

ENTROPIE DE Argon2ID combiné aux primitives du NIST

La Force du générateur de mots de passe sécurisé par Argon2id dans CryptoToolkit™

en Dehors du Fait qu'il soit BIP-39, Ce générateur de mots de passe possède un curseur extensible jusqu'à 128 caractères (839 bits) avec 4 critères (Alpha, num, Maj/min, Caractères spéciaux, le tout avec un coffre fort extensible de 50 emplacement de stockage de PassPhrase ou de mot de passe.

Les avantages d'une telle application dans l'application face à la vulnérabilité des mots de passe actuels qui sont cassable en un claquement de doigt par les cartes graphiques puissantes.

* Comme vu précédemment, les IA actuelles cassent les mots de passe simples en quelques minutes voire secondes

* L'utilisation d'Argon2id (le meilleur algorithme de hachage de mots de passe au monde) force l'ordinateur ou l'IA du pirate à consommer énormément de mémoire et de temps pour tester une seule passphrase.

Cela rend les attaques de type "force brute" mathématiquement et financièrement impossibles..

démonstration

Entropie (bits) = longueur × $\log_2$ (taille de l'alphabet)

Avec les 4 critères activés, l'alphabet est l'ASCII imprimable :

Table

Critère	Caractères
Minuscules (a-z)	26
Majuscules (A-Z)	26
Chiffres (0-9)	10
Spéciaux (!@#\$\$%...)	~32
Total	~94

$\log_2(94) \approx 6,55$ bits par caractère

Ce que ça donne concrètement

- **128 caractères** × 6,55 = **~839 bits d'entropie** (pas 256 !)
- Pour atteindre **256 bits**, il suffit de **~39 caractères** ($256 \div 6,55 \approx 39$)
- Pour atteindre **128 bits** (déjà incassable en pratique) : **~20 caractères**

Deux précisions importantes pour la documentation

- **Argon2id ne crée pas d'entropie.** L'entropie vient uniquement du générateur aléatoire (CSPRNG). Argon2id protège le *stockage* du mot de passe maître en rendant chaque tentative de cassage coûteuse en mémoire/temps — c'est exact, mais c'est un mécanisme distinct de la force du mot de passe généré.
- **BIP-39** : la référence « 256 bits » vient probablement de là — une phrase mnémonique BIP-39 de 24 mots encode exactement 256 bits d'entropie. Si votre générateur supporte BIP-39, c'est là que le chiffre 256 est justifié.

Le générateur de mots de passe d'un outil comme CryptoToolkit™ combine une **entropie maximale** (via la génération de caractères) et une **résistance extrême aux attaques** (via le hachage Argon2id).

La force de ce système face aux vulnérabilités actuelles.



Pourquoi les mots de passe actuels "fondent" face aux IA

- **Vitesse de calcul phénoménale** : Les cartes graphiques (GPU) modernes et les IA optimisées testent des milliards de combinaisons par seconde sur des hachages classiques (comme MD5 ou SHA-256).
- **Prévisibilité humaine** : Les humains créent des variations prévisibles (ex: Motdepasse123!). Les IA entraînées sur des milliards de fuites de données devinent ces schémas en quelques minutes.



La force brute neutralisée : L'effet de curseur (Jusqu'à 128 caractères)

- **Entropie gigantesque** : Un mot de passe de 128 caractères combinant les 4 critères (Majuscules, minuscules, chiffres, caractères spéciaux) puise dans un alphabet d'environ 90 à 95 caractères.
- **Mathématiquement inviolable** : Cela représente 95^{128} combinaisons possibles (bien au-delà de 256 bits d'entropie).
- **Temps de calcul universel** : Même en alignant tous les superordinateurs et les IA de la Terre, casser une telle clé par force brute prendrait des milliards d'années.



Le bouclier ultime : Argon2id

Si ce générateur utilise **Argon2id** pour dériver ou sécuriser les clés/passphrases de son coffre-fort, le paradigme change totalement pour le pirate :

- **Goulot d'étranglement mémoire (Memory-Hard)** : Contrairement aux anciens algorithmes, Argon2id exige une quantité massive de mémoire RAM pour valider une seule tentative.
- **Anti-GPU et Anti-ASIC** : Les pirates utilisent des milliers de puces en parallèle (GPU/ASIC) pour aller vite. Argon2id bloque cela, car ces puces ont beaucoup de cœurs de calcul, mais très peu de mémoire dédiée par cœur.
- **Résistance aux attaques de canaux auxiliaires** : La variante "id" combine la résistance aux attaques temporelles (Argon2i) et aux attaques par GPU (Argon2d).
- **Asphyxie financière** : Tester une supposition ne prend plus une microseconde, mais par exemple 0,5 seconde et demande 1 Go de RAM. Pour un pirate, tenter de craquer une passphrase Argon2id devient financièrement et matériellement impossible (facture d'électricité et de serveurs astronomique).

Les avantages de l'intégration "Application dans l'application"

Avoir ce générateur et ce coffre de 50 emplacements directement intégrés dans CryptoToolkit™ offre des bénéfices majeurs :

- **Zéro fuite par le presse-papier** : Pas besoin de copier-coller vos mots de passe ultra-longes depuis un site web tiers ou un bloc-notes non sécurisé, évitant les virus "keyloggers" ou "clipboard hijackers".
- **Souveraineté des données** : Le coffre de 50 emplacements est local, chiffré par Argon2id, et ne dépend pas d'un serveur cloud vulnérable aux piratages de masse (comme les fuites historiques d'autres gestionnaires en ligne).
- **Complémentarité BIP-39 / Passphrase** : Vous pouvez générer une clé de chiffrement maîtresse via le standard de sécurité crypto (BIP-39), puis utiliser le gestionnaire Argon2id pour stocker les "Passphrases" complexes (le 13ème ou 25ème mot) qui protègent vos portefeuilles crypto contre le vol physique.

calculons ensemble le temps théorique nécessaire pour qu'une IA grand public puisse forcer l'un de ces mots de passe de 128 caractères:

[Voici le calcul de l'entropie et du temps théorique, suivi des recommandations de configuration pour Argon2id.](#)



Partie 1 : Calcul du temps théorique de force brute

Pour un mot de passe de **128 caractères** utilisant les **4 critères**, l'alphabet disponible (N) est d'environ **95 caractères** (26 majuscules + 26 minuscules + 10 chiffres + 33 caractères spéciaux).

1. Le nombre de combinaisons:

Le nombre total de combinaisons possibles est :

$$95^{128} \approx 4,37 \times 10^{253}$$

95^{128} s'écrit en entier :

14 080 610 235 352 233 733 974 796 807 107 488 225 431 294 946 326 679 091 971 204 643
836 621 273 151 625 656 703 817 627 574 060 339 920 877 177 547 088 562 233 219 059 963
291 317 578 633 826 442 480 919 226 256 285 964 090 078 137 242 215 787 111 679 714 448
326 688 998 187 048 752 763 761 427 786 448 621 191 084 384 918 212 890 625

Soit un nombre à 254 chiffres, cohérent avec l'ordre de grandeur $4,37 \times 10^{253}$ affiché sur la capture.

En bits d'entropie, cela représente $\log_2(95^{128}) \approx 841$ bits d'entropie (ce qui dépasse largement le seuil des 256 bits évoqué).

2. La vitesse d'une "IA grand public" / Superordinateur

Même si une IA grand public est rapide pour deviner des mots de passe humains, sur un mot de passe *totalelement aléatoire*, elle doit tester les combinaisons une par une...

- Supposons qu'un pirate utilise une excellente carte graphique grand public (type RTX 4090) optimisée pour tester **100 milliards de combinaisons par seconde** (10^{11} essais/sec)
- Soyons encore plus extrême : imaginons qu'il dispose d'un super-ordinateur capable de tester 1000 péta-essais par seconde (10^{18} essais/sec)

3. Le résultat du temps de calcul

En divisant le nombre de combinaisons par la vitesse du superordinateur :

Voici la formule en texte :

$$4,37 \times 10^{253} \div 10^{18} \text{ essais/sec} = 4,37 \times 10^{235} \text{ secondes}$$

présentation en fraction (numérateur/dénominateur) plutôt qu'avec le signe $\div$, en texte simple ça donne :

$$\frac{4,37 \times 10^{253}}{10^{18} \text{ essais/sec}} = 4,37 \times 10^{235} \text{ secondes}$$

Pour convertir cela en années : $4,37 \times 10^{235} \div (365,25 \times 24 \times 3600) \approx 1,38 \times 10^{228}$ années

Verdict : L'âge de l'Univers est d'environ $1,38 \times 10^{10}$ ans.

Il faudrait donc un temps représentant des milliards de milliards de fois l'âge de l'Univers pour casser ce mot de passe par force brute,

rendant l'attaque mathématiquement obsolète.... 

Et avec un bilan économique désastreux pour l'attaquant