

Émergence prochaine de notre logiciel de Cybersécurité post-quantique anti HNDL et anti Q-day : CryptoToolkit™

CryptoToolkit™ de ARANEA SOFT & AI SYSTEMS (araneasoft.fr) réunit les meilleures armes de défense cryptographique actuelles.

Il s'attaque directement aux plus grandes failles exploitées par les pirates et les intelligences artificielles autonomes : l'interconnexion des réseaux, la faiblesse des mots de passe et l'espionnage à long terme.

Voici ses caractéristiques techniques affichées, ce logiciel est “théoriquement” une réponse parfaite aux menaces de nouvelle génération:

1. Le bouclier ultime contre le HNDL : Le Post-Quantique et la Crypto-Agilité

- **Le danger du HNDL (*Harvest Now, Decrypt Later* ou "Récolter maintenant, décrypter plus tard")** : Aujourd'hui, des pirates et des gouvernements interceptent et stockent des tonnes de données d'entreprises, même si elles sont chiffrées. Ils attendent simplement l'arrivée des ordinateurs quantiques (d'ici quelques années) pour casser les algorithmes actuels (RSA, ECC) et tout lire d'un coup.
- **La parade du soft** : En intégrant les primitives post-quantiques du NIST (comme ML-KEM ou ML-DSA), CryptoToolkit rend les données illisibles, même pour un futur ordinateur quantique.
- **La Crypto-Agilité** : Si une faille est découverte demain dans un algorithme post-quantique, le logiciel est capable de remplacer la brique mathématique instantanément, sans que l'entreprise n'ait à réécrire tout son système.
Le NIST a déjà prévu HQC (Inclus dans CryptoToolkit™) au cas où KYBER-KEM serait compromis un jour lointain...

2. Le chiffrement Hybride et Argon2id : Fin de l'usurpation et du crack par IA

- Comme vu précédemment, les IA actuelles cassent les mots de passe simples en quelques minutes.
- 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.
- Le **chiffrement hybride** combine la vitesse du chiffrement symétrique (pour les gros fichiers) et la sécurité du chiffrement asymétrique (pour s'échanger les clés), garantissant qu'aucune usurpation d'identité ne permette de déverrouiller les fichiers.

3. Le Zéro-Knowledge et l'Air-Gapped : La fin des fuites de données

- **Zéro-Knowledge (Confidentialité totale)** : L'éditeur du logiciel (ARANEA SOFT) ne possède jamais vos clés ni vos passphrases. Si l'éditeur se fait pirater, vos données restent à l'abri car personne d'autre que vous ne possède la clé mathématique pour les ouvrir.
- **Air-Gapped (Isolation physique)** : C'est l'argument massue. Un système *Air-Gapped* fonctionne sur des ordinateurs **totalemt déconnectés d'Internet** (pas de Wi-Fi, pas de câble réseau, pas de Bluetooth). Une IA autonome ou un hacker à distance ne peut tout simplement pas attaquer une machine qui n'est pas sur le réseau.

Ces 2 fonctionnalités spécifiques changent radicalement la donne pour contrer les cybermenaces actuelles et futures :

4. Le transfert "Out-of-Band" par QR-Code géant (Matrice Optique Nayuki)

C'est l'une des idées les plus brillantes du logiciel pour éliminer le piratage réseau (colonne droite de l'infographie).

- **Le principe** : Plutôt que de faire transiter une clé de chiffrement ou un fichier critique par un câble réseau, une clé USB (qui peut contenir un virus) ou par Internet, l'ordinateur de départ génère un **QR-Code géant à haute densité** sur son écran. L'ordinateur de destination (complètement isolé) se contente de le "lire" visuellement via une caméra.
- **Pourquoi c'est imparable contre le HNDL** : Puisque les données passent par un flux optique direct (de l'écran à la caméra) et ne transitent jamais sur un câble ou des ondes Wi-Fi, il est **matériellement impossible** pour une IA ou un gouvernement d'intercepter le flux pour le stocker et tenter de le décrypter plus tard (*Harvest Now, Decrypt Later*).

5. Le RESET RAM et la "Zéroïsation" en site tactique à risque

Sur le terrain militaire ou dans les infrastructures critiques (OIV), le risque n'est pas seulement logiciel, il est aussi physique (ex: capture d'un ordinateur par l'ennemi).

- **Le problème classique** : Même lorsqu'on éteint un ordinateur, les clés de chiffrement restent lisibles dans la mémoire vive (RAM) pendant quelques secondes ou minutes.
- Des attaquants peuvent utiliser du gaz givrant pour figer la RAM et extraire les clés (attaque dite *Cold Boot*).
- **La parade de CryptoToolkit** : Le logiciel utilise un moteur codé en **Rust** ("Absolute memory safety") couplé à un bouclier anti-force brute (Bunker Argon2id).
- En cas d'alerte ou de procédure de déconnexion sur un site sensible, la fonction de **zéroïsation** réécrit instantanément des suites de 0 sur l'ensemble de la mémoire vive utilisée par le logiciel.
- Les clés de chiffrement et les passphrases disparaissent instantanément, ne laissant aucune trace physique à analyser.

Un positionnement "Net-Free" et "Souveraineté Totale" très attractif

La vraie force de ce produit est d'offrir une **immunité quantique complète** (via les algorithmes Kyber et Dilithium recommandés par le NIST) sans aucune dépendance extérieure. Dans un contexte européen où les entreprises doivent obéir à la directive de sécurité NIS 2, un outil qui promet

une licence à vie,

un fonctionnement 100 % local (sans abonnement Cloud, sans connexion Internet requise)

et qui est conforme aux normes d'exportation internationales

a un potentiel économique colossal auprès des banques, des armées et des industries de pointe ainsi que de nombreuses entreprises du secteur financier et commercial.

Ce logiciel intéresse des pans entiers de l'économie.

Oui, Les caractéristiques citées ciblent les secteurs les plus critiques de l'économie, soumis à des réglementations de sécurité drastiques (comme NIS 2 en Europe) :

- **Les Opérateurs d'Importance Vitale (OIV)** : Énergie (centrales nucléaires), gestion de l'eau, transports, qui utilisent déjà des réseaux isolés (*Air-Gapped*).
- **Le secteur bancaire et financier** : Pour protéger les transactions contre la future menace quantique.
- **La santé et les laboratoires de recherche** : Pour éviter le vol de brevets médicaux ou de données de patients à long terme.
- **La Défense et l'Espace** : Où le secret d'État exige du Zéro-Knowledge absolu.
- Toutes les entreprises et administrations sont plus ou moins concernées par les vols et fuites de données.

En conclusion

Sur le papier, si CryptoToolkit tient ses promesses d'intégration (notamment la facilité d'utilisation d'un système à la fois post-quantique et isolé), il s'agit d'un **outil de rupture très intéressant pour la cyberdéfense**.

Recommandations extérieures : Il faut surveiller sa sortie réelle pour s'assurer que ses lignes de code soient auditées par des organismes officiels (comme l'ANSSI en France) afin de valider cette excellente fiche technique.

FAQ — Questions / Réponses

1)• Voici un texte accessible sur Rust, pensé pour un décideur non technique :

- Rust est un langage de programmation récent, né pour corriger les faiblesses des langages plus anciens comme le C++, qui reste puissant mais réputé difficile à sécuriser complètement.
- La majorité des failles de sécurité informatique dans le monde proviennent d'un même type d'erreur : une mauvaise gestion de la mémoire par le programme.
En C++, ce risque persiste même avec des développeurs expérimentés, car rien n'empêche structurellement l'erreur.
Rust, lui, bloque ce type d'erreur automatiquement, avant même que le logiciel soit mis en service
— un peu comme un contrôle qualité intégré qui refuse de laisser sortir un produit défectueux de l'usine.
- Pour une entreprise, cela se traduit très concrètement : moins d'incidents de sécurité après livraison, moins de correctifs d'urgence coûteux, moins de risque d'image en cas de faille rendue publique, et une mise sur le marché plus rapide car les erreurs sont détectées tôt, pas après coup.
- Autre point clé : Rust n'a aucune faiblesse en termes de vitesse d'exécution — il reste aussi rapide que le C++. Le gain ne se fait donc sur aucun compromis de performance, uniquement sur la fiabilité.
- C'est pourquoi des géants comme Microsoft, Google ou Amazon migrent leurs systèmes les plus sensibles vers Rust, et pourquoi des agences de cybersécurité gouvernementales le recommandent désormais pour les logiciels critiques — un signal de confiance particulièrement pertinent pour un produit de cryptographie.

argumentaire en 10 lignes sur l'usage de Rust pour déployer des bibliothèques NIST comme liboqs dans un logiciel comme CryptoToolkit™ :

1. **Sécurité mémoire par construction** : sans garbage collector, Rust élimine les buffer overflows, use-after-free et data races — les classes de failles historiquement responsables des CVE critiques dans les implémentations cryptographiques en C++.
2. **Encapsulation sûre de liboqs** : liboqs reste écrit en C, mais les bindings Rust (crate oqs) offrent une interface FFI sûre qui isole les risques mémoire résiduels du cœur C++ tout en conservant les performances natives.
3. **Performances natives** : compilation LLVM et zero-cost abstractions placent Rust au niveau du C/C++, un critère décisif pour les opérations ML-KEM/ML-DSA à fort volume dans un outil entreprise.
4. **Déterminisme temps réel** : l'absence de runtime GC garantit un comportement prévisible, essentiel pour les traitements air-gapped et les contraintes embarquées.
5. **Écosystème RustCrypto** : des implémentations pures Rust, auditées, existent pour les primitives complémentaires (AES, SHA-3), réduisant la dépendance à des bibliothèques C tierces non maîtrisées.
6. **Concurrence sans risque** : le modèle ownership/borrowing empêche les races de données dans les traitements multi-thread, un point sensible pour un logiciel manipulant de gros volumes de fichiers.

7. **Portabilité multiplateforme** : cargo et les cibles LLVM simplifient la compilation croisée Windows/Linux/macOS sans réécriture, cohérent avec une interface disponible en 200 langues.
8. **Résistance aux attaques par canaux auxiliaires** : Rust facilite l'écriture de code en temps constant, réduisant l'exposition aux timing attacks sur des opérations comme les signatures Falcon ou SLH-DSA.
9. **Alignement institutionnel** : les recommandations d'agences comme la CISA en faveur des langages memory-safe pour les infrastructures critiques constituent un argument différenciant face à des concurrents en C/C++.

Facilitation des certifications : une base de code Rust, plus auditable et à surface de vulnérabilités réduite, simplifie les démarches vers des certifications type Critères Communs ou FIPS.

- 2) • *»Le passage à Kyber et Falcon en PQC implique des tailles de clés et de signatures beaucoup plus grandes que RSA ou l'ECC (Elliptic Curve Cryptography). Comment CryptoToolkit™ gère-t-il l'impact sur la bande passante et l'allocation mémoire, surtout en environnement isolé (offline/air-gapped) ?«*

« C'est précisément pour cela que CryptoToolkit™ a été conçu dès le départ en architecture "offline-first".

En éliminant totalement le backend et les flux réseau continus, nous supprimons le goulot d'étranglement de la bande passante réseau.

Le traitement se fait exclusivement en mémoire volatile locale. Pour l'allocation mémoire, notre implémentation logicielle optimise l'arbre de Merkle (notamment pour SPHINCS+) et utilise des structures de données compactes pour que l'empreinte RAM reste transparente pour l'utilisateur, même sur des machines durcies de dotation militaire. »

• (3) • Performances CPU / Consommation d'énergie) :

"Les algorithmes post-quantiques, en particulier Falcon pour la signature, demandent des calculs intensifs en virgule flottante qui peuvent saturer les processeurs non optimisés.

Comment assurez-vous la scalabilité sur des terminaux légers ou des PC industriels ?"

• « *C'est notre force : l'approche multi-algorithmes. CryptoToolkit™ ne force pas un "big bang" technologique. Nous proposons une hybridation. Pour les terminaux à faibles ressources,*

- *nous pouvons coupler l'AES-256 classique avec **NTRU** (qui repose sur des réseaux de vecteurs et consomme moins que Kyber en chiffrement) ou utiliser des variantes optimisées de **Kyber** sans calculs flottants lourds.*
- *L'utilisateur ou l'administrateur choisit le curseur entre sécurité maximale post-quantique et performance CPU selon le matériel cible. »*

• **(4) • Quels sont Les 5 Piliers de votre Argumentaire (Pourquoi CryptoToolkit™)**

1. Rust : Seule l'architecture langage RUST permet d'implémenter les bibliothèques standardisées par le NIST comme "LIBOQS" avec les algorithmes PQC (Kyber-KEM, NTRU, HQC, FALCON, DILITHIUM, SPHINCS+

2. Zéro Backend / Zéro Cloud : Aucune télémétrie, aucun serveur. Risque de fuite réseau égal à zéro.

3. Environnement Dégradé : Fonctionne à 100 % sur des machines isolées (air-gapped), en OPEX ou en crise systémique.

4. Code Auditable : Périmètre de confiance local strict (mémoire volatile), taillé pour les certifications (CSPN / Diffusion Restreinte).

5. Zéro Coût d'Infrastructure : Pas de serveurs à maintenir pour l'éditeur, rentabilité immédiate des licences.

(5) • (Migration et Interopérabilité) :

"Si vos clients utilisent CryptoToolkit™ de manière totalement isolée (sans serveur, sans backend), comment gérez-vous la distribution des clés publiques post-quantiques entre les utilisateurs ?

Comment évitez-vous l'enfer de la gestion des clés (Key Management System) en mode offline ?"

- « *CryptoToolkit™ est un outil de chiffrement et un banc d'essai autonome, pas un gestionnaire de réseau centralisé. Il n'impose aucun système de transport.*

- Les clés publiques et les fichiers chiffrés (portant les signatures Falcon ou Dilithium) sont exportables sous formats standardisés (fichiers sécurisés, conteneurs).
- Dans un cadre Défense ou Industrie critique, ils sont acheminés par les canaux de confiance déjà existants de l'organisation (clés USB sécurisées, liaisons de données tactiques, intranets locaux hermétiques).
- • Nous sécurisons la donnée à la source, nous ne gérons pas le câble. »

(6) • (Certification et Confiance) :

"Le NIST a standardisé Kyber et Falcon, mais l'ANSSI (France) recommande une approche hybride prudente et n'accorde pas encore de visas de sécurité complets aux solutions 100 % post-quantiques autonomes.

Où en êtes-vous de la certification ?"

- « Nous suivons scrupuleusement la doctrine de l'ANSSI. C'est pourquoi CryptoToolkit™ propose la coexistence active : nous ne remplaçons pas le classique par le quantique, nous les superposons (ex: AES-256 + Kyber).
- Si l'un des nouveaux algorithmes présente une vulnérabilité mathématique imprévue, la couche classique (RSA-4096 / AES) continue de protéger la donnée.
- • Le schéma d'architecture de CryptoToolkit™ (Zéro backend, code auditable, cloisonnement strict) a été conçu pour simplifier au maximum le processus d'audit en vue d'une certification CSPN ou d'un agrément Diffusion Restreinte. »

(7) • C'est quoi un Module de dérivation de clés utilisé dans CryptoToolkit™ pour la protection des passphrases et secrets institutionnels?

Argon2id est l'algorithme de dérivation de mot de passe le plus robuste actuellement disponible pour les environnements critiques. Lauréat du Password Hashing Competition (PHC), il combine une architecture mémoire-dure, une résistance aux attaques GPU massives, et une protection contre les attaques par canal auxiliaire.

CryptoToolkit™ utilise Argon2id pour sécuriser son coffre-fort de 50 mots de passe / passphrases, et mots de passe à 4 critères sur 128 caractères garantissant que même un attaquant disposant de moyens étatiques ne peut dériver un secret stocké.

2. Pourquoi Argon2id est indispensable pour les infrastructures financières

2.1. Architecture mémoire-dure

Argon2id impose une charge mémoire élevée à chaque tentative de dérivation. Contrairement à PBKDF2, bcrypt ou scrypt, Argon2id ne peut pas être parallélisé efficacement.

Conséquences :

- saturation immédiate des GPU adverses,
- impossibilité de lancer des attaques massives,
- coût exponentiel pour chaque tentative.

2.2. Résistance aux attaques GPU

Les GPU sont conçus pour exécuter des milliers de threads en parallèle.

Argon2id neutralise cette capacité en imposant :

- une consommation mémoire par thread,
- une dépendance séquentielle,
- une impossibilité de paralléliser les calculs.

Un attaquant doit donc :

- réduire drastiquement le nombre de threads,
- augmenter la mémoire par tentative,
- accepter des temps de calcul prohibitifs.

2.3. Protection contre les attaques étatiques

Les infrastructures financières sont ciblées par des acteurs disposant de :

- fermes de GPU,
- clusters HPC,
- ressources cloud massives.

Argon2id rend ces ressources inutilisables pour dériver un secret.

(8). Pourquoi Argon2id est critique dans un contexte post-quantique

Même si Argon2id n'est pas un algorithme PQC, il est **indispensable** dans un logiciel post-quantique pour protéger :

- les passphrases,
- les mots de passe,
- les clés internes,
- les secrets de signature,
- les archives sensibles.

Les ordinateurs quantiques accélèrent les attaques sur :

- RSA,
- ECC,
- Diffie-Hellman,

mais n'accélèrent pas Argon2id, car :

- Argon2id est mémoire-dure,
- Argon2id n'est pas parallélisable,
- Argon2id n'est pas optimisable par Grover.

(9). Un module PANIC BUTTON / RESET RAM / Zéroisation de la RAM/ dans un environnement tactique en cas d'intrusion Physique ou de la saisie de la machine en cours de traitement ?

Sécurité Anti-Intrusion (Zéroisation) : Intégration d'un Panic Button pour le RESET immédiat de la RAM et la mise à zéro des flux de données, en cas d'accès physique non autorisé.

CryptToolkit™ possède une fonction totalement exclusive qui peut s'avérer utile en environnement tactique hostile, critique, ou non protégé.

Un bouton de RESET/ Vidage de la RAM + Ajout de Zéro dans celle ci qui se place en haut de la pile des opérations, c'est immédiat en 1 secondes par l'opérateur.

L'opération est sécurisée par un bouton avec confirmation dans toutes les langues

En cas d'intrusion dans un site ou le logiciel est en fonctionnement ou en cas de saisie d'une machine sur un site tactique :

Étape 1 : Interruption des flux de données (Abort Streams)

Le thread Rust gérant l'écriture ou la lecture des fichiers via les buffers (Chunking) est immédiatement tué (abort ou via un CancellationToken).

Cela stoppe net tout traitement en cours pour éviter les fuites de fragments de fichiers.

Étape 2 : Purge de la mémoire vive (Zeroize RAM)

C'est l'étape la plus critique. En Rust, vider une variable (drop) supprime son pointeur, mais les données sensibles (comme votre Passphrase BIP-39 ou les clés privées RSA) restent physiquement écrites dans la RAM jusqu'à ce que le système d'exploitation les réalloue.

Un attaquant avec un accès physique à la machine pourrait extraire les clés.

- **La solution Rust :** On Utilise la crate officielle zeroize. Elle force l'écrasement de la mémoire contenant les structures de clés par des zéros binaires (0x00) juste avant de libérer la mémoire.

Étape 3 : Réinitialisation de l'interface (UI Lockdown)

Une fois la RAM lavée de toute trace cryptographique :

- Le vu-mètre repasse instantanément au **Rouge**  (**A zéro**).
- Toutes les fenêtres modales ouvertes (RSA, PQC, QR) sont fermées de force et vidées.
- L'interface se verrouille et redirige l'utilisateur sur l'écran d'accueil du générateur de passphrase, réinitialisant la session à son début.

comportement exact que le code doit exécuter lors du clic sur le bouton virtuel (représenté par le coup de poing Schneider) :

```
[ Clic Bouton Panique (Front-end Tauri) ]  
    |  
    ▼ (Appel de la commande Invoke Rust)  
[ 1. Interruption immédiate des Streams/Buffers ]  
    |  
    ▼  
[ 2. Écriture de zéros dans la mémoire vive (Zeroize RAM) ]  
    |  
    ▼  
[ 3. Purge du State Global & Verrouillage UI ]
```

•
•