avec minuties annotées

L’authentification biométrique statique (Pas pour l’intra)

Quelle est la probabilité de deux personnes avoir la même empreinte digitale ?

- Les motifs globaux (arches/boucles/verticilles) ne suffisent pas : ~65 % boucles, ~30 % verticilles, ~5 % arches — beaucoup de personnes se ressemblent à ce niveau.
- L’identité se joue sur les minuties (terminaisons, bifurcations, etc.) et leurs positions. Avec des dizaines de minuties, la combinaison devient astronomique.

En pratique, on parle de FMR : probabilité qu’un système déclare à tort “match” pour deux doigts différents au seuil choisi.

Systèmes courants : FMR ~ 10^{-6} à 10^{-8} (1 fausse acceptation sur 1 à 100 millions de comparaisons).

Systèmes haut de gamme/laboratoire : FMR encore plus bas (jusqu’à 10^{-12})

.

Donc, la probabilité que deux personnes au hasard aient une empreinte “assez similaire pour être acceptée” ≈ le FMR du système utilisé—notamment si on compare un seul doigt. Utiliser plusieurs doigts fait chuter la probabilité (ex. deux doigts indépendants à FMR 10^{-6} ⇒ ~10^{-12}).

Authentification Iris et rétine

Authentification par l'iris

Analyse du motif coloré et de la texture de l'iris (la partie circulaire autour de la pupille).

Caractéristiques : chaque iris est unique, même entre jumeaux.

Avantages :

Très fiable (taux d'erreur extrêmement bas).

Moins intrusif : on peut capturer une image de l'iris avec une caméra infrarouge à distance raisonnable (quelques cm à quelques mètres).

Plus pratique pour l'utilisateur.

Inconvénients :

Peut être affecté par des lunettes ou lentilles.

Nécessite un équipement spécialisé.

Authentification par la rétine

Scan du réseau de vaisseaux sanguins au fond de l'œil, capturé via un faisceau lumineux.

Caractéristiques : modèle très difficile à falsifier, unique à chaque individu.

Avantages :

Extrêmement précis (encore plus que l'iris).

Très difficile à contourner.

Inconvénients :

Très intrusif (il faut approcher l'œil d'un appareil qui projette un faisceau lumineux).

Moins confortable pour l'utilisateur → adoption limitée.

Plus coûteux et plus lent que le scan d'iris.

Usage

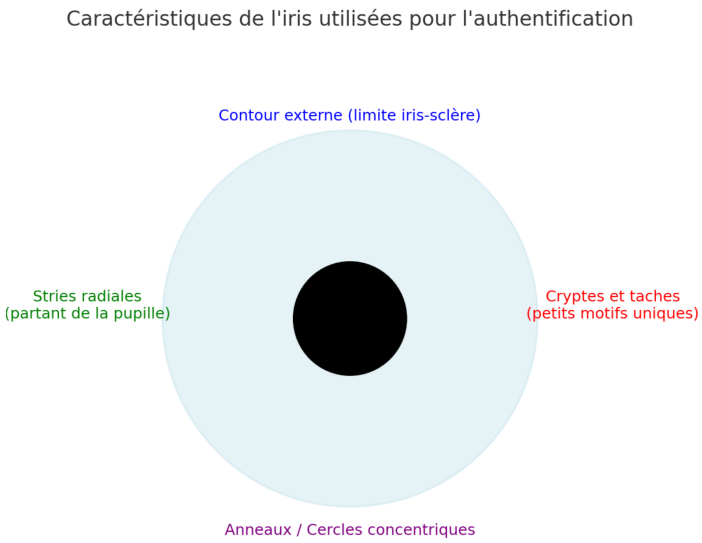
- Iris : beaucoup plus utilisé dans les systèmes commerciaux et gouvernementaux (contrôle aux frontières, smartphones haut de gamme, systèmes de sécurité).
- Rétine : rare en dehors de cas très spécifiques nécessitant une sécurité maximale (militaire, laboratoires sensibles).

Authentification Iris vs Rétine

Critère	Iris 👁️	Rétine 👁️‍🗨️
Principe	Analyse de la texture et des motifs de l'iris (partie colorée de l'œil).	Analyse du réseau de vaisseaux sanguins au fond de l'œil.
Précision	Très élevée (faible taux d'erreur).	Extrêmement élevée (plus précis que l'iris).
Confort	Relativement confortable : caméra infrarouge à distance.	Peu confortable : nécessite de rapprocher l'œil d'un scanner avec faisceau lumineux.
Vitesse	Rapide (quelques secondes).	Plus lent (procédure plus intrusive).
Coût	Moyen, largement disponible (caméras spécialisées, intégrées parfois aux smartphones).	Élevé, équipements spécifiques et rares.
Sécurité	Très bonne, difficile à falsifier.	Exceptionnelle, pratiquement impossible à falsifier.
Adoption	Couramment utilisé (contrôle aux frontières, smartphones, systèmes d'accès).	Rare, réservé à des environnements très sensibles (militaire, laboratoires).

Authentification Iris – caractéristiques

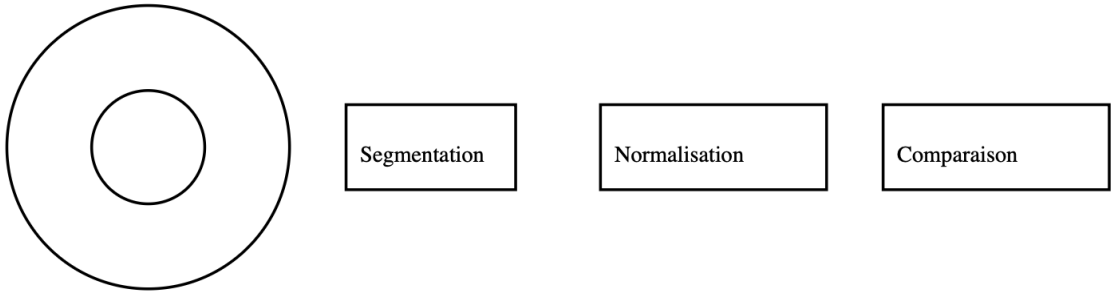
- Motifs texturaux
  - Les anneaux concentriques, stries radiales, taches pigmentées ou taches cryptiques.
  - Ces structures microscopiques forment une signature unique.
- Contraste et granularité
  - Variation d'intensité et de texture capturée par une caméra infrarouge.
  - La complexité permet de générer un gabarit difficile à reproduire.
- Structures géométriques
  - Forme du contour de l'iris (limite pupille–iris et iris–sclère).
  - La distance et la distribution spatiale des motifs internes.
- Caractéristiques invariantes
  - Contrairement à l'empreinte digitale, l'iris reste stable toute la vie.
  - Ne change pas avec l'âge (sauf maladies oculaires rares).



Authentification Iris (pas pour l'Intra)

L'authentification par l'iris repose sur plusieurs étapes clés :

1. Capture : une caméra infrarouge prend l'image de l'œil.
2. Segmentation : l'iris est isolé de la pupille et du reste de l'œil.
3. Normalisation : l'image est transformée dans une taille standard.
4. Extraction des caractéristiques : les motifs uniques de l'iris sont convertis en un vecteur numérique.
5. Comparaison : le vecteur est comparé à une base de données pour vérifier l'identité.



Authentification Iris (pas pour l'Intra)

Quelle est la probabilité de deux personnes avoir la même authentification avec iris ?

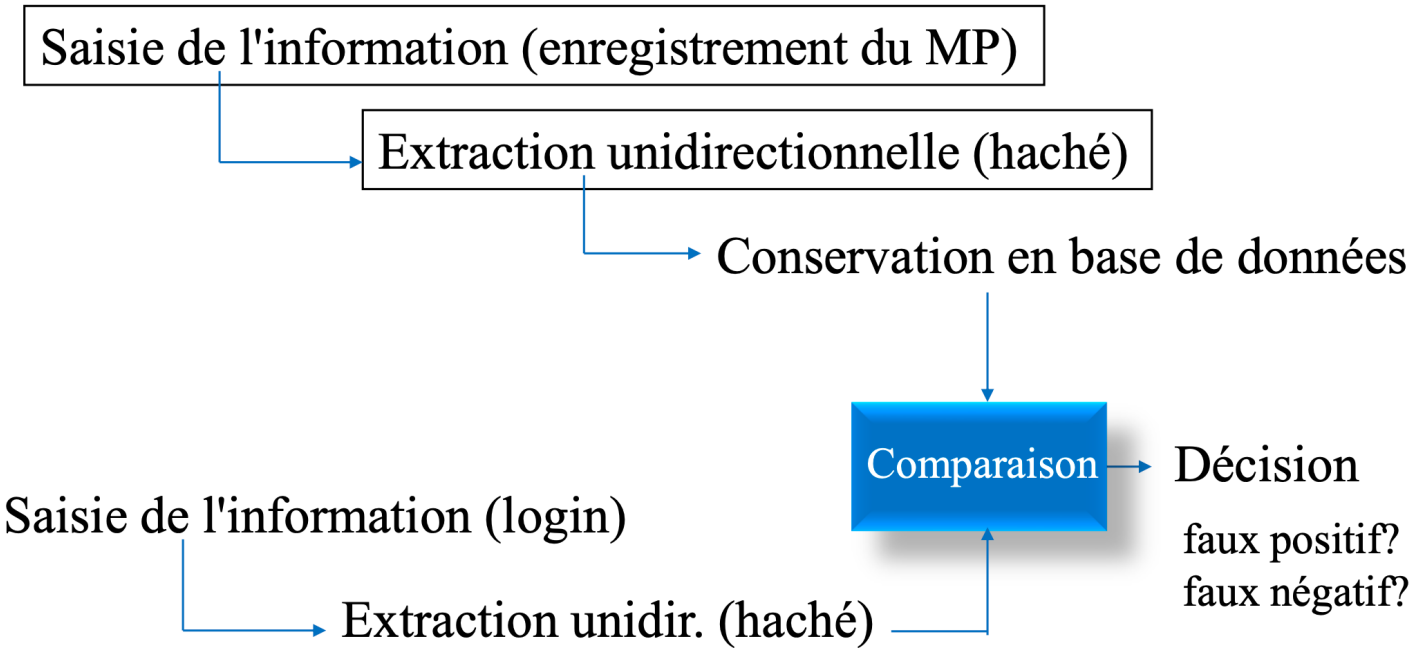
- Les systèmes d'iris bien réglés atteignent couramment un FMR de 10^{-8} à 10^{-12} (1 fausse acceptation sur 100 millions à 1 000 milliards de comparaisons).
- Avec les deux yeux (supposés indépendants), la probabilité se multiplie : p(ex. FMR 10^{-9} par œil) ⇒ $\approx 10^{-18}$ pour "iris gauche ET droit" en même temps.

Ce que signifie "deux personnes ont la même authentification"

- Pour une comparaison 1:1 (A contre B), la probabilité qu'un système les déclare identiques ≈ FMR au seuil choisi.
- Pour une recherche 1:N (A contre une base de N personnes), la probabilité d'au moins une fausse acceptation ≈ N×FMR (approximation valable si N×FMR $\ll 1$).
  Ex. FMR $= 10^{-9}$ et base de 10^{6} personnes ⇒ ~10^{-3} (% 0,1) de faux "match" en moyenne par recherche.

Avec un capteur correct, un bon éclairage IR et un seuil strict, la probabilité que deux personnes différentes soient acceptées comme la même via l'iris est extrêmement faible (jusqu'à 10^{-12} par comparaison).

Authentification par mot de passe - modèle général



Linux : Hash et mot de passe – (pas pour l'intra)

/etc/shadow (Linux)

```
root:$6$rounds=10000$saltexemple$HshWk...fE1:19345:0:99999:7:::
alice:$5$saltsha256$Ff9d3c...AbY/:19346:0:99999:7:::
bob:$2b$12$abcdefghijklmnopqrstuv$Nq...zQy:19347:0:99999:7:::
carol:$argon2id$v=19$m=65536,t=3,p=4$base64salt$base64hash:19348:0:99999:7:::
```

- `root` → SHA-512
- `alice` → SHA-256
- `bob` → bcrypt
- `carol` → Argon2id

Fonction de hash par défaut utilisée par Linux

- Sur la plupart des distributions Linux modernes, le hachage des mots de passe dans /etc/shadow est réalisé via la fonction crypt(3) (implémentation fournie par glibc ou libxcrypt).
- Le schéma par défaut courant est SHA-512, identifié par le préfixe 6 dans la chaîne de hachage stockée dans /etc/shadow.
- la longueur d'un "hash de mot de passe" sous Linux n'est pas fixe, elle dépend de l'algorithme (et du format d'encodage) utilisé pour produire la chaîne stockée dans /etc/shadow. Voici les cas courants et leurs longueurs typiques (Formats courants dans /etc/shadow) :
  - SHA-512-crypt (6…$hash)
    - Partie « hash » (la dernière portion après le dernier $) : 86 caractères (alphabet ./0-9A-Za-z, encodage modifié base64).
    - Entrée complète : "6rounds=5000$salt$<86 chars>" (longueur totale variable à cause du sel et des paramètres).
  - SHA-256-crypt (5…$hash)
    - Partie hash : 43 caractères.
  - MD5-crypt (1…$hash)
    - Partie hash : 22 caractères.
  - bcrypt ($2b$cost$salt+hash)
    - Chaîne complète : 60 caractères (forme standard, inclut version, coût, sel et hash).

Attaques - Mot de passe

Entropie d'un mot de passe (symboles en octets). Source markovienne d'ordre 0 (chaque symbole est indépendant).

Entropie d'un symbole d'un alphabet de N symboles (i.e. alphabet = {A, B, C, …, Z}, N = 26) :

$$H = -\sum_{i=1}^{N} p_i \log_2(p_i) \quad = -N \cdot \frac{1}{N} \cdot \log_2\left(\frac{1}{N}\right) \quad = \log_2(N)$$

Entropie d'un mot de passe de longueur M composé des symboles d'un alphabet de taille N

Application aux mots de passe

- Prenons un alphabet de **62 symboles possibles** (A–Z, a–z, 0–9).
- Entropie maximale par caractère = $log_2(62) \approx 5{,}95$ bits.

- Un mot de passe de 8 caractères **totalement aléatoire** → $8 \times 5{,}95 \approx 47{,}6$ bits.
- Un mot de passe de 12 caractères aléatoires → $12 \times 5{,}95 \approx 71{,}4$ bits.
- Cela veut dire que le nombre de combinaisons possibles est $2^{71} \approx 2{,}36 \times 10^{21}$,donc presque impossible à casser par force brute.

Qu'est-ce que ça mesure ?

- L'entropie en bits mesure la **quantité d'incertitude** ou d'**imprévisibilité**.
- En sécurité, cela correspond au **nombre de tentatives qu'un attaquant doit faire, en moyenne, pour deviner le mot de passe** s'il ne connaît rien d'autre.

2. Exemple simple

- Un mot de passe de **1 bit d'entropie** → 2 possibilités (comme pile ou face).
- 2 bits d'entropie → 4 possibilités.
- 10 bits d'entropie → environ $2^{10} = 1024$ possibilités.
- 20 bits d'entropie → environ $2^{20} \approx 1million$ possibilités.
- 40 bits d'entropie → environ $2^{40} \approx 1trillion$ possibilités.
- Donc, **plus le nombre de bits est élevé, plus le mot de passe est difficile à deviner par force brute**.

Mots de passes les plus souvent utilisés

Source: https://wpengine.com/resources/passwords-unmasked-infographic/

Et les caractères ! @ # $ % & * ? ne sont pas utilisés

Mots de passes les plus souvent utilisés

What Are the 50 Most Common Passwords?

Based on most common duplicate passwords within a breach of over 30 million accounts.

SecurityScorecard

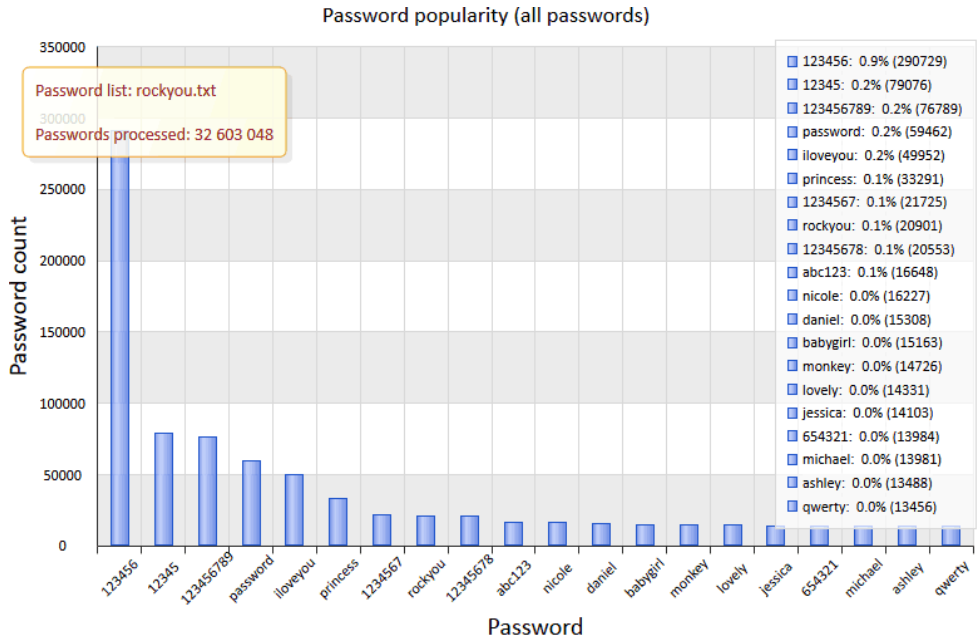
1.	123456	11.	123321	21.	222222	31.	333333	41.	password1
2.	123456789	12.	1q2w3e4r5t	22.	112233	32.	123qwe	42.	q1w2e3r4
3.	qwerty	13.	iloveyou	23.	abc123	33.	159753	43.	qqww1122
4.	password	14.	1234	24.	999999	34.	q1w2e3r4t5y6	44.	sunshine
5.	1234567	15.	666666	25.	777777	35.	987654321	45.	zxcvbnm
6.	12345678	16.	654321	26.	qwerty123	36.	1q2w3e	46.	1qaz2wsx3edc
7.	12345	17.	555555	27.	qwertyuiop	37.	michael	47.	liverpool
8.	1234567890	18.	gfhjkm	28.	888888	38.	lovely	48.	monkey
9.	111111	19.	7777777	29.	princess	39.	123	49.	1234qwer
10.	123123	20.	1q2w3e4r	30.	1qaz2wsx	40.	qwe123	50.	computer

Source: https://securityscorecard.com/blog/worlds-worst-passwords

Et les caractères ! @ # $ % & * ? ne sont pas utilisés

Attaque par force brute

- A
- B
- C
- …
- Z
- AA
- AB
- …
- ZZZZZZZZZZZZZZZZZZZZZZZZZZ
  ZZZZZZZZZZZZZZ



Attaque par dictionnaire

Source:
https://github.com/danielmiessler/SecLists/tree/master/Passwords

https://weakpass.com/wordlists/rockyou.txt

Les règles

- Ajouter un chiffre à la fin du mot de passe
- Ajouter deux chiffres à la fin du mot de passe
- Ajouter quatre chiffres à la fin du mot de passe
- Ajouter un caractère spécial à la fin du mot de passe
- Ajouter un chiffre au début du mot de passe
- Remplacer les *i* par des 1
- Mettre en majuscule la première lettre du mot de passe

Force brute et John the Ripper

- 123456
- password
- qwerty
- azerty
- welcome
- iloveyou
- admin
- dragon
- football
- monkey
- shadow
- princess
- 123456789
- letmein

John the Ripper va tester ces mots tels quels.
Ensuite, il peut appliquer des règles de transformation (par exemple ajouter 123, mettre en majuscule la première lettre, remplacer a par @, etc.).
Exemple : password → Password1 → P@ssword!

Lien avec l'entropie

Si un mot de passe figure dans un dictionnaire, son entropie réelle est très faible : ce n'est pas un mot de passe aléatoire.
L'attaque par dictionnaire exploite précisément le fait que les humains ne choisissent pas leurs mots de passe comme une source markovienne d'ordre 0 uniforme, mais plutôt en fonction de mots communs → faible entropie effective.

Zero-Knowledge Proofs - Fiat–Shamir (pas pour l'intra)

Protocole d'authentification Fiat–Shamir (ZKP)

Rôles

- Prouver (Alice) → veut prouver qu'elle connaît un secret.
- Vérificateur (Serveur) → doit être convaincu sans jamais apprendre le secret.

Étapes

- Initialisation (clé secrète)
  Alice choisit un secret s et un grand nombre premier n.
  Elle publie $v = s^2 \mod n$.
  s reste secret.
- Engagement
  Alice choisit un nombre aléatoire r.
  Elle envoie $x = r^2 \mod n$ au serveur.
- Challenge
  Le serveur envoie un bit aléatoire $c \in \{0, 1\}$.
- Réponse
  Si $c = 0$, Alice répond $y = r$.
  Si $c = 1$, Alice répond $y = r \cdot s \mod n$.

Vérification

Si $c = 0$, le serveur vérifie que $y^2 \equiv x \mod n$.
Si $c = 1$, le serveur vérifie que $y^2 \equiv x \cdot v \mod n$.

Si Alice réussit plusieurs rounds consécutifs, le serveur est convaincu qu'elle connaît s, mais n'a rien appris sur slui-même.

Contrôle d’accès

Linux → **chmod 754 fichier**

- Les permissions s’expriment en trois chiffres octaux (propriétaire, groupe, autres) :
  - 7 → propriétaire : lecture (r=4), écriture (w=2), exécution (x=1) → rwx
  - 5 → groupe : lecture (r=4) + exécution (x=1) → r-x
  - 4 → autres : lecture (r=4) → r--

Contrôle d’accès

En Linux, le **sticky bit** est un droit spécial qu’on peut appliquer à un répertoire (et, historiquement, à des fichiers).

- Quand le sticky bit est activé sur un répertoire, les fichiers qui s’y trouvent ne peuvent être supprimés ou renommés que par :
  - le propriétaire du fichier,
  - le propriétaire du répertoire,
  - ou l’utilisateur root.
- Sans ce bit, tout utilisateur ayant les droits d’écriture sur le répertoire pourrait supprimer ou renommer n’importe quel fichier qui s’y trouve.
- Exemple : ls -ld /tmp
  - drwxrwxrwt 10 root root 4096 Sep 30 12:00 /tmp
  - Les permissions montrent t à la fin : drwxrwxrw**t**
  - Cela signifie que le sticky bit est actif.
  - Résultat : chaque utilisateur peut créer ses fichiers temporaires, mais ne peut pas supprimer ceux des autres.

Contrôle d’accès

setuid est un bit de permission spécial appliqué à un fichier exécutable.

- Quand un utilisateur exécute ce fichier, il hérite temporairement des privilèges du propriétaire du fichier (au lieu de ses propres privilèges).
- Cela permet à un programme d’effectuer des actions nécessitant des droits plus élevés que ceux de l’utilisateur qui le lance.

Exemple classique

- /usr/bin/passwd (qui sert à changer son mot de passe) a le setuid activé :
- $ ls -l /usr/bin/passwd
  - -rwsr-xr-x 1 root root 54256 Sep 30 12:00 /usr/bin/passwd
- Remarque la lettre s dans -rwsr-xr-x → le s dans la partie des permissions utilisateur (u).
- Ici, le fichier appartient à root, donc quand un utilisateur normal exécute passwd, le programme s’exécute avec les privilèges root.
- C’est nécessaire pour que l’utilisateur puisse modifier le fichier /etc/shadow, qui est normalement uniquement accessible à root.

Contrôle d’accès - DAC vs MAC

Critère	**DAC (Contrôle d’accès discrétionnaire)**	**MAC (Contrôle d’accès obligatoire – SELinux, AppArmor)**
Qui décide des accès ?	Le **propriétaire du fichier** (user) choisit les permissions avec `chmod`, `chown`, `chgrp`.	Le **système** (politique de sécurité centralisée) impose les règles. Même root doit obéir.
Principe	Accès basé sur **user / group / others** et leurs droits (rwx).	Accès basé sur des **politiques de sécurité** prédéfinies (labels, profils, règles).
Flexibilité	Simple et rapide à gérer, mais dépend de la vigilance des utilisateurs.	Plus strict et granulaire (ex. : tel programme peut lire un fichier mais pas le modifier).
Exemple typique	Un fichier `-rw-r--r-- alice dev fichier.txt` : Alice décide qui lit/écrit.	SELinux peut dire : "même si Alice est propriétaire, son programme n’a pas le droit d’accéder à ce fichier sensible".
Root	Root peut tout faire (bypass complet).	Même root est soumis aux règles (root ≠ omnipotent).
Sécurité	Suffisante dans un contexte simple.	Beaucoup plus sécurisée contre les erreurs humaines ou les exploits (isolation stricte).
Utilisation	Par défaut dans toutes les distributions Linux.	Optionnelle, activée sur certaines distributions (Fedora, RHEL, Ubuntu avec AppArmor).

Bell & LaPadula

Le modèle de Bell & LaPadula (1973) est un modèle de contrôle d'accès obligatoire (MAC) centré sur la confidentialité des données.

- Son objectif : empêcher les fuites d'information d'un niveau de sécurité élevé vers un niveau plus bas.
- Il s'applique souvent à des environnements classifiés (par ex. : Top Secret, Secret, Confidentiel, Non classifié).

Principes de base

- Le modèle repose sur deux règles fondamentales :
  - "No Read Up" – pas de lecture vers le haut
    → Un sujet (utilisateur, processus) ne peut pas lire des données d'un niveau plus élevé que son autorisation.
    Exemple : un utilisateur Secret ne peut pas lire un document Top Secret.
  - "No Write Down" – pas d'écriture vers le bas
    → Un sujet ne peut pas écrire vers un niveau de sécurité inférieur.
    Exemple : un utilisateur Top Secret ne peut pas écrire un fichier en Secret (risque de fuite).

Exemple

- Imaginons 3 niveaux :
  - Top Secret
  - Secret
  - Confidentiel
  - Alice (autorisation Secret) :
    - ✅ peut lire Secret et Confidentiel.
    - ❌ ne peut pas lire Top Secret (No Read Up).
    - ✅ peut écrire Secret et Top Secret.
    - ❌ ne peut pas écrire en Confidentiel (No Write Down).

ABAC

ABAC = Attribute-Based Access Control

- C’est un modèle de contrôle d’accès où les permissions sont déterminées non pas seulement par l’utilisateur ou son rôle, mais par un ensemble d’attributs associés :
  - Attributs du sujet (ex. : identité, rôle, âge, grade de sécurité, localisation)
  - Attributs de la ressource (ex. : type de fichier, classification, propriétaire)
  - Attributs de l’environnement/contexte (ex. : heure, adresse IP, appareil utilisé, réseau interne/externe)
- L’accès est décidé par une politique qui évalue ces attributs.

Exemple

- Politique : "Un utilisateur peut accéder à un dossier médical uniquement si :
  - (1) son rôle est médecin,
  - (2) il est dans le service cardiologie,
  - (3) il se connecte depuis l’hôpital,
  - (4) et il est entre 8h et 18h."
- Ici, l’autorisation dépend de plusieurs attributs combinés.

Contrôle d’accès - DAC vs MAC vs RBAC vs ABAC

Modèle	Principe	Exemple	Avantages	Limites
DAC(Discretionary Access Control)	Le **propriétaire** d’une ressource décide qui a accès (permissions rwx).	Alice met `chmod 640` sur son fichier.	Simple, intuitif.	Trop permissif, root peut tout, erreurs humaines.
MAC(Mandatory Access Control)	Le **système impose** des règles centralisées (politiques de sécurité).	SELinux bloque un programme même si l’utilisateur est propriétaire.	Très sécurisé, adapté aux environnements sensibles.	Rigide, complexe à administrer.
RBAC (Role-Based Access Control)	Les droits sont attribués selon des **rôles**.	Un "admin" peut créer des comptes, un "lecteur" peut seulement lire.	Facile à gérer à grande échelle, adapté aux entreprises.	Peu flexible si besoin de conditions spécifiques.
ABAC(Attribute-Based Access Control)	Les droits dépendent de **plusieurs attributs** (utilisateur, ressource, contexte, environnement).	"Un médecin du service cardio peut consulter les dossiers entre 8h-18h depuis l’hôpital".	Très flexible, granulaire, adapté au cloud et à la mobilité.	Complexité élevée, besoin d’un moteur de règles centralisé.

Contrôle d’accès - DAC vs MAC vs RBAC vs ABAC

En résumé :

DAC = je décide
MAC = le système décide
RBAC = ton rôle décide
ABAC = tes attributs et ton contexte décident

IAM = Identity and Access Management

IAM est un ensemble de politiques, processus et technologies qui permettent de :

- Identifier les utilisateurs (humains ou machines).
- Contrôler leurs accès aux systèmes, applications et données.
- Garantir que chaque utilisateur a le bon niveau d'accès (principe du moindre privilège).

Composants clés de l'IAM

- Identification & authentification
  - Qui est l'utilisateur ?
  - Ex. : login/mot de passe, MFA (authentification multifacteur), biométrie, certificats.
- Autorisation
  - Que peut-il faire ?
  - Ex. : lecture seule, modification, accès limité par rôle (RBAC), attributs (ABAC).
- Gestion des identités
  - Création, modification, désactivation des comptes.
- Audit & traçabilité
  - Suivi des connexions, logs, conformité réglementaire (ex. RGPD, HIPAA).

Exemple

Dans une entreprise l'IAM gère que :

- Paul (employé RH) accède aux dossiers du personnel.
- Marie (développeuse) accède au code source, mais pas aux dossiers RH.
- Les anciens employés perdent immédiatement leurs accès.
- L'authentification est obligatoire pour tout accès externe