Crypto-Agilité : Plan de Migration Post-Quantique DSI

Crypto-agilité, inventaire cryptographique, migration post-quantique : méthode en 4 phases pour les DSI. Outils, priorités et plan d'action opérationnel inclus.

La crypto-agilité est le concept central de toute stratégie de migration post-quantique réussie. Contrairement à une migration classique qui remplace un système par un autre, la crypto-agilité vise à rendre le système d'information capable de changer d'algorithme cryptographique sans refonte architecturale majeure — comme on changerait un moteur sans changer de voiture. Ce concept n'est pas nouveau : il est apparu dans les années 2000 lors de la transition de SHA-1 vers SHA-256, puis lors de la migration de TLS 1.0 vers TLS 1.2 et 1.3. Mais l'ampleur de la migration post-quantique, qui touche simultanément tous les usages de la cryptographie asymétrique dans le SI — TLS, SSH, PKI, VPN, signatures de code, messagerie chiffrée — donne à la crypto-agilité une importance stratégique sans précédent. Pour les DSI, la question n'est pas seulement de migrer vers les bons algorithmes, mais de construire un SI qui sera capable de migrer à nouveau dans 10 ou 20 ans, lorsque les prochaines évolutions cryptographiques se présenteront.

Comprendre la crypto-agilité : définition opérationnelle

Un système est crypto-agile lorsqu'il peut changer son algorithme cryptographique sans modification de son code applicatif, de son architecture ou de ses interfaces. En pratique, cela se traduit par plusieurs propriétés techniques :

Abstraction algorithmique : les algorithmes sont référencés par des identifiants standardisés (OID, suite names) dans la configuration, pas codés en dur dans le code source.

Négociation dynamique : les protocoles (TLS, IKEv2) négocient les algorithmes à chaque connexion, permettant un déploiement progressif sans coupure.

Séparation des préoccupations : la logique cryptographique est centralisée dans des bibliothèques ou services dédiés, pas dupliquée dans chaque application.

Paramétrage externe : les listes d'algorithmes autorisés sont gérées via des politiques externes (fichiers de configuration, HSM, service de gestion de clés), pas dans le code.

A contrario, un système non agile est un système qui importe directement `javax.crypto.Cipher.getInstance("RSA/ECB/PKCS1Padding")` dans son code Java, ou qui génère des clés RSA-2048 hardcodées dans ses scripts d'initialisation — migrer ce système vers ML-DSA nécessite de trouver et modifier chaque occurrence dans le code.

Étape 1 : L'inventaire cryptographique — cartographier l'existant

Vous ne pouvez pas migrer ce que vous ne connaissez pas. L'inventaire cryptographique est la première phase obligatoire, et souvent la plus longue — comptez entre 3 et 12 mois selon la taille et la complexité du SI.

> ⚡ **Retour terrain**
>
> Dans mes missions d'audit, je rencontre régulièrement la même configuration à risque : des règles de firewall héritées depuis 5 à 10 ans, que personne n'ose supprimer par crainte de casser quelque chose. J'ai développé une méthode de nettoyage progressive — analyser les logs de connexion sur 90 jours, identifier les règles sans trafic, les désactiver sans supprimer pendant 30 jours, puis valider avec les équipes métier. Sur un parc de 340 règles dans un groupe logistique, nous en avons supprimé 218 sans incident.

Que chercher dans l'inventaire ?

L'inventaire doit couvrir quatre catégories :

1. Les certificats et clés : tous les certificats X.509 déployés dans le SI — certificats de serveur TLS, certificats client, certificats CA intermédiaires et racine, certificats de signature de code, certificats S/MIME. Pour chaque certificat : algorithme de la clé (RSA-2048, ECDSA P-256...), algorithme de signature (sha256WithRSAEncryption...), date d'expiration, usage (TLS serveur, authentification, signature...), localisation (serveur, HSM, appliance, keystore).

2. Les configurations protocolaires : suites cipher TLS configurées sur chaque serveur web, reverse proxy, équipement réseau. Algorithmes d'échange de clés configurés dans les VPN IPsec et SSL. Algorithmes SSH autorisés sur les serveurs et équipements. Paramètres Kerberos dans l'Active Directory.

3. Les bibliothèques cryptographiques applicatives : les applications qui embarquent leurs propres bibliothèques crypto (JAR Java, DLL .NET, .so Linux), les versions d'OpenSSL utilisées, les modules Python (cryptography, pycryptodome), les modules Node.js. La dépendance est particulièrement critique pour les applications qui n'utilisent pas la bibliothèque système mais une version embarquée.

4. Les secrets à longue durée de vie : clés maîtresses dans les HSM, master secrets dans les systèmes de gestion de mots de passe, clés de chiffrement de bases de données (TDE), clés de chiffrement de sauvegardes. Ces secrets ont souvent des durées de vie de plusieurs années et sont les plus exposés au risque HNDL.

Outils d'inventaire automatisé

Plusieurs outils permettent d'automatiser partiellement l'inventaire :

OpenSSL en ligne de commande : `openssl s_client` et `openssl x509` permettent d'inspecter manuellement les certificats et suites cipher, mais ne s'appliquent pas à grande échelle sans scripting.

Nmap avec scripts NSE : les scripts `ssl-enum-ciphers` et `tls-alpn` permettent de scanner les suites cipher sur l'ensemble d'un réseau. Combiné à un script d'analyse automatisée, c'est un outil puissant pour l'inventaire réseau.

Keyfactor Command : plateforme commerciale de Certificate Lifecycle Management qui découvre automatiquement les certificats sur le réseau, les classe par algorithme et suit les expirations. C'est la référence pour les grandes PKI d'entreprise.

Venafi TLS Protect : concurrent de Keyfactor, avec des capacités similaires de découverte et de gestion du cycle de vie des certificats. Intègre des modules d'évaluation de la maturité post-quantique.

CryptoScan (ANSSI) : outil développé par l'ANSSI pour analyser la surface cryptographique d'un SI — moins complet que les solutions commerciales, mais gratuit et reconnu dans le contexte français.

Complétez l'inventaire automatisé par un questionnaire adressé aux équipes de développement applicatif — les outils de scan réseau ne trouvent pas les bibliothèques embarquées dans les

applications ou les clés stockées dans des keystores applicatifs. L'inventaire des actifs ISO 27001 constitue la base de données à enrichir avec les métadonnées cryptographiques.

À RETENIR

Points clés à retenir

La crypto-agilité est la capacité de changer d'algorithme sans refonte architecturale — construire cette capacité est aussi important que migrer vers PQC.

L'inventaire cryptographique couvre : certificats, configurations protocolaires, bibliothèques applicatives, secrets à longue durée de vie.

Les outils Keyfactor, Venafi et OpenSSL/Nmap scripting permettent d'automatiser 60-70% de l'inventaire.

L'hybridation (classique + PQC) est la stratégie de transition recommandée par l'ANSSI et le NIST.

Priorisez par risque HNDL : les flux contenant des données de durée de vie >5 ans migrent en premier.

Étape 2 : Analyse et priorisation par risque

L'inventaire produit une liste — souvent longue — de systèmes à migrer. Il faut les prioriser par niveau de risque pour allouer les ressources efficacement.

Matrice de priorisation post-quantique

Système	Risque HNDL	Complexité migration	Priorité
VPN accès sensibles (R&D, direction)	Très élevé	Moyenne	P1 — 2026
TLS serveurs exposés données sensibles	Élevé	Faible	P1 — 2026
PKI interne (CA racine + intermédiaires)	Élevé	Très élevée	P1 — 2027
SSH accès serveurs production	Moyen	Faible	P2 — 2027
S/MIME messagerie direction	Élevé	Moyenne	P2 — 2027
Signature de code/logiciels	Moyen	Moyenne	P2 — 2028
TLS serveurs internes non sensibles	Faible	Faible	P3 — 2028-2029
Applications legacy (COBOL, AS/400)	Variable	Très élevée	P3 — 2029-2030
OT/SCADA/IoT	Variable	Extrême	P3 — 2030+

Le risque HNDL est évalué selon deux critères : la sensibilité des données transportées ou protégées, et la durée de vie utile de ces données. Un flux VPN transportant des plans de R&D pharmceutique a un risque HNDL maximal. Un flux TLS vers un site web marketing a un risque minimal.

Identifier les quick wins

Certaines migrations sont simples et offrent un gain de sécurité immédiat :

TLS hybride sur serveurs web : une simple mise à jour de configuration OpenSSL ou Nginx avec le provider OQS suffit pour ajouter ML-KEM aux suites cipher. Délai : quelques jours, risque opérationnel : quasi nul (compatibilité ascendante garantie).

Mise à jour des suites SSH : OpenSSH 9.0+ supporte les algorithmes post-quantiques hybrides. Une mise à jour des serveurs et une modification du fichier `sshd_config` suffisent.

Rotation des clés RSA courtes : remplacer toute clé RSA inférieure à 4096 bits par une clé plus longue ou une clé ECDSA P-384 offre un gain immédiat à faible coût, en attendant la migration PQC complète.

Étape 3 : Stratégie d'hybridation — la transition en douceur

La migration post-quantique ne peut pas être un big bang. Les systèmes migrant vers PQC doivent rester interopérables avec les systèmes non encore migrés, parfois pendant des années. L'hybridation est la solution.

Principes de l'hybridation

Un algorithme hybride combine un algorithme classique (ECDH-P256) avec un algorithme post-quantique (ML-KEM-768) de façon à ce que la sécurité de l'ensemble soit garantie si *au moins un* des deux algorithmes reste sûr. Si ML-KEM s'avérait avoir une faiblesse non découverte (improbable mais théoriquement possible), le classique ECDH maintiendrait la sécurité. Inversement, si ECDH est cassé par un CRQC, ML-KEM maintient la protection.

Cette approche est recommandée par l'ANSSI et le NIST pour la période de transition. Elle est déjà implémentée dans Chrome (X25519Kyber768), dans les serveurs Cloudflare et AWS, et dans OpenSSL 3.x avec le provider OQS.

Hybridation dans TLS 1.3

TLS 1.3 supporte nativement la négociation de groupes d'échange de clés hybrides via l'extension `supported_groups`. Configurer un serveur pour proposer `X25519Kyber768Draft00` (ou le nom standardisé final) en premier dans la liste des groupes supportés suffit pour activer l'hybridation. Les clients non compatibles (navigateurs anciens, systèmes embarqués) continueront à négocier X25519 classique — la compatibilité est maintenue.

Hybridation dans les certificats (X.509)

Les certificats hybrides combinent une clé publique classique (RSA ou ECDSA) et une clé publique post-quantique (ML-DSA) dans un seul certificat. Le standard en cours de finalisation à l'IETF (draft-ietf-lamps-pq-composite-sigs) définit le format. DigiCert, Entrust et Keyfactor proposent déjà des implémentations expérimentales. Ces certificats peuvent être vérifiés par des

clients classiques (qui ignorent la signature PQC) et offrent une protection PQC pour les clients compatibles.

Étape 4 : Migration opérationnelle et retrait des algorithmes classiques

La dernière phase est le retrait progressif des algorithmes classiques, une fois l'hybridation déployée et les systèmes non compatibles PQC identifiés et migrés ou mis hors ligne.

Procédure de retrait TLS

Le retrait des suites cipher classiques dans TLS doit suivre la même méthodologie que le retrait de TLS 1.0/1.1 : d'abord désactiver sur les serveurs de test, monitorer les erreurs de connexion, identifier les clients incompatibles, les contacter pour mise à jour, puis désactiver en production. Un plan de changement structuré, aligné sur la procédure de gestion des changements ISO 27001, est indispensable.

Gestion des systèmes legacy

Les systèmes anciens qui ne peuvent pas être mis à jour (applications COBOL, systèmes embarqués, équipements industriels) constituent le cas le plus difficile. Les options sont :

Gateway PQC : un proxy ou une appliance qui termine les connexions PQC côté réseau moderne et présente une interface classique au système legacy — les solutions Thales et Fortanix offrent cette architecture.

Isolation réseau : segmenter le système legacy dans un réseau isolé, sans connexion vers l'extérieur, minimisant l'exposition aux attaques HNDL.

Fin de vie accélérée : la menace PQC peut justifier d'accélérer le remplacement d'un système legacy qui était prévu dans 10 ans — le coût de remplacement doit être comparé au coût du risque.

Intégration dans les processus SSI existants

La migration post-quantique ne doit pas être un projet parallèle déconnecté de la gouvernance SSI existante. Elle s'intègre dans les processus ISO 27001 en place :

Politique de cryptographie (A.8.24) : mise à jour pour inclure les algorithmes PQC approuvés, les critères d'hybridation, et les algorithmes en cours de retrait. La politique de cryptographie conforme à l'ISO 27001 doit être le document de référence.

Gestion des risques : le risque HNDL doit apparaître dans le registre des risques avec sa probabilité, son impact et ses mesures de traitement.

Gestion des actifs : l'inventaire cryptographique s'intègre à l'inventaire des actifs existant.

Gestion des changements : chaque migration d'algorithme est un changement à gérer selon le processus existant.

Audits internes : les audits SSI doivent inclure des questions sur l'avancement de la migration post-quantique et la conformité de la politique cryptographique.

La gap analysis ISO 27001 2022 peut être enrichie d'une dimension post-quantique, évaluant la maturité crypto-agile de chaque domaine de contrôle concerné.

Cas pratique : migration TLS post-quantique avec OpenSSL

Pour illustrer concrètement la démarche, voici les étapes d'une migration TLS hybride sur un serveur Nginx utilisant OpenSSL 3.x :

Prérequis : OpenSSL 3.x compilé avec le provider OQS (Open Quantum Safe), disponible sur Ubuntu 24.04 LTS via les paquets `oqs-provider`.

Configuration Nginx : dans le bloc `ssl_conf_command`, ajouter les groupes hybrides : `ssl_conf_command Groups X25519Kyber768Draft00:prime256v1:secp384r1;`. Cette configuration propose ML-KEM hybride en priorité, avec fallback sur les groupes classiques pour les clients non compatibles.

Vérification : `openssl s_client -connect monserveur.fr:443 -curves X25519Kyber768Draft00` permet de vérifier que la négociation hybride fonctionne. Côté monitoring, les métriques TLS des équipements réseau montreront progressivement la proportion de connexions utilisant les nouveaux groupes.

Budget et ressources humaines

La crypto-agilité a un coût humain souvent sous-estimé. La migration post-quantique nécessite des compétences rares : cryptographes applicatifs capables d'analyser le code source pour les usages crypto hardcodés, ingénieurs PKI maîtrisant les nouvelles extensions X.509, architectes réseau connaissant les nouvelles suites cipher des équipements. Former ces compétences en interne ou les recruter doit faire partie du plan de programme.

Pour une organisation de 5000 employés avec un SI de taille moyenne, le budget programme sur 5 ans inclut typiquement : 1 à 2 ETP dédiés à la coordination du programme, les coûts de licences Keyfactor ou Venafi (100k€ à 300k€/an selon le volume de certificats), les coûts de remplacement des HSM non compatibles PQC (50k€ à 500k€ selon le nombre et le modèle), les coûts de consulting pour les phases d'inventaire et de migration PKI, et les coûts de formation.

Pour approfondir, consultez les recommandations officielles : NIST Post-Quantum Cryptography et le guide de l'ANSSI sur les mécanismes cryptographiques.

Cas d'usage concrets : crypto-agilité par secteur

La crypto-agilité ne s'applique pas de la même façon selon le secteur d'activité. Les contraintes réglementaires, les systèmes en place et les données à protéger varient considérablement, ce qui détermine la priorité et la nature des premières migrations à entreprendre. Voici comment les principaux secteurs abordent concrètement le défi.

Finance (SWIFT, SEPA, paiements). Le secteur bancaire est particulièrement exposé au risque HNDL : les données de transactions enregistrées aujourd'hui pourraient être déchiffrées dans 5 à 10 ans, révélant des flux financiers sensibles. SWIFT travaille depuis 2023 sur une roadmap PQC pour ses messageries interbancaires. En France, les DSI des banques et des établissements de paiement doivent anticiper les exigences DORA (Digital Operational Resilience Act) qui imposent des tests de résilience incluant les scénarios de rupture cryptographique. La priorité est la migration des canaux de communication interbancaires (HSM de paiement, flux SEPA), puis la PKI interne, puis les API open banking.

Santé (DMP, RPPS, données patients). Le Dossier Médical Partagé et le Répertoire Partagé des Professionnels de Santé contiennent des données à durée de vie extrêmement longue — un dossier médical peut rester sensible 50 ans. C'est l'un des cas HNDL les plus critiques : des données chiffrées aujourd'hui avec RSA-2048 et capturées par un adversaire pourront potentiellement être déchiffrées avant la fin de vie du patient. L'ANS (Agence du Numérique en Santé) doit inclure des exigences PQC dans les futures versions des référentiels HDS et dans les spécifications des éditeurs de logiciels de santé.

Industrie (SCADA, OT, systèmes embarqués). Le secteur industriel représente le cas le plus difficile techniquement : les automates SCADA ont des cycles de vie de 15 à 25 ans, fonctionnent avec des systèmes d'exploitation obsolètes, et n'ont souvent pas été conçus avec des capacités de mise à jour cryptographique. La crypto-agilité est quasi inexistante dans ces environnements. La stratégie recommandée est le déploiement de gateways PQC qui protègent les flux SCADA en terminant les connexions PQC côté réseau IT, sans toucher aux systèmes OT. La segmentation réseau et la surveillance des flux constituent des mesures compensatoires en attendant le renouvellement naturel des équipements.

Secteur	Défi principal	Priorité migration
Finance	Messageries SWIFT/SEPA, HSM de paiement, conformité DORA	Très haute — données HNDL à risque immédiat
Santé	Durée de vie des données patients, référentiels HDS, DMP/RPPS	Très haute — 50 ans de sensibilité résiduelle
Industrie / OT	Cycles de vie 15–25 ans, pas de mise à jour cryptographique native	Moyenne — gateways PQC en compensation
Défense / OIV	Exigences ANSSI, qualification CSPN, systèmes classifiés	Critique — migration imposée par les autorités

Quel que soit le secteur, la crypto-agilité est la propriété architecturale qui détermine la vitesse et le coût de la migration. Les organisations qui ont investi dans des couches d'abstraction cryptographique — via des HSM modernes, des PKI managées, des bibliothèques système unifiées — migrent 3 à 5 fois plus vite que celles dont les applications embarquent leur propre crypto. Pour comprendre les fondements des algorithmes concernés, notre guide sur les standards NIST post-quantiques FIPS 203/204/205 détaille ML-KEM, ML-DSA et SLH-DSA.

Mesurer la maturité crypto-agile de votre organisation

Piloter un programme de crypto-agilité sans indicateurs objectifs revient à naviguer sans boussole. Les cinq KPI suivants permettent de mesurer la maturité de votre organisation et d'en rendre compte au COMEX de façon concrète. Ils sont conçus pour être mesurables dès la phase d'inventaire et suivis trimestriellement tout au long du programme de migration.

% de systèmes cryptographiques inventoriés. Mesure la complétude de la cartographie. Cible : 100% des systèmes classifiés P1 et P2 (selon la matrice de priorisation), 80% du SI global. Un taux inférieur à 60% indique un inventaire insuffisant pour piloter la migration. Outil de mesure : exports Keyfactor/Venafi + questionnaire équipes applicatives.

Délai moyen de rotation des clés (DMRK). Mesure la réactivité opérationnelle en cas de compromission ou d'obsolescence algorithmique. Cible : moins de 72 heures pour les clés critiques (CA intermédiaires, VPN, TLS exposés), moins de 2 semaines pour les clés standard. Un DMRK supérieur à 30 jours sur des clés critiques est un signal d'alerte direct sur la maturité opérationnelle.

% de protocoles en mode hybride (classique + PQC). Mesure l'avancement de la migration pour la phase de transition. Cible 2026 : 30% des flux TLS exposés en hybride. Cible 2028 : 80%. Ce KPI suit directement l'exécution du plan de migration et identifie les retards par domaine (TLS web, VPN, SSH, PKI).

Couverture des tests adversariaux (pentest crypto). Mesure la robustesse des implémentations en production. Cible : au moins un pentest annuel ciblant les configurations cryptographiques, avec revue des suites cipher TLS, des paramètres SSH et des algorithmes de signature. La

menace quantique sur RSA et AES illustre pourquoi les configurations cryptographiques doivent être testées de façon adversariale et pas seulement auditées sur documentation.

Temps de migration estimé résiduel (TMER). Mesure la distance restante jusqu'à la conformité cible. Calculé en agrégeant les estimations de migration par système P1/P2/P3, pondérées par la complexité. Ce KPI permet au COMEX de visualiser la trajectoire du programme et d'arbitrer les ressources en cas de dérive. Mise à jour recommandée : semestrielle, avec révision annuelle des hypothèses de complexité.

FAQ

Qu'est-ce qu'un système crypto-agile concrètement ?

Un système crypto-agile est un système dont on peut changer les algorithmes cryptographiques via une modification de configuration, sans toucher au code source. Par exemple : un serveur web dont les suites TLS sont définies dans un fichier de configuration Nginx, qu'on peut modifier pour ajouter ML-KEM en quelques minutes. À l'opposé, une application Java qui appelle directement `new RSAKeyPairGenerator()` dans son code n'est pas crypto-agile — il faudra modifier, recompiler et redéployer le code pour migrer.

Doit-on remplacer tous les certificats RSA immédiatement ?

Non, l'approche progressive est recommandée. Commencez par les certificats à longue durée de vie (CA racine, intermédiaires) et ceux protégeant des données à forte sensibilité temporelle. Les certificats TLS de serveur web, qui ont généralement une durée de vie de 90 jours à 1 an, peuvent suivre lors de leur prochain renouvellement naturel. Profitez de chaque renouvellement de certificat pour adopter ML-DSA ou un certificat hybride.

Comment gérer les fournisseurs qui ne supportent pas encore PQC ?

Intégrez des clauses de crypto-agilité dans vos contrats fournisseurs : exigence de support des standards NIST PQC dans un délai défini (typiquement 2 à 3 ans), fourniture d'une roadmap PQC documentée, droit d'audit sur la maturité post-quantique. Pour les fournisseurs critiques qui ne peuvent pas s'engager, évaluez les options d'isolation (gateway PQC) ou de remplacement.

OpenSSL supporte-t-il déjà les algorithmes post-quantiques ?

OpenSSL 3.x en combinaison avec le provider OQS (Open Quantum Safe) supporte ML-KEM, ML-DSA et SLH-DSA pour les échanges de clés et signatures TLS. Le support natif dans OpenSSL sans provider externe est prévu dans les versions futures (OpenSSL 3.4+). Pour les environnements de production, le provider OQS est suffisamment mature pour des déploiements pilotes, mais demande une validation soigneuse pour les environnements critiques.
