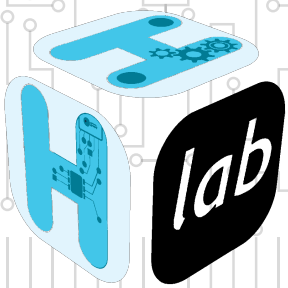
H2LAB-STIG-SECCOMP
08-2026

Systèmes embarqués : méthodologie de choix des composants de sécurité



STIG H²Lab

Informations

Créée en 2020, H2Lab est une association loi 1901 dont l'objectif est de concevoir, développer et diffuser des solutions open-hardware et open-source dédiées à l'expérimentation autour des systèmes embarqués.
Ses principaux axes de travail sont :

- Conception de plateformes matérielles pour l'analyse hardware et software
- Développement d'alternatives ouvertes aux outils propriétaires, souvent opaques
- Publication de guides techniques et de recommandations pour promouvoir les bonnes pratiques en matière de sécurité, de confidentialité et de durabilité dans l'écosystème des objets connectés et des systèmes embarqués

Soutien aux chercheurs, enseignants et étudiants via la mise à disposition d'outils, de contenus pédagogiques et de formations.
Partenariats académiques pour encourager la mutualisation des ressources et des savoirs.

Historique des révisions

Version	Date	Auteur	Modifications
1.0	Août 2026	H2Lab	Version initiale du guide

Table des matières

1
Glossaire et acronymes

1.1 Glossaire

Surface d'attaque. Ensemble des interfaces physiques, logiques et organisationnelles exploitables par un attaquant (ports de débogage, bus internes, mise à jour, cloud, maintenance), notion formalisée dans les approches de modélisation de menace et de gestion du risque [5, 60, 53].

RoT (Root of Trust). Ensemble minimal de composants de confiance (matériels et logiciels) dont dépend toute la chaîne de vérification. Les cadres NIST de protection de firmware et d'intégrité de plateforme en font un prérequis de base [47, 43].

SE (Secure Element). Composant sécurisé, résistant aux attaques physiques, pour stocker des secrets et exécuter des opérations cryptographiques sous politique de sécurité stricte [25, 59, 17].

TPM (Trusted Platform Module). Module normalisé pour l'identité machine, la mesure d'intégrité (PCR), le scellement et l'attestation distante [30, 66, 43].

TEE (Trusted Execution Environment). Environnement d'exécution isolé au sein du SoC, séparant monde sécurisé et monde normal, avec surfaces d'attaque principalement micro-architecturales et logicielles [8, 35, 29].

Attestation. Mécanisme de preuve cryptographique de l'état d'un composant ou d'une plateforme vers un vérificateur [63, 43].

Agilité cryptographique. Capacité à remplacer algorithmes, tailles de clefs et profils de sécurité sans re-architecture du produit, point de contrôle pour la transition post-quantique [57, 6, 22].

Chaîne de confiance (Chain of Trust). Enchaînement de vérifications d'intégrité et d'authenticité depuis la racine de confiance jusqu'aux composants logiciels applicatifs.

Secure Boot. Vérification cryptographique de chaque étape de démarrage afin d'empêcher l'exécution de code non autorisé.

Anti-rollback. Mécanisme empêchant l'installation d'une version antérieure vulnérable d'un firmware ou d'un composant logiciel.

Provisioning. Ensemble des opérations d'initialisation sécurité en fabrication et en déploiement (identité, clefs, certificats, politiques).

Scellement (Sealing). Liaison cryptographique d'un secret à un état de plateforme mesuré (ex : valeurs PCR), pour empêcher sa réutilisation hors contexte attendu.

Canal auxiliaire (Side-channel). Fuite d'information via des grandeurs physiques (consommation, temps, rayonnement, faute) plutôt que via l'interface logique prévue.

Injection de faute. Perturbation volontaire (glitch, tension, horloge, EMI) visant à provoquer un comportement fautif exploitable.

Attestation distante. Vérification à distance de l'état d'un dispositif par un vérificateur externe, sur la base de preuves signées et de mesures.

HNDL (Harvest Now, Decrypt Later). Stratégie d'attaque consistant à collecter aujourd'hui des données chiffrées pour les déchiffrer plus tard avec des moyens cryptanalytiques supérieurs (notamment quantiques).

Hybridation cryptographique. Combinaison d'un mécanisme classique et d'un mécanisme post-quantique (signature ou établissement de clef) afin de conserver la sécurité si l'un des deux schémas est affaibli.

TCB (Trusted Computing Base). Ensemble matériel et logiciel qui doit rester digne de confiance pour garantir la sécurité du système.

TOCTOU (Time Of Check To Time Of Use). Classe de vulnérabilités où l'état vérifié diffère de l'état réellement utilisé en raison d'un changement entre contrôle et exécution.

Cryptoagilité post-quantique. Capacité à introduire, remplacer ou retirer des primitives PQC sans redévelopper entièrement la plateforme.

1.2 Liste des acronymes

ANSSI. Agence nationale de la sécurité des systèmes d'information

APDU. Application Protocol Data Unit (format de commande/réponse des cartes à puce)

API. Application Programming Interface

BIOS. Basic Input/Output System

CI/CD. Continuous Integration / Continuous Delivery

CPA. Correlation Power Analysis (attaque par canal auxiliaire)

CRA. Cyber Resilience Act (Règlement UE 2024/2847) [24]

CRQC. Cryptographically Relevant Quantum Computer

DFA. Differential Fault Analysis

DFU. Device Firmware Update

DH. Diffie-Hellman

DPA. Differential Power Analysis (attaque par canal auxiliaire) [34]

DRBG. Deterministic Random Bit Generator

EAL. Evaluation Assurance Level (Common Criteria) [17]

ECC. Elliptic Curve Cryptography

ECDH. Elliptic Curve Diffie-Hellman

ECDSA. Elliptic Curve Digital Signature Algorithm

ENISA. European Union Agency for Cybersecurity

FIPS. Federal Information Processing Standards (ex : FIPS 140-3, FIPS 203, FIPS 204) [49, 54, 55]

GPP. General Purpose Processor

HSE. Hardware Security Engine

I2C. Inter-Integrated Circuit

IEC. International Electrotechnical Commission

ISO. International Organization for Standardization

JTAG. Joint Test Action Group (interface de débogage/test)

KDF. Key Derivation Function

MAC. Message Authentication Code

MCU. Microcontroller Unit

MITM. Man-In-The-Middle

ML-DSA. Module-Lattice Digital Signature Algorithm (standard FIPS 204)

ML-KEM. Module-Lattice Key-Encapsulation Mechanism (standard FIPS 203)

MPU. Memory Protection Unit

NIST. National Institute of Standards and Technology

OTA. Over-The-Air (mise à jour distante)

PCR. Platform Configuration Registers (TPM) [30]

PQC. Post-Quantum Cryptography

PSA. Platform Security Architecture

PUF. Physical Unclonable Function [38]

RBAC. Role-Based Access Control

RNG. Random Number Generator

RSA. Rivest-Shamir-Adleman

RSSI. Responsable de la sécurité des systèmes d'information

SBOM. Software Bill of Materials

SCP. Secure Channel Protocol GlobalPlatform [25]

SCP03. Version SCP03 du protocole de canal sécurisé GlobalPlatform

SLH-DSA. Stateless Hash-Based Digital Signature Algorithm (standard FIPS 205)

SoC. System on Chip

SPI. Serial Peripheral Interface

SSI. Sécurité des systèmes d'information

SWP. Single Wire Protocol

TEE. Trusted Execution Environment

TLS. Transport Layer Security

TOCTOU. Time Of Check To Time Of Use

TPM. Trusted Platform Module

TRNG. True Random Number Generator

UART. Universal Asynchronous Receiver-Transmitter

UEFI. Unified Extensible Firmware Interface

2
Introduction et périmètre

2.1 Objectif du guide

Ce guide fournit une méthode complète pour sélectionner, intégrer et auditer des briques de sécurité matérielle (SE, TPM, TEE, HSE) dans des produits embarqués. L'objectif est double : réduire le risque technique (clonage, extraction de clefs, compromission du boot, fraude applicative) et démontrer la conformité aux référentiels de sécurité applicables [5, 53, 31, 24].

Le document est volontairement opérationnel : chaque chapitre relie les choix techniques à des menaces observées dans l'état de l'art (canaux auxiliaires, injections de fautes, attaques micro-architecturales, attaques supply chain), puis mappe les contrôles aux standards et guides reconnus [34, 13, 33, 36, 52, 4].

2.2 Public visé

Le guide s'adresse :

- aux architectes sécurité et architectes produits
- aux équipes hardware/firmware/logiciel embarqué
- aux responsables SSI, RSSI et équipes conformité
- aux auditeurs techniques et évaluateurs de produits connectés

Il est applicable aussi bien aux produits grand public qu'aux environnements industriels et critiques, avec adaptation du niveau d'assurance (ex : EAL, FIPS, IEC 62443) [17, 49, 27].

2.3 Périmètre technique et hypothèses

2.3.1 Type d'équipement et contexte

Le périmètre couvre les systèmes embarqués contraints (MCU, microprocesseurs applicatifs, SoC hétérogènes), objets connectés, passerelles edge et équipements industriels ayant des besoins de confidentialité, intégrité, disponibilité et traçabilité. Les profils visés incluent les architectures à faible consommation et à longue durée de vie, exposées au risque *Harvest Now, Decrypt Later* pour les données durables [40, 6].

2.3.2 Cycle de vie considéré

Le guide traite le cycle de vie de bout en bout :

- conception (menaces, exigences, architecture)
- industrialisation (provisioning, personnalisation, tests)
- déploiement (identité, enrôlement, mise en service)
- exploitation (supervision, rotation de clefs, réponse à incident)
- maintenance (patch, mise à jour sécurisée, decommissioning)

Cette approche est alignée avec les exigences de sécurité du cycle de vie dans les référentiels NIST et européens [52, 24, 31].

2.3.3 Hypothèses de menace

Les hypothèses de menace incluent :

- attaquant opportuniste avec accès local au produit
- attaquant expérimenté disposant de moyens de laboratoire
- attaquant supply chain (firmware modifié, composant clone, CI/CD compromis)
- attaquant distant exploitant des vulnérabilités logicielles ou protocolaires

Les familles d'attaques prises en compte sont détaillées à partir de la littérature : DPA/CPA [34, 14], injections de fautes [13, 12], attaques par exécution spéculative [33, 36], attaques TEE/SGX [15, 69], rejeu et interception sur bus [25, 47].

2.3.4 Lecture opérationnelle

Le guide peut être lu selon trois axes :

- **Architecture :** principes de confiance et choix technologiques
- **Intégration :** schémas d'implémentation, interfaces, politiques de clefs
- **Audit/conformité :** preuves attendues, critères d'acceptation, mappings normatifs

2.4 Modèle de menace, profils d'attaquants et objectifs de sécurité

Le guide s'appuie sur un modèle de menace explicite afin d'éviter une lecture purement technique des mécanismes de sécurité matérielle. Les menaces sont considérées comme des scénarios d'attaque en chaîne, combinant accès physique, compromission logicielle, mauvaise gestion des secrets et défaut de traçabilité.

2.4.1 Menaces principales

Les menaces prises en compte sont les suivantes :

- **Usurpation d'identité de plateforme :** clonage, substitution de cartes ou de composants, contamination de la chaîne de personnalisation
- **Compromission du boot et du firmware :** injection de code non autorisé, rollback, modification de la chaîne de confiance

- **Extraction de secrets et d'éléments sensibles :** clefs racines, secrets applicatifs, données de provisioning, certificats
- **Altération d'opérations critiques :** falsification de commandes, rejeu, usurpation de transaction, perte d'intégrité
- **Abus de maintenance et de débogage :** ouverture de mode service, accès non tracé à des interfaces de test, exfiltration via canaux auxiliaires

2.4.2 Profils d'attaquants

Le guide distingue formellement plusieurs profils d'attaquants :

- **AM-01 – Attaquant local opportuniste :** accès physique limité, outils standards, objectif principal d'extraction de secrets ou de contournement simple
- **AM-02 – Mainteneur malveillant ou sous-traitant compromis :** accès à des opérations de maintenance, de provisionnement ou de diagnostic
- **AM-03 – Attaquant expérimenté en laboratoire :** moyens matériels et logiciels avancés, capable d'exploiter des canaux auxiliaires et des fautes physiques
- **AM-04 – Attaquant hybride supply chain/terrain :** combinaison de compromission de la chaîne de production et d'intervention sur le terrain pour obtenir une persistance durable

2.4.3 Objectifs de sécurité

Les choix techniques doivent être évalués par rapport à des objectifs de sécurité explicites :

- **OS-01 – Confidentialité des actifs critiques :** empêcher l'extraction exploitable des secrets, des clefs et des images sensibles
- **OS-02 – Intégrité du système :** garantir qu'aucune modification non autorisée ne puisse altérer l'exécution ou la chaîne de confiance
- **OS-03 – Contrôle d'accès aux fonctions sensibles :** limiter les opérations de boot, de maintenance, de provisioning et d'attestation à des entités autorisées
- **OS-04 – Résistance aux attaques physiques de base :** réduire la faisabilité des attaques par faute, canaux auxiliaires ou manipulation directe
- **OS-05 – Traçabilité et audit :** produire des preuves exploitables pour la supervision, l'investigation et la conformité

2.5 Rappel de l'état de l'art des technologies de sécurité matérielle

2.5.1 Les Secure Elements (SE)

Les SE sont conçus pour la protection des secrets : anti-tamper, stockage cloisonné, contrôle d'usage des clefs et opérations cryptographiques limitées par politique [25, 59, 17]. Ils sont adaptés aux usages de long terme (identité, paiement, signature, credentials IoT) lorsque l'extraction physique est une menace crédible.

Les variantes de SE (Java Card applet, firmware propriétaire, open hardware) différencient principalement le modèle de programmation, la portabilité et la gouvernance de la sécurité. Les solutions ouvertes, par exemple les architectures basées sur OpenTitan ou des approches industrielles comme Tropic Square, augmentent l'auditabilité et la traçabilité du design, mais imposent un pilotage rigoureux de la chaîne de build et de vérification formelle [37, 65, 52].

2.5.2 Les Hardware Security Engines (HSE)

Les HSE sont des blocs sécurisés intégrés aux SoC qui offrent des primitives cryptographiques, du stockage de clefs et des services d'amorçage sécurisé à faible latence. Leur caractéristique principale est l'intégration au SoC ; leur limite est la dépendance au domaine de confiance du SoC, avec exposition directe aux attaques système selon l'architecture retenue [11, 8, 47].

2.5.3 Les Trusted Platform Modules (TPM)

Le TPM 2.0 fournit une ancre de confiance standardisée pour mesurer l'état du boot, sceller des secrets à un état de plateforme et produire des preuves d'attestation vérifiables [30, 66, 43]. Il est adapté à la conformité à distance et à la gouvernance de flotte, du fait de ses primitives de mesure (PCR), de politique et de quote [30, 66].

Sa limite principale en embarqué contraint est le coût d'intégration logicielle (pile de commandes, gestion des politiques, orchestration serveur d'attestation), ainsi que la latence de certaines opérations comparativement à des chemins purement locaux [66, 51].

2.5.4 Les Trusted Execution Environments (TEE)

Un TEE isole des traitements sensibles dans le processeur généraliste (ex : TrustZone), ce qui facilite des services de sécurité à faible coût matériel additionnel [8, 35]. Cette approche traite une partie des menaces logicielles, mais reste exposée à des attaques micro-architecturales et de configuration si le durcissement n'est pas appliqué [33, 36, 69].

2.6 Guide de navigation

Le tableau suivant oriente le lecteur du besoin vers le chapitre qui le traite ; chaque chapitre de cas d'usage reste lisible indépendamment, avec son analyse de menace, sa checklist industrielle et sa suite de tests.

Votre besoin	**Où aller**
Choisir entre SE, TPM, TEE et HSE selon la menace	Chapitre 3
Prouver l'authenticité de la carte et de ses composants (FPGA inclus)	Chapitre 4
Garantir l'intégrité du logiciel au démarrage (secure boot)	Chapitre 5
Protéger les commandes et transactions applicatives	Chapitre 6
Sécuriser la personnalisation, les clefs et les identités en usine	Chapitre 7
Mettre à jour la flotte sans créer de brèche (OTA/DFU)	Chapitre 8
Attester, superviser et retirer les équipements déployés	Chapitre 9
Placer et implémenter les opérations cryptographiques	Chapitre 10
Anticiper la menace quantique et organiser la migration PQC	Annexe A
Choisir les bus et durcir les interfaces physiques	Annexe B
Intégrer sur PCB et organiser la MCO	Annexe C
Préparer certification et conformité réglementaire	Annexe D

TABLE 2.1 – Guide de navigation : du besoin au chapitre

3
Choix des mécanismes par cas d'usage

3.1 Quel mécanisme pour authentifier la carte électronique

Ici, la finalité est une **identité matérielle de plateforme** non clonable et vérifiable, associée à l'état de configuration de la carte. Le but est d'assurer que les composants actifs participant à la chaîne de confiance du produit final sont tous présents, intègres et authentiques.

Il est alors nécessaire, au démarrage, d'assurer que ces composants sont aptes à s'authentifier mutuellement, ce qui implique, de manière naturelle, l'utilisation d'une racine de confiance matérielle (SE/HSE/ROM) pour initialiser les secrets et les mesures de plateforme [47, 43]. De plus, chaque composant s'authentifiant doit pouvoir assurer la confidentialité et l'intégrité des secrets liés à son authentification sur la carte, afin de ne pas pouvoir être substitué par modification physique du TCB.

Enfin, lorsqu'il est nécessaire de se prémunir contre des patchs matériels, la chaîne d'authentification entre chaque composant doit alors être résistante aux attaques par rejeu.

Le chapitre 4 détaille l'implémentation de ce cas d'usage.

3.2 Quel mécanisme pour authentifier le logiciel du GPP au boot

Pour ce besoin, le **TPM** est adapté lorsque l'enjeu principal est la **preuve mesurable** de l'état logiciel (firmware, bootloader, noyau) grâce aux PCR et à l'attestation distante [30, 43]. Les attaques de type rollback ou bootkit peuvent être traitées lorsque les mesures sont historisées et vérifiées côté vérificateur [47].

Le **TEE** peut compléter ou remplacer partiellement cette fonction sur certaines plateformes, à condition de disposer d'une RoT matérielle définie (ROM immuable, vérification de signatures, gestion de clefs correcte) [8, 47]. Sans ce socle, les attaques sur le boot peuvent contourner les contrôles applicatifs.

Le **SE** peut stocker les clefs de vérification ou signer des preuves, mais il n'observe pas nativement toute la chaîne de boot GPP sans orchestration explicite [25, 43].

Le chapitre 5 détaille l'implémentation de ce cas d'usage.

3.3 Quel mécanisme pour authentifier les transactions applicatives

Ce cas vise la protection continue des opérations métier (signature de trames, authentification de commandes, chiffrement de données applicatives). Le **SE** est adapté lorsque la clef doit rester inextractible malgré un accès physique prolongé [25, 17]. Le **TEE** est adapté aux décisions locales rapides et au contrôle de flux en mémoire sécurisée [8, 35].

Les attaques de rejeu, d'usurpation et de falsification de commandes imposent des mécanismes anti-rejeu (nonce, compteurs, fenêtres temporelles) et une journalisation probante [61, 53].

Le chapitre 6 détaille l'implémentation de ce cas d'usage.

3.4 Règle pratique de choix

- Prioriser **TPM** pour la mesure et l'attestation d'état de plateforme (conformité et vérification distante) [30, 43]
- Prioriser **SE** pour la protection des clefs applicatives, la résistance physique documentée et les cas de menace d'extraction (profils AM-03/AM-04) [17, 25, 34]
- Utiliser **TEE** pour l'isolation d'exécution locale et la performance, en appliquant un durcissement explicite contre les attaques micro-architecturales [8, 33, 36]
- Concevoir **hybride** (SE + TPM + TEE) lorsque les exigences de preuve distante, de protection physique des secrets et de latence coexistent [47, 27]

3.5 Aide à la sélection renforcée

Le projet Eurisko [19] et le projet WooKey [7] apportent des retours de terrain exploitables pour arbitrer entre **SE**, **TPM** et **TEE/HSE** dans un produit réel, au-delà des comparatifs théoriques.

3.5.1 Ce que ces travaux confirment

- **Valeur d'un SE externe pour la racine de confiance et la pré-authentification.** Le projet Eurisko [19] met en avant une chaîne de démarrage sécurisée avec mécanisme d'authentification pré-boot reposant sur un composant certifié EAL5+, utilisé à la fois comme racine de confiance et comme élément d'authentification extractible.
- **Défense en profondeur par cloisonnement explicite.** Le projet WooKey [7] montre l'intérêt d'un découpage explicite entre modules de données exposés et modules de confiance, avec isolation MPU et séparation des chemins de données et d'authentification.

- **Canal sécurisé obligatoire avec le token.** Le projet WooKey [7] impose une authentification mutuelle SoC-token en début de session (dérivée ECDH, clefs de session, anti-rejeu), ce qui documente une réduction de la surface d'attaque pré-authentification et des scénarios de token malveillant.
- **Mise à jour traitée comme surface d'attaque à couvrir explicitement.** Les deux approches insistent sur le rôle de la chaîne de boot/DFU, avec anti-rollback et contrôles cryptographiques explicites [19, 7, 47].

3.5.2 Faiblesses et compromis à prendre en compte

- **SE seul : protection des secrets, visibilité système limitée.** Un SE protège les clefs avec des mécanismes de résistance physique et de politique d'usage, mais n'apporte pas nativement la mesure complète de l'état logiciel du GPP ; sans mécanisme d'attestation de plateforme, le niveau de preuve reste partiel [17, 25, 43].
- **TEE/HSE seul : performance et intégration, mais exposition résiduelle SoC.** Le projet WooKey [7] rappelle que certaines protections flash/readout du MCU ne sont pas infaillibles face aux fautes, ce qui maintient un risque sur des actifs stockés côté SoC si le secret n'est pas externalisé.
- **Confidentialité sans intégrité : compromis à expliciter.** Le projet WooKey [7] documente que la protection des données au repos peut privilégier la confidentialité (FDE) sans garantir l'intégrité, qui doit alors être traitée à une couche supérieure. Ce choix doit être tracé dans l'analyse de risque et compensé par des contrôles applicatifs.
- **Complexité du durcissement de la mise à jour.** Le projet WooKey [7] discute des scénarios TOCTOU et injection de faute sur vérification de signature, puis introduit un token actif pendant le flux DFU pour contraindre ces scénarios. Cette approche augmente les contrôles appliqués, avec un coût de conception, de test et d'opération élevé.
- **Dépendance opérationnelle au token.** L'architecture à token externe introduit des contraintes de disponibilité, logistique et support (perte, remplacement, personnalisation, récupération), à anticiper dès la phase d'architecture [19, 7].

3.5.3 Règles de décision dérivées

- **Contexte haute menace physique (AM-03/AM-04) :** retenir une approche **SE + TEE/HSE**, avec séparation stricte des domaines de confiance et secret racine hors SoC
- **Contexte priorité preuve distante et gouvernance flotte :** ajouter **TPM** pour bénéficier de mesures et d'attestation standardisées
- **Contexte contraint en coût/énergie sans exigence d'attestation distante formelle : TEE/HSE** peut suffire pour certains usages, sous réserve d'un durcissement explicite de la chaîne de boot, des interfaces de débogage et du mode mise à jour
- **Aucune exigence d'intégrité des données au repos explicitement couverte :** refuser la mise en production tant qu'un mécanisme d'intégrité (mode AEAD ou protection applicative équivalente) n'est pas défini et testé

3.5.4 Checklist d'acceptation avant choix final

- Le démarrage est-il bloqué en absence d'authentification pré-boot valide ?
- Les clefs racines restent-elles hors du domaine mémoire non fiable du SoC ?

- Le mode DFU est-il protégé contre rollback, TOCTOU et fautes simples plausibles ?
- Les flux critiques disposent-ils de confidentialité **et** d'intégrité, pas seulement de chiffrement ?
- Le plan opérationnel couvre-t-il perte/renouvellement/révocation des tokens et incidents supply chain ?

4

Authentifier la carte électronique

4.1 Analyse de la menace et des risques

Ce cas d'usage vise à établir une identité matérielle non clonable de la carte et à prouver que les composants actifs de la chaîne de confiance sont présents, intègres et authentiques dès le démarrage. L'objectif n'est donc pas seulement de vérifier un logiciel, mais aussi d'empêcher la substitution d'une plateforme entière, la modification de la chaîne de personnalisation et l'usage non autorisé d'un équipement de terrain.

Les menaces dominantes sont les suivantes :

- clonage ou substitution de carte en production, en déploiement ou sur le terrain ;
- modification de composants, de firmware ou de paramètres de personnalisation ;
- attaques par faute (glitch tension/horloge, coupure, perturbation de l'exécution) visant la vérification initiale ou les échanges de clefs ;
- attaques par canaux auxiliaires destinées à l'extraction de secrets racines, de signatures ou de données de provisioning ;
- rejeu de messages ou de sessions d'authentification, y compris pendant les phases de maintenance ou de mise à jour ;
- abus d'interfaces de service, de débogage ou de récupération, qui peuvent créer une voie de contournement de la chaîne de confiance.

La modélisation de risque doit distinguer les actifs critiques (identité matérielle, clefs racines, politiques d'usage, journal d'audit, état de plateforme) et les chemins d'attaque réalistes selon le profil de l'attaquant, la valeur du secret et la difficulté de maintenance du produit en environnement réel [5, 53]. En pratique, la menace physique et la menace de chaîne d'approvisionnement doivent être traitées ensemble, car une carte authentique au moment de la personnalisation peut devenir non fiable si la confiance est contournée plus tard.

4.1.1 Traçabilité des attaques, attaquants et contrôles

Scénario d'attaque	Menace de référence	Profils AM	Objectifs OS	Contrôles associés
Clonage ou substitution de carte en terrain	Usurpation d'identité de plateforme	AM-01, AM-04	OS-03, OS-05	Identité matérielle, preuve d'authenticité, anti-rejeu
Injection de firmware non autorisé via maintenance ou provisioning	Compromission du boot et du firmware	AM-02, AM-04	OS-02, OS-03	Vérifications cryptographiques au boot, contrôle de version, blocage sur échec
Extraction de secrets racine par faute ou canaux auxiliaires	Extraction de secrets et d'éléments sensibles	AM-03	OS-01, OS-04	Isolation des secrets, canaux protégés, politiques d'usage des clefs
Rejeu de session ou de preuve d'authentification	Altération d'opérations critiques	AM-01, AM-03	OS-02, OS-05	Nonces, compteurs monotones, fenêtres temporelles, journalisation
Abus d'interfaces de service ou de débogage	Abus de maintenance et de débogage	AM-02, AM-03	OS-03, OS-05	Authentification mutuelle, traçabilité des opérations, durcissement des interfaces

TABLE 4.1 – Traçabilité des attaques du chapitre "Authentifier la carte électronique" vers menaces, profils AM et objectifs OS

4.2 Fonctions de sécurité attendues

Pour répondre à cette menace, l'architecture doit intégrer plusieurs fonctions de sécurité complémentaires :

- **Racine de confiance matérielle :** un composant initialement fiable (SE, HSE, ROM immuable ou combinaison) doit décider de la confiance au début du démarrage.
- **Authentification mutuelle des composants actifs :** le MCU, le SE, le TPM ou les éléments réseau doivent prouver leur identité avant de s'échanger des secrets ou des ordres critiques.
- **Protection des secrets et des canaux :** les échanges inter-composants doivent garantir confidentialité, intégrité, anti-rejeu et authentification de l'origine (SCP, sessions TPM, TLS mutuel, dérivation de session).
- **Anti-rejeu et anti-rollback :** les preuves, les mesures de plateforme et les mises à jour doivent être liées à un contexte de session et à une version minimale acceptable.
- **Tolérance aux fautes et mode fail-secure :** toute détection d'anomalie doit conduire à un blocage ou à une maintenance tracée, plutôt qu'à une transition vers un état opérationnel ambigu.
- **Traçabilité et audit :** les opérations sensibles doivent produire des traces exploitables pour la supervision, l'investigation et la conformité.

Ces fonctions ne sont pas de simples mécanismes additionnels ; elles constituent le socle permettant d'obtenir une authentification de carte robuste, vérifiable et résistante à la fois aux attaques logicielles et physiques [47, 43, 25].

4.3 Attaques typiques et contre-mesures

4.3.1 Clonage et substitution de cartes

Un attaquant peut tenter de remplacer une carte par un équivalent non autorisé, de copier ses attributs ou d'utiliser un composant de remplacement pendant la personnalisation. La protection repose sur une identité matérielle liée à une racine non clonable, à des certificats ou preuves cryptographiques et à des mécanismes de vérification embarqués dès le premier démarrage. Cette problématique est centrale dans les architectures de cartes sécurisées et de plateformes de confiance, où la résistance physique et la gouvernance de la personnalisation sont décrites dans les spécifications industrielles et les évaluations de sécurité [25, 17, 38].

4.3.2 Injection de firmware et modification de la personnalisation

La chaîne de personnalisation, la maintenance ou la mise à jour constituent des cibles privilégiées. Une image modifiée peut être injectée ou une reprise de version ancienne peut être utilisée pour réintroduire un comportement vulnérable. Les contre-mesures incluent la vérification cryptographique à chaque étape, le contrôle de version, l'anti-rollback et la séparation des rôles de personnalisation et de test. La littérature de durcissement de firmware et de chaîne de confiance documente cette menace, en particulier dans les guides de résilience des firmwares et de sécurité de la chaîne d'approvisionnement [47, 52, 4, 3]. Les retours d'expérience issus de plateformes sécurisées ou de projets de recherche, notamment le projet Eurisko [19] et le projet WooKey [7], complètent cette analyse.

4.3.3 Fautes physiques et canaux auxiliaires

Les attaques par fautes (variation de tension, d'horloge, perturbation lumineuse) ou par canaux auxiliaires (consommation, émission électromagnétique) visent à falsifier une vérification ou à extraire un secret critique. La réponse consiste à durcir le code de vérification, à dissocier les secrets de la mémoire non fiable, à limiter les opérations sensibles et à placer les calculs critiques dans un domaine de confiance protégé. Ces familles d'attaques sont décrites de manière classique dans la littérature académique sur les fautes cryptographiques et les canaux auxiliaires [12, 13, 34, 14]. Elles se recoupent également avec les travaux plus récents sur les vulnérabilités micro-architecturales et les exécutions spéculatives, qui montrent que l'attaque ne se limite pas au seul niveau logique ou matériel [33, 36, 15, 69].

4.3.4 Rejeu, abus de maintenance et contournement de la chaîne de confiance

Un attaquant peut réinjecter une preuve ancienne, détourner une session de maintenance ou utiliser une interface de service non protégée pour obtenir un état de confiance non prévu. Le mécanisme de sécurité doit donc intégrer des nonces, des compteurs monotones, un contrôle d'accès explicite et une journalisation des opérations de maintenance et d'attestation. Les protocoles et architectures de sécurité associés à l'attestation, aux sessions sécurisées et à l'authentification mutuelle sont détaillés dans les spécifications et recommandations de référence [61, 63, 30, 25].

4.4 Principe d'amorçage de confiance

L'amorçage de confiance est établi lorsque la décision initiale est prise dans une racine de confiance matérielle (SE, HSE, ROM immuable ou combinaison) avant tout code mutable. Ce principe limite les contournements précoces typiques des campagnes bootkit et réduit la fenêtre d'attaque pré-authentification [47, 42].

La séquence recommandée est la suivante :

- vérification cryptographique du premier code exécutable ;
- mesure des étapes critiques de la chaîne de boot ;
- activation progressive des ressources selon un état valide ;
- déclaration d'authentification uniquement si l'intégrité et la provenance restent conformes.

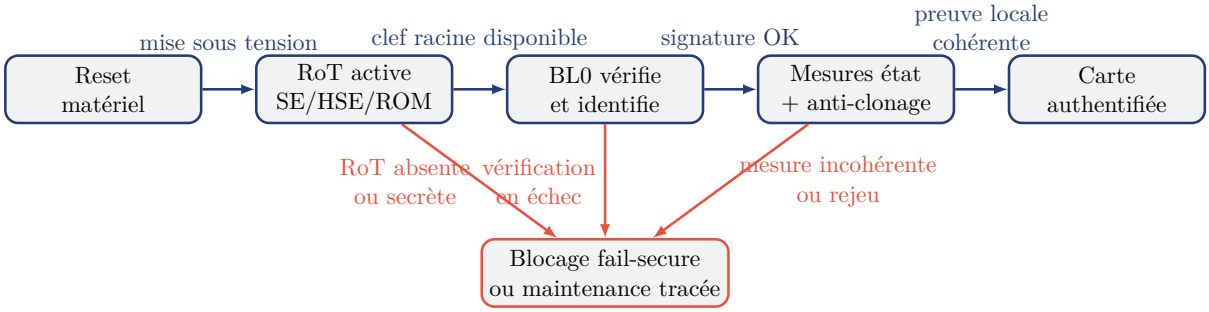


FIGURE 4.1 – Automate d'état de la chaîne de boot pour le cas d'usage "Authentifier la carte électronique"

Descriptif synthétique

L'automate formalise une chaîne de boot centrée sur l'identité matérielle : la racine de confiance active d'abord les secrets, le premier code exécutable est vérifié, puis les mesures de plateforme sont consolidées avant de déclarer la carte authentifiée. Toute perte de racine, échec de vérification, incohérence de mesure ou anomalie de session bascule immédiatement vers un état de blocage *fail-secure* ou de maintenance explicitement tracée.

4.5 Spécifications de sécurité pour l'implémentation

Démarrer le composant de sécurité en premier. Cette contrainte réduit la fenêtre d'attaque pré-authentification et évite qu'un code non fiable ne prenne la main avant l'établissement de la confiance [47].

Authentification mutuelle des composants actifs. L'usage d'une dérivation de clef unique par équipement, d'un PUF ou d'une identité persistante appuyée par un secret de fabrication réduit les scénarios de clonage, de substitution et de rejeu inter-dispositifs [38, 51].

Protection des échanges inter-composants. Les canaux entre MCU, SE, TPM et éléments réseau doivent assurer confidentialité, intégrité, anti-rejeu et authentification de l'origine (SCP, sessions TPM, TLS mutuel) [25, 30, 61].

Isolation des secrets et séparation des rôles. Les secrets racines doivent rester hors du domaine mémoire non fiable du SoC, et les opérations de provisioning, de test et de maintenance doivent être séparées pour limiter les compromissions de masse [49, 52, 4].

4.6 Composants configurables (FPGA) dans la chaîne de confiance

Un FPGA n'est pas un composant de sécurité : il n'offre par lui-même ni résistance physique certifiée, ni politique d'usage de clefs, ni attestation standardisée, et n'a donc pas sa place dans le comparatif SE/TPM/TEE des chapitres de choix. Il est en revanche un **composant actif de la chaîne de confiance** dès lors qu'il est présent sur la carte : sa configuration (le bitstream) détermine son comportement au même titre qu'un firmware, et un bitstream substitué équivaut à une injection de firmware qui contourne l'ensemble des contrôles imposés au GPP.

4.6.1 Le bitstream est un firmware

Le bitstream doit être traité avec exactement la grille appliquée aux logiciels du produit :

- **Authenticité et intégrité :** bitstream signé et vérifié avant chargement, en utilisant les mécanismes natifs du composant lorsqu'ils existent (authentification de bitstream des familles récentes) ou une vérification portée par la racine de confiance de la carte ;
- **Confidentialité :** chiffrement du bitstream lorsque le design contient de la propriété intellectuelle ou des éléments exploitables par un attaquant, avec des clefs gérées comme des clefs racines (provisioning contrôlé, non-exportabilité, plan de compromission) ;
- **Anti-rollback :** contrôle de version du bitstream lorsque le composant le supporte, sinon compensation au niveau de la chaîne de boot qui charge et vérifie le bitstream ;
- **Mesure et traçabilité :** la version et le condensat du bitstream participent aux mesures de plateforme et à l'attestation, et le bitstream est inclus dans la SBOM et le registre de build signé [47, 52].

La mise à jour du bitstream suit le flux DFU durci du chapitre "Mettre à jour le firmware en exploitation (OTA/DFU)" : manifeste signé, installation réversible, revalidation au chargement.

4.6.2 Attaques connues et limites des protections natives

Les protections de bitstream des FPGA ont un historique documenté qui interdit de leur accorder une confiance aveugle : extraction de clefs de chiffrement de bitstream par analyse de consommation sur plusieurs générations de composants [39], et rupture complète du chiffrement de bitstream des Xilinx série 7 par l'attaque Starbleed, non corrigeable sur le silicium déployé [20]. Les interfaces de configuration et de test (JTAG, SelectMAP, readback, reconfiguration partielle) constituent par ailleurs des voies de contournement analogues aux interfaces de débogage du MCU et doivent être désactivées ou verrouillées en production [47].

La conséquence architecturale est la même que pour les protections flash des MCU évoquées par le projet WooKey [7] : les secrets de forte valeur ne doivent pas résider dans le domaine

FPGA si le modèle de menace inclut un attaquant de laboratoire (AM-03) ; ils restent dans le SE ou le HSE, le FPGA consommant des clefs de session dérivées et révocables.

4.6.3 Le FPGA comme support de racine de confiance

À l'inverse, certaines familles orientées sécurité (secure boot natif, contre-mesures aux canaux auxiliaires, PUF intégré) peuvent servir d'ancre de confiance de la carte, et un FPGA permet d'implémenter des fonctions de sécurité auditables : soft-core de racine de confiance, accélérateurs cryptographiques, prototypage de designs ouverts type OpenTitan [37, 38]. Cette approche a un intérêt particulier pour les architectures ouvertes, où l'auditabilité du design compense l'absence de certification tierce ; elle impose en contrepartie la même rigueur de chaîne de build et de vérification que pour tout composant de confiance [52], et une évaluation explicite de la résistance physique réellement obtenue, qui reste généralement inférieure à celle d'un SE certifié [17].

En synthèse : le FPGA se traite comme un **porteur de firmware à part entière** dans l'analyse de menace, les checklists et les tests de ce chapitre (substitution de composant, injection de firmware, verrouillage des interfaces), et ne se substitue à un composant de sécurité certifié que sur décision argumentée par l'analyse de risque.

4.7 Checklist industrielle et suite de tests synthétique

4.7.1 Checklist des actions côté industriel

ID	Action à réaliser	Responsable	Preuve attendue
CARTE-CL-01	Geler la nomenclature de la carte et la liste des composants de confiance (SE/HSE/TPM, MCU, mémoire de boot)	Archi produit + industrialisation	BOM signée et versionnée
CARTE-CL-02	Définir et valider la chaîne de personnalisation (sites, rôles, secrets, séparation des tâches)	RSSI + production	Procédure de provisioning approuvée
CARTE-CL-03	Verrouiller les interfaces de débogage/service en mode production	Hardware + firmware	Rapport de configuration et vérification du verrouillage
CARTE-CL-04	Mettre en place la politique d'identité matérielle et d'authentification mutuelle inter-composants	Firmware sécurité	Dossier de conception + tests d'authentification
CARTE-CL-05	Définir le plan de gestion incident (substitution carte, compromission secrets, lot suspect)	Qualité + SOC/CSIRT	Playbook incident et matrice d'escalade

Table 4.2 – Checklist industrielle pour "Authentifier la carte électronique"

4.7.2 Suite de tests synthétique

ID	Test	Méthode	Criticité	Critère de réussite
CARTE-T-01	Tentative de démarrage avec composant substitué	Test d'intégration + variation BOM	Haute	Démarrage refusé, événement d'audit émis
CARTE-T-02	Injection firmware non autorisé au provisioning	Test négatif en banc	Haute	Image rejetée, aucune activation opérationnelle
CARTE-T-03	Rejeu de preuve d'authentification inter-composants	Test protocolaire	Moyenne	Session refusée (nonce/compteur invalide)
CARTE-T-04	Vérification du mode fail-secure sur erreur de mesure	Fault injection contrôlée	Haute	Basculer en blocage/-maintenance tracée
CARTE-T-05	Audit de traçabilité des opérations sensibles	Revue de journaux	Moyenne	100% des refus avec `reason_code` exploitable

TABLE 4.3 – Suite de tests synthétique pour "Authentifier la carte électronique"

5

Authentifier le logiciel du processeur généraliste (GPP) au boot

5.1 Analyse de la menace et des risques

Les attaques typiques ciblent la chaîne de mise à jour, les mécanismes de signature, la protection anti-rollback et la configuration secure boot. Les campagnes réelles montrent que la compromission CI/CD peut diffuser un firmware légitime en apparence mais malveillant [52, 4].

La criticité est corrélée à l'exposition distante et à la difficulté de patch en environnement de terrain [24, 53].

5.1.1 Traçabilité des attaques, attaquants et contrôles

Scénario d'attaque	**Menace de référence**	**Profils AM**	**Objectifs OS**	**Contrôles associés**
Downgrade vers image signée ancienne et vulnérable	Compromission du boot et du firmware	AM-01, AM-04	OS-02, OS-03	Anti-rollback, version minimale, politiques de vérification
Injection d'image modifiée dans la chaîne CI/CD	Compromission du boot et du firmware	AM-04	OS-02, OS-05	Séparation des rôles de signature, attestation d'état, traçabilité build
Bypass vérification signature par faute synchronisée	Altération d'opérations critiques	AM-03	OS-02, OS-04	Contre-mesures fault injection, vérification multi-étapes, mode fail-secure
Contournement de la chaîne de confiance via interface de débogage active	Abus de maintenance et de débogage	AM-02, AM-03	OS-03, OS-05	Durcissement du débogage, journalisation, contrôles d'accès maintenance

TABLE 5.1 – Traçabilité des attaques du chapitre "Authentifier le logiciel du processeur généraliste (GPP) au boot" vers menaces, profils AM et objectifs OS

5.2 Chaîne de confiance logicielle

La chaîne de confiance doit implémenter :

- vérification de signature à chaque étape
- mesure d'intégrité cumulative
- contrôle de version anti-rollback
- politique explicite de mode dégradé et recovery

Ce modèle est cohérent avec les recommandations firmware resiliency NIST [47, 42].

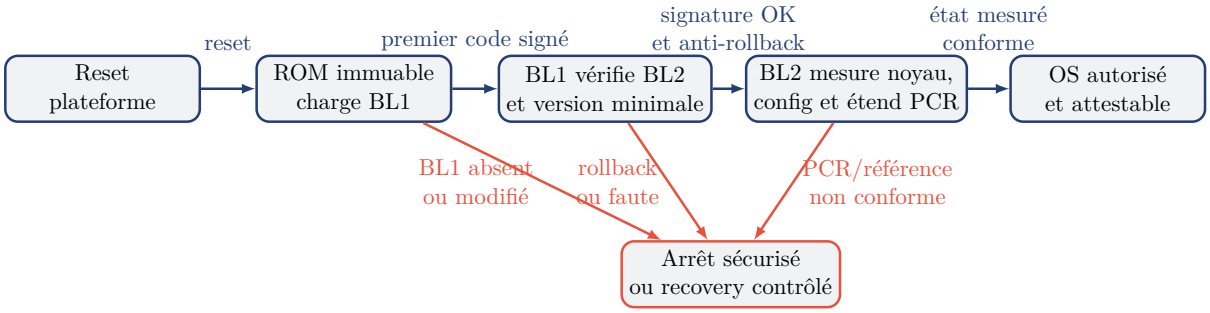


FIGURE 5.1 – Automate d'état de la chaîne de boot pour le cas d'usage "Authentifier le logiciel du processeur généraliste (GPP) au boot"

Descriptif synthétique

Cet automate décrit la progression minimale d'un boot mesuré du GPP : ROM immuable, vérification successive des bootloaders, contrôle anti-rollback, puis extension des mesures avant libération du noyau. Le passage à l'état final n'est autorisé que si l'intégrité et la version restent conformes ; sinon la plateforme tombe en arrêt sécurisé ou en recovery contrôlé.

5.3 SE/HSE versus TPM

Avec **SE/HSE**, la vérification est généralement locale, avec un coût énergie/latence limité. Avec **TPM**, la preuve d'état est standardisée et exploitable à distance [66, 43]. Une architecture combinée (SE/HSE pour décision locale, TPM pour évidence distante) permet un compromis entre ces contraintes.

5.4 Spécifications de sécurité pour l'implémentation

5.4.1 Politique de vérification des images logicielles

Définir les algorithmes autorisés, tailles minimales, autorités de signature et règles de révocation. Éviter les primitives obsolètes et aligner la politique avec les transitions NIST/ANSSI [50, 1, 51].

5.4.2 Mesure et attestation de l'état logiciel

Lister les composants mesurés (ROM ext., BL1, BL2, kernel, config) et le schéma d'attestation. Les valeurs de référence (golden measurements) doivent être gouvernées et versionnées [43, 30].

5.4.3 Cycle de vie des clefs de signature

Documenter génération, escrow éventuel, rotation, révocation et destruction. La compromission d'une clef de build impose un plan de réémission et de reprise de confiance à l'échelle flotte [52, 51].

5.5 Checklist industrielle et suite de tests synthétique

5.5.1 Checklist des actions côté industriel

ID	Action à réaliser	Responsable	Preuve attendue
GPP-CL-01	Formaliser la politique secure boot (algorithmes, autorités, révocation, anti-rollback)	Architecture sécurité	Politique signée et versionnée
GPP-CL-02	Sécuriser la chaîne CI/CD des images de boot (séparation rôles, signature, traçabilité)	DevSecOps	Registre de build signé + SBOM
GPP-CL-03	Constituer les mesures de référence (golden) et la procédure de mise à jour contrôlée	Firmware + exploitation	Base golden validée et historisée
GPP-CL-04	Désactiver/protéger les voies de contournement boot (débogage, récupération non autorisée)	Intégration plateforme	Rapport de configuration durcie
GPP-CL-05	Définir la procédure flotte en cas de compromission de clef de signature	RSSI + opérations	Plan de rotation/réémission testé

TABLE 5.2 – Checklist industrielle pour "Authentifier le logiciel du processeur généraliste (GPP) au boot"

5.5.2 Suite de tests synthétique

ID	Test	Méthode	Criticité	Critère de réussite
GPP-T-01	Tentative de boot sur image signée mais rollbackée	Test d'intégration boot	Haute	Démarrage refusé (version minimale violée)
GPP-T-02	Injection d'image non signée dans pipeline de release	Test CI/CD + contrôle signature	Haute	Publication bloquée et alerte émise
GPP-T-03	Bypass de vérification par faute simple	Test de robustesse (fault campaign)	Haute	Pas de passage en état opérationnel
GPP-T-04	Vérification de cohérence attestation/mesures golden	Test d'attestation	Moyenne	Équipement non conforme refusé
GPP-T-05	Validation de la traçabilité des décisions de boot	Revue de journaux	Moyenne	Décisions `ALLOW/DENY` complètes et exportables

TABLE 5.3 – Suite de tests synthétique pour "Authentifier le logiciel du processeur généraliste (GPP) au boot"

6

Authentifier des transactions applicatives après le boot

6.1 Analyse de la menace et des risques

Les menaces dominantes sont le rejeu de transaction, l'usurpation d'identité applicative, la manipulation de payload et l'escalade de privilèges locale. Les attaques par canal auxiliaire restent applicables si les clefs applicatives sont abondamment sollicitées [34, 14, 61].

6.1.1 Traçabilité des attaques, attaquants et contrôles

Scénario d'attaque	**Menace de référence**	**Profils AM**	**Objectifs OS**	**Contrôles associés**
Rejeu de commandes légitimes sur canal applicatif	Altération d'opérations critiques	AM-01, AM-02	OS-02, OS-03	Nonces/compteurs, fenêtres temporelles, vérification de contexte
Usurpation d'identité applicative après compromission locale	Usurpation d'identité de plateforme	AM-02, AM-04	OS-03, OS-05	Authentification forte composant-service, attestations périodiques
Manipulation de payload en transit (MITM local)	Altération d'opérations critiques	AM-01, AM-02	OS-02, OS-03	Intégrité bout-en-bout, MAC/signature, sessions authentifiées
Exfiltration ou extraction de clef applicative par canal auxiliaire	Extraction de secrets et d'éléments sensibles	AM-03	OS-01, OS-04	Usage non exportable des clefs, cloisonnement SE/TEE, contrôle des usages

TABLE 6.1 – Traçabilité des attaques du chapitre "Authentifier des transactions applicatives après le boot" vers menaces, profils AM et objectifs OS

6.2 Objectif post-boot

L'objectif est d'assurer qu'une transaction est émise par un composant autorisé, dans un état autorisé, et qu'elle reste inaltérée jusqu'au vérificateur. Cela repose sur des preuves cryptographiques courtes, contextualisées et auditables [63, 53].

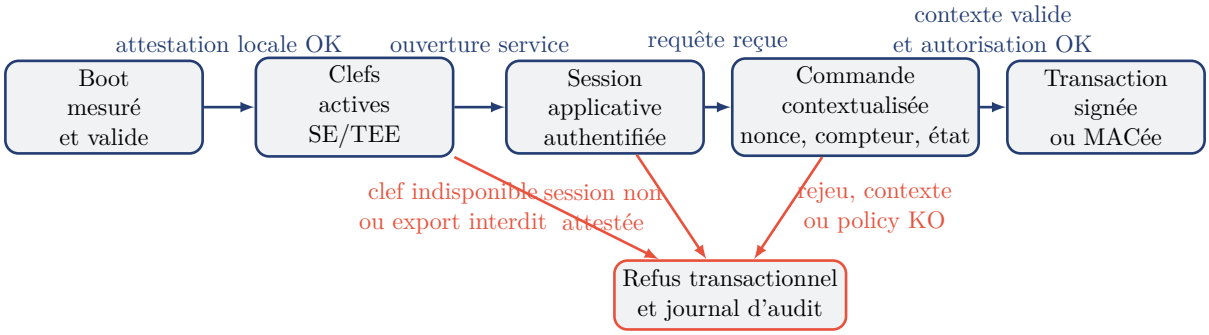


FIGURE 6.1 – Automate d'état reliant le boot de confiance au cas d'usage "Authentifier des transactions applicatives après le boot"

Descriptif synthétique

Pour ce cas d'usage, l'automate relie explicitement la chaîne de boot à l'autorisation transactionnelle : les clefs ne sont activées qu'après un boot valide, la session applicative doit rester authentifiée, puis chaque commande est contrôlée avec contexte, anti-rejeu et politique d'usage. Toute divergence d'état, indisponibilité de clef ou violation de politique entraîne un refus immédiat assorti d'une trace d'audit.

6.3 Répartition des rôles SE et TEE

SE : conservation des clefs racines, signatures non exportables, politiques d'usage strictes [25, 17].

TEE : vérification de contexte d'exécution, décisions dynamiques, accélération locale [8, 35].

Pattern recommandé : clef racine en SE, clef session dérivée et usage court en TEE, avec re-attestation périodique.

6.4 Spécifications de sécurité pour l'implémentation

6.4.1 Services à authentifier après le boot

L'implémentation doit commencer par un inventaire des services exposés après le boot, puis une classification par impact métier et impact sécurité. Les familles minimales à couvrir sont :

- commandes d'actionneurs (ouverture/fermeture, mise en marche/arrêt)

- télémétrie sensible (mesures critiques, état d'équipement, alarmes)
- gestion de configuration (paramètres réseau, seuils de sécurité, profils)
- opérations d'administration (provisioning, rotation de clefs, politique de mise à jour)

Pour rendre les exigences auditables, on définit quatre niveaux de preuve applicative :

- **N0 – Faible criticité :** authentification canal seulement (ex : TLS mutuel), sans signature métier par message
- **N1 – Standard :** authentification canal + intégrité message (MAC) + anti-rejeu basique (nonce ou compteur)
- **N2 – Renforcé :** N1 + preuve liée à l'état de l'équipement (contexte TEE/SE) + autorisation fine par type de commande
- **N3 – Critique/sûreté :** N2 + signature non exportable en SE/TPM + contre-signature ou validation serveur forte + journal probant immédiat

Exemple de mapping initial recommandé :

Service	Niveau	Mécanismes minimaux	Exemple
Commande actionneur critique	N3	Signature SE, compteur monotone, vérification de contexte d'état	Ordre d'ouverture de vanne industrielle
Configuration sécurité	N2	MAC + attestation périodique + RBAC strict	Changement de liste blanche réseau
Télémétrie sensible	N1/N2	MAC, timestamp, fenêtre anti-rejeu	Envoi de mesures de process
Télémétrie non sensible	N0/N1	Authentification du canal + intégrité transport	Statut batterie non critique

TABLE 6.2 – Exemple de classification des services post-boot et niveau de preuve associé

Le niveau retenu doit être justifié par l'analyse de risque et relié aux objectifs OS-01..OS-05 du chapitre [27, 3, 53].

6.4.2 Politique d'usage des clefs applicatives

La politique de clefs doit être explicite, versionnée et appliquée par le composant de confiance (SE/HSE/TPM/TEE selon architecture). Elle couvre au minimum :

- **Propriété et cycle de vie :** création, activation, suspension, rotation, révocation, destruction
- **Non-exportabilité :** clefs racines et clefs de signature N2/N3 non exportables
- **Contexte d'usage :** usages autorisés (signer, vérifier, chiffrer), périmètre de service, plage horaire, état machine
- **Limites d'usage :** quotas par période, seuils d'erreurs consécutives, verrouillage temporaire puis définitif
- **Séparation des clefs :** une clef par finalité (authentification, chiffrement, mise à jour, administration) pour éviter les usages croisés

Exemple de règle concrète (service N3) :

- la clef `K_cmd_critique` signe uniquement la famille `ACTUATOR_CTRL`
- signature autorisée seulement si compteur local synchronisé et attestation valide depuis moins de 10 minutes

- blocage après 5 échecs de vérification consécutifs
- rotation forcée tous les 90 jours ou immédiate en cas d'incident

Exemple de structure de payload à signer :

```
{
 "device_id": "A1F4-22",
 "service": "ACTUATOR_CTRL",
 "cmd": "OPEN_VALVE",
 "nonce": "7f9c...",
 "ctr": 184233,
 "ts": "2026-08-09T10:45:12Z",
 "state_ref": "TEE_OK:4d2a"
}
```

La vérification doit échouer si un champ contextuel critique manque, si le compteur recule, ou si la fenêtre temporelle est dépassée [51, 53, 61].

6.4.3 Traçabilité et audit des opérations cryptographiques

La traçabilité doit permettre de répondre à trois questions en audit : *qui* a demandé l'opération, *quoi* a été exécuté, et *dans quel contexte de sécurité*. Les journaux doivent donc être :

- horodatés (source temps fiable)
- liés à l'identité équipement et à l'identité service
- reliés à l'identifiant de clef
- protégés contre altération (hash-chain ou signature périodique)
- exportables vers un collecteur central avec rétention définie

Champs minimaux d'un événement d'audit cryptographique :

- `event_id`, `ts`, `device_id`, `service`, `op_type`
- `key_id`, `policy_version`, `result`, `reason_code`
- `nonce/ctr`, `attestation_ref`, `prev_hash`

Exemple de trace compacte :

```
2026-08-09T10:45:12Z evt=9c2e dev=A1F4-22 svc=ACTUATOR_CTRL
op=SIGN key=K_cmd_critique policy=v12 res=DENY reason=CTR_ROLLBACK
ctr=184200 att=TEE_OK:4d2a prev=1e88...
```

6.4.4 Critères d'acceptation et tests minimaux

Pour éviter des exigences déclaratives, chaque niveau N0..N3 doit avoir des tests d'acceptation automatisables :

- **Anti-rejeu :** rejeu d'un message signé déjà accepté ⇒ refus systématique
- **Contexte invalide :** attestation expirée ou état non conforme ⇒ refus pour N2/N3
- **Robustesse de la politique :** tentative d'usage d'une clef hors périmètre ⇒ refus + journal d'alerte
- **Traçabilité :** toute décision `DENY` doit produire un événement complet et exportable

- **Mode dégradé :** en perte de connectivité, seules les opérations explicitement autorisées en local restent actives

Exemple de règle d'acceptation simple :

- service critique classé N3
- taux maximal de faux rejet applicatif $< 0.1\%$ sur campagne de 10 000 requêtes
- 100% des rejets reliés à un `reason_code` normalisé

Cette formalisation facilite l'alignement avec les pratiques de gouvernance, d'audit et de gestion du risque demandées par les référentiels industriels [31, 53, 27].

6.5 Checklist industrielle et suite de tests synthétique

6.5.1 Checklist des actions côté industriel

ID	Action à réaliser	Responsable	Preuve attendue
TRX-CL-01	Inventorier les services post-boot et classer chaque flux en niveau N0..N3	Product owner + sécurité	Matrice service/niveau validée
TRX-CL-02	Déployer la politique d'usage des clefs applicatives (non-export, quotas, contexte)	Responsable PKI/cryptographie	Politique versionnée + règles appliquées
TRX-CL-03	Implémenter anti-rejeu unifié (nonce, compteur, fenêtre temporelle) sur tous flux critiques	Équipe applicative	Dossier d'architecture + tests d'intégration
TRX-CL-04	Activer la journalisation probante des opérations `ALLOW/DENY`	Exploitation + SOC	Schéma d'audit et preuve d'export central
TRX-CL-05	Définir le mode dégradé autorisé en perte d'attestation/connectivité	Architecture + métier	Procédure MCO et règles de continuité

Table 6.3 – Checklist industrielle pour "Authentifier des transactions applicatives après le boot"

6.5.2 Suite de tests synthétique

ID	Test	Méthode	Criticité	Critère de réussite
TRX-T-01	Rejeu d'un message signé déjà accepté	Test protocolaire	Haute	Rejet systématique + alerte anti-rejeu
TRX-T-02	Envoi commande critique avec attestation expirée	Test négatif applicatif	Haute	Refus N2/N3 avec code d'erreur explicite
TRX-T-03	Usage d'une clef hors périmètre de service	Test de politique cryptographique	Haute	Signature refusée + trace d'audit complète
TRX-T-04	Vérification complétude des journaux de refus	Revue de logs sur campagne	Moyenne	100% des `DENY` corrélables à un `reason_code`
TRX-T-05	Simulation perte de connectivité et mode dégradé	Test de résilience opérationnelle	Moyenne	Seuls services explicitement autorisés restent actifs

Table 6.4 – Suite de tests synthétique pour "Authentifier des transactions applicatives après le boot"

7

Provisionner les secrets et les identités en production

7.1 Analyse de la menace et des risques

La phase d'industrialisation est le moment du cycle de vie où les actifs les plus critiques du produit (clefs racines, identités, paramètres de sécurité) sont manipulés dans l'environnement le moins maîtrisé : sites de production distants, sous-traitance, cadence élevée, personnel nombreux. Une compromission à ce stade a un effet de masse : contrairement à une attaque de terrain qui vise un équipement isolé, une attaque de la chaîne de personnalisation peut affecter un lot entier, voire toute la flotte si un secret commun est exposé [4, 52].

Les menaces dominantes sont les suivantes :

- exfiltration de secrets pendant la personnalisation (clefs maîtres, données de diversification, certificats) ;
- sur-production et clonage : fabrication d'unités hors quota porteuses d'identités valides ;
- substitution de composants ou de firmware dans la chaîne d'approvisionnement ;
- persistance d'un mode de test, de débogage ou d'un firmware de pré-production après la sortie d'usine ;
- compromission du poste ou du serveur de personnalisation, permettant l'injection de clefs connues de l'attaquant ;
- corruption des bases d'identités : association falsifiée entre numéro de série, clefs et certificats.

Le profil d'attaquant dominant est ici l'initié ou le sous-traitant compromis (AM-02), éventuellement combiné à une intervention de terrain ultérieure (AM-04). L'analyse de risque doit couvrir chaque site et chaque transfert de responsabilité, y compris la logistique entre fabrication du composant, assemblage de la carte et personnalisation finale [5, 4].

7.1.1 Traçabilité des attaques, attaquants et contrôles

7.2 Fonctions de sécurité attendues

- **Séparation des rôles et des sites :** la génération des clefs, la personnalisation, le test et l'expédition relèvent de responsabilités distinctes ; aucun acteur seul ne doit pouvoir

Scénario d'attaque	Menace de référence	Profils AM	Objectifs OS	Contrôles associés
Exfiltration de clefs maîtres depuis le site de personnalisation	Extraction de secrets et d'éléments sensibles	AM-02, AM-04	OS-01, OS-05	Frontière HSM, export chiffré uniquement, cérémonies de clefs
Sur-production d'unités porteuses d'identités valides	Usurpation d'identité de plateforme	AM-02	OS-03, OS-05	Quotas signés, compteurs de production, rapprochement comptable
Injection de firmware ou de composant substitué en assemblage	Compromission du boot et du firmware	AM-02, AM-04	OS-02, OS-03	Vérification cryptographique en fin de ligne, contrôle de provenance
Poste de personnalisation compromis injectant des clefs connues	Extraction de secrets et d'éléments sensibles	AM-02	OS-01, OS-02	Génération des secrets dans le HSM/SE, canal sécurisé de bout en bout
Sortie d'usine avec interfaces de test ou de débogage actives	Abus de maintenance et de débogage	AM-01, AM-02	OS-03, OS-04	Verrouillage de fin de ligne, autotest de sortie, audit de configuration

TABLE 7.1 – Traçabilité des attaques du chapitre "Provisionner les secrets et les identités en production" vers menaces, profils AM et objectifs OS

produire une unité authentique complète [52, 4].

- **Frontière HSM explicite :** les secrets sont générés et conservés dans un HSM d'usine (ou directement dans le SE cible) ; tout export se fait sous forme chiffrée (*key wrapping*), jamais en clair, même transitoirement [49, 51].
- **Diversification par équipement :** chaque unité reçoit des clefs dérivées d'un identifiant unique (KDF depuis une clef maître, ou secret issu d'un PUF), de sorte que la compromission d'une unité ne compromette pas le lot [51, 38].
- **Comptage et quotas de production :** le nombre d'identités émises est borné cryptographiquement (quotas signés, compteurs monotones côté HSM) et rapproché de la production physique déclarée.
- **Traçabilité unitaire :** chaque personnalisation produit un enregistrement signé (numéro de série, lot, références de clefs installées, résultat des tests) exporté vers un référentiel central.
- **Verrouillage de fin de ligne :** avant expédition, les interfaces de test et de débogage sont désactivées (fusibles, OTP), le firmware de production est vérifié et l'équipement passe dans un état transport contrôlé [47, 3].

Descriptif synthétique

L'automate formalise la personnalisation en usine : la session avec le HSM d'usine doit être mutuellement authentifiée avant toute injection, les clefs sont diversifiées par équipement, chaque identité émise décrémente un quota signé, et la ligne se termine par le verrouillage et l'autotest de sortie. Tout échec d'authentification, d'injection, de comptage ou de verrouillage envoie l'unité en quarantaine tracée, jamais en expédition.

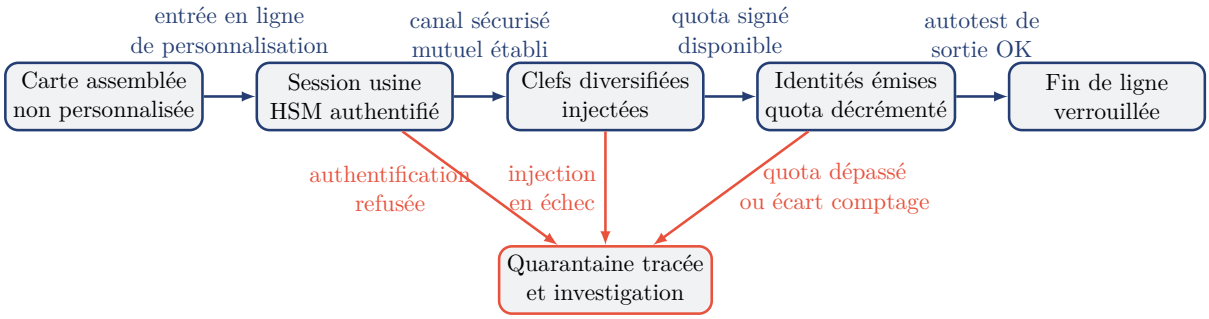


FIGURE 7.1 – Automate d'état de la personnalisation pour le cas d'usage "Provisionner les secrets et les identités en production"

7.3 Identités de fabrication et identités opérationnelles

Le modèle recommandé distingue deux familles d'identités, à la manière des identités IDevID et LDevID normalisées par IEEE 802.1AR [28] :

- **Identité de fabrication (type IDevID) :** installée en usine, portée par un certificat émis par une autorité de certification dédiée à la production, elle prouve l'origine et l'authenticité du matériel pendant toute sa vie. Elle est immuable et non révocable unitairement sans retour atelier.
- **Identité opérationnelle (type LDevID) :** enrôlée au déploiement à partir de l'identité de fabrication, elle porte les droits applicatifs de l'équipement chez son exploitant. Elle est renouvelable et révocable sans toucher à l'identité de fabrication.

Cette séparation permet le transfert de propriété (l'exploitant ré-enrôle une identité opérationnelle sans dépendre des secrets d'usine) et la révocation ciblée : un lot suspect est révoqué au niveau de l'autorité de production, sans invalider le reste de la flotte. L'autorité de certification d'usine doit être maintenue hors ligne, ses cérémonies documentées, et sa compromission traitée comme un scénario de crise avec plan de ré-émission [51, 4].

Lorsque le composant le permet, un secret issu d'un PUF peut ancrer l'identité de fabrication sans stockage de clef en mémoire non volatile, ce qui réduit la surface d'extraction physique [38].

7.4 Spécifications de sécurité pour l'implémentation

Générer les secrets au plus près de leur lieu d'usage. La génération dans le SE cible (*on-die*) élimine le transport du secret ; à défaut, la génération dans le HSM d'usine avec injection par canal sécurisé (SCP03 ou équivalent) est acceptable, l'injection en clair ne l'est jamais [25, 49].

Établir un canal sécurisé de bout en bout. Le canal entre HSM d'usine et composant cible doit assurer confidentialité, intégrité, anti-rejeu et authentification mutuelle, y compris au travers des équipements de test intermédiaires, qui doivent être traités comme non fiables [25, 61].

Rendre la production comptable. Chaque émission d'identité décrémente un quota signé ; les écarts entre quotas consommés, unités déclarées et unités testées déclenchent une investigation. Ce rapprochement est la principale défense contre la sur-production [4].

Vérifier l'état de sortie d'usine. Un autotest final doit prouver, enregistrement signé à l'appui : débogage verrouillé, firmware de production authentique, identités correctement installées, mode transport actif. Toute unité en échec est mise en quarantaine tracée [47, 3].

7.5 Checklist industrielle et suite de tests synthétique

7.5.1 Checklist des actions côté industriel

ID	Action à réaliser	Responsable	Preuve attendue
PROV-CL-01	Cartographier sites, rôles et transferts de responsabilité de la chaîne de personnalisation	RSSI + industrialisation	Cartographie validée + contrats sous-traitants
PROV-CL-02	Mettre en place le HSM d'usine, les cérémonies de clefs et la politique d'export chiffré	Responsable PKI/cryptographie	Procès-verbaux de cérémonie, politique signée
PROV-CL-03	Déployer la diversification par équipement et la double identité fabrication/opérationnelle	Firmware sécurité + PKI	Dossier de conception + certificats de test
PROV-CL-04	Activer quotas signés, comptage de production et rapprochement périodique	Production + qualité	Rapports de rapprochement archivés
PROV-CL-05	Formaliser le verrouillage de fin de ligne et l'autotest de sortie d'usine	Hardware + firmware	Procédure de fin de ligne + journaux signés

TABLE 7.2 – Checklist industrielle pour "Provisionner les secrets et les identités en production"

7.5.2 Suite de tests synthétique

ID	Test	Méthode	Criticité	Critère de réussite
PROV-T-01	Tentative d'injection de clef en clair sur la ligne	Test négatif en banc	Haute	Injection refusée, secret jamais observable en clair
PROV-T-02	Émission d'identité au-delà du quota autorisé	Test de politique HSM	Haute	Émission bloquée, alerte de dépassement émise
PROV-T-03	Rejeu d'une session de personnalisation capturée	Test protocolaire	Haute	Session refusée (anti-rejeu du canal usine)
PROV-T-04	Contrôle de l'état de sortie d'usine sur échantillon	Audit banc + lecture configuration	Haute	Débogage verrouillé, mode transport actif sur 100% de l'échantillon
PROV-T-05	Rapprochement quotas/production/expéditions sur un lot	Revue comptable sécurité	Moyenne	Écart nul ou justifié par quarantaine tracée

TABLE 7.3 – Suite de tests synthétique pour "Provisionner les secrets et les identités en production"

8

Mettre à jour le firmware en exploitation (OTA/DFU)

8.1 Analyse de la menace et des risques

La capacité de mise à jour est à la fois une exigence réglementaire (correction des vulnérabilités pendant toute la période de support [24]) et l'une des surfaces d'attaque les plus rentables pour un attaquant : un mécanisme de mise à jour compromis fournit une exécution de code persistante, légitime en apparence et déployable à l'échelle de la flotte [47, 4].

Les menaces dominantes sont les suivantes :

- diffusion d'une image malveillante via une chaîne CI/CD ou un serveur de mise à jour compromis ;
- retour forcé (rollback) vers une version signée mais vulnérable ;
- désynchronisation entre vérification et installation (TOCTOU) : l'image contrôlée n'est pas celle réellement écrite ou exécutée [7] ;
- interruption volontaire ou accidentelle (coupure d'alimentation) laissant l'équipement dans un état incohérent ou inamorçable ;
- interception ou altération du flux de mise à jour (MITM, image tronquée, manifeste rejoué) ;
- abus du mode recovery ou du DFU local pour charger du code arbitraire.

8.1.1 Traçabilité des attaques, attaquants et contrôles

Scénario d'attaque	Menace de référence	Profils AM	Objectifs OS	Contrôles associés
Image malveillante diffusée via l'infrastructure de mise à jour	Compromission du boot et du firmware	AM-04	OS-02, OS-05	Manifeste signé, séparation des rôles de signature, traçabilité build
Rollback vers une version signée vulnérable	Compromission du boot et du firmware	AM-01, AM-04	OS-02, OS-03	Compteur monotone anti-rollback, version minimale
Altération entre vérification et exécution (TOCTOU)	Altération d'opérations critiques	AM-03	OS-02, OS-04	Vérification à l'installation et à chaque boot, écriture atomique
Rejeu d'un manifeste ancien (freeze attack)	Altération d'opérations critiques	AM-01, AM-04	OS-02, OS-05	Expiration des manifestes, horodatage vérifié
Abus du mode recovery ou du DFU local	Abus de maintenance et de débogage	AM-01, AM-02	OS-03, OS-05	Recovery signé minimal, authentification de session DFU, journalisation

TABLE 8.1 – Traçabilité des attaques du chapitre "Mettre à jour le firmware en exploitation (OTA/DFU)" vers menaces, profils AM et objectifs OS

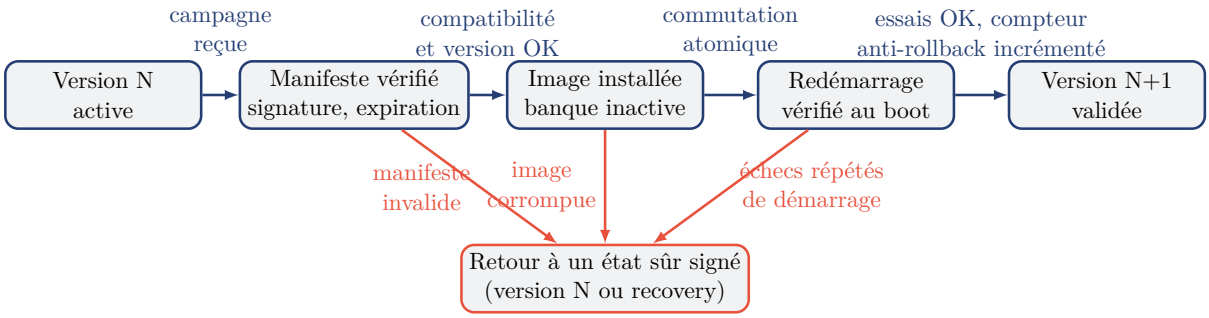


FIGURE 8.1 – Automate d'état de la mise à jour pour le cas d'usage "Mettre à jour le firmware en exploitation (OTA/DFU)"

8.2 Architecture de mise à jour recommandée

Descriptif synthétique

L'automate relie la campagne OTA à la chaîne de boot : le manifeste est vérifié (signature, compatibilité, expiration) avant installation dans la banque inactive, la commutation est atomique, puis la nouvelle image doit être revalidée au démarrage avant que la version ne soit entérinée et le compteur anti-rollback incrémenté. Toute anomalie ramène l'équipement sur un état sûr signé (banque antérieure ou recovery), sans état intermédiaire inamorçable.

8.2.1 Manifeste signé et métadonnées

Toute mise à jour est décrite par un manifeste signé contenant au minimum : la version, les condensats des images, la cible matérielle et ses contraintes de compatibilité, et une date d'expiration. L'expiration protège contre les attaques par gel (*freeze attack*), où un attaquant rejoue indéfiniment un manifeste ancien pour empêcher la flotte de se corriger. La séparation des rôles de signature (clefs de release distinctes des clefs de développement, seuils multi-signatures pour les opérations sensibles) suit le modèle établi par TUF et sa déclinaison embarquée Uptane [64, 68].

8.2.2 Partitionnement A/B et recovery

L'installation s'effectue dans une banque inactive pendant que la banque active reste intacte ; la commutation est atomique et réversible. Un compteur d'essais de démarrage borne les tentatives sur la nouvelle image : en cas d'échecs répétés, l'équipement revient automatiquement sur la banque antérieure, puis, en dernier recours, sur un recovery signé minimal. Aucun état intermédiaire ne doit être inamorçable, et aucun chemin de récupération ne doit accepter de code non signé [47].

8.2.3 Anti-rollback et cohérence avec le secure boot

Le mécanisme de mise à jour installe, mais ne décide pas : l'autorité d'exécution reste la chaîne de boot décrite au chapitre "Authentifier le logiciel du processeur généraliste (GPP) au boot". L'image est donc vérifiée deux fois, à l'installation puis à chaque démarrage, ce qui ferme les scénarios TOCTOU entre les deux étapes [7, 47]. La version minimale acceptable est portée par un compteur monotone matériel, incrémenté de façon irréversible lors de la validation d'une mise à jour de sécurité.

8.2.4 Protection du canal et du flux DFU local

Le canal OTA repose sur une session mutuellement authentifiée (TLS mutuel ou équivalent) [61] ; l'authenticité de l'image reste toutefois portée par le manifeste, jamais par le seul canal, afin que la sécurité ne dépende pas des infrastructures intermédiaires (CDN, relais). Le DFU local (UART, USB, JTAG de maintenance) doit exiger une authentification d'opérateur et une session anti-rejeu ; l'approche WooKey, où un token actif contraint le flux DFU, illustre le durcissement atteignable lorsque la menace locale est élevée [7, 19].

8.3 Diffusion en flotte et gouvernance

La diffusion doit être progressive : une population canari, puis des vagues croissantes, avec une télémétrie de taux de succès et des critères d'arrêt automatique de campagne. Pour les environnements industriels, les fenêtres de maintenance et les contraintes de disponibilité imposent une planification concertée avec l'exploitant [27]. Le CRA impose par ailleurs de distinguer les mises à jour de sécurité des évolutions fonctionnelles et d'en informer l'utilisateur [24].

Deux scénarios de crise doivent être préparés à froid : l'échec de masse d'une campagne (plan de retour arrière et de reprise) et la compromission d'une clef de signature, qui renvoie au plan de ré-émission à l'échelle de la flotte défini pour le secure boot (action GPP-CL-05).

8.4 Spécifications de sécurité pour l'implémentation

Vérifier avant d'installer et avant d'exécuter. La vérification unique à la réception est insuffisante : signature et compatibilité sont contrôlées à l'installation, puis l'intégrité est revérifiée par la chaîne de boot à chaque démarrage [47].

Ne jamais dégrader l'existant. L'image active reste intacte et amorçable jusqu'à la validation complète de la nouvelle image ; l'écriture est tolérante aux coupures (journalisation, reprise sur interruption) et testée comme telle.

Chiffrer l'image lorsque la confidentialité l'exige. L'authenticité et l'intégrité sont obligatoires dans tous les cas ; le chiffrement s'ajoute lorsque l'image contient de la propriété intellectuelle ou des paramètres sensibles, avec un mode authentifié [41].

Protéger l'infrastructure de signature. Les clefs de release résident en HSM, les signatures sont journalisées et rattachées à un build reproductible identifié (SBOM, registre signé), et la rotation des clefs est exercée régulièrement [52, 51].

8.5 Checklist industrielle et suite de tests synthétique

8.5.1 Checklist des actions côté industriel

ID	Action à réaliser	Responsable	Preuve attendue
OTA-CL-01	Spécifier le manifeste signé (champs, expiration, rôles et seuils de signature)	Architecture sécurité	Spécification versionnée
OTA-CL-02	Implémenter le partitionnement A/B avec commutation atomique et recovery signé	Firmware	Dossier de conception + tests d'intégration
OTA-CL-03	Déployer l'anti-rollback matériel cohérent avec la politique secure boot	Firmware sécurité	Rapport de tests croisés avec GPP-T
OTA-CL-04	Mettre en place la diffusion progressive (canari, vagues, critères d'arrêt automatique)	DevSecOps + exploitation	Procédure de campagne validée
OTA-CL-05	Préparer les plans de crise : échec de masse et compromission de clef de signature	RSSI + opérations	Plans testés en exercice

TABLE 8.2 – Checklist industrielle pour "Mettre à jour le firmware en exploitation (OTA/DFU)"

8.5.2 Suite de tests synthétique

ID	Test	Méthode	Criticité	Critère de réussite
OTA-T-01	Installation d'une image au manifeste expiré ou rejoué	Test protocolaire	Haute	Rejet systématique + événement d'audit
OTA-T-02	Coupure d'alimentation à chaque étape de l'installation	Test de robustesse instrumenté	Haute	Redémarrage systématique sur une image valide
OTA-T-03	Tentative de rollback vers une version signée vulnérable	Test d'intégration boot	Haute	Refus (compteur monotone violé)
OTA-T-04	Altération de l'image entre vérification et écriture (TOCTOU)	Test négatif instrumenté	Haute	Détection au boot, aucune exécution du code altéré
OTA-T-05	Campagne canari avec taux d'échec simulé au-delà du seuil	Test de gouvernance de flotte	Moyenne	Arrêt automatique de la vague + alerte exploitation

TABLE 8.3 – Suite de tests synthétique pour "Mettre à jour le firmware en exploitation (OTA/DFU)"

9

Attester, superviser et retirer les équipements en exploitation

9.1 Analyse de la menace et des risques

Une fois déployée, la flotte vit pendant des années dans des environnements non maîtrisés. Les mécanismes des chapitres précédents (boot mesuré, clefs protégées, mise à jour signée) ne produisent de la valeur en exploitation que s'ils sont observés : un équipement compromis qui n'est jamais interrogé reste compromis en silence. La maintenance en condition opérationnelle (MCO) de sécurité couvre donc l'attestation périodique, la supervision, la réponse à incident et le retrait maîtrisé des équipements [53, 31, 24].

Les menaces dominantes sont les suivantes :

- persistance silencieuse : un équipement compromis continue d'opérer sans jamais être détecté, faute d'attestation ou de télémétrie exploitée ;
- fausses preuves d'attestation : rejeu d'une preuve ancienne, équipement émulé, ou preuve produite par un composant compromis ;
- dérive des mesures de référence : des golden values corrompues ou non gouvernées font accepter un état malveillant, ou saturent la supervision de faux positifs ;
- compromission de l'infrastructure de vérification (serveur d'attestation, collecteur de journaux), qui aveugle la flotte entière ;
- fuite de secrets en fin de vie : équipements réformés, revendus ou retournés en SAV avec secrets et données intacts ;
- révocation défaillante : un équipement retiré ou volé conserve des identités et des accès valides.

9.1.1 Traçabilité des attaques, attaquants et contrôles

9.2 Architecture d'attestation distante

Scénario d'attaque	Menace de référence	Profils AM	Objectifs OS	Contrôles associés
Rejeu ou émulation d'une preuve d'attestation	Usurpation d'identité de plateforme	AM-03, AM-04	OS-02, OS-05	Nonce de fraîcheur, clef d'attestation liée à l'identité matérielle
Compromission persistante non détectée sur le terrain	Compromission du boot et du firmware	AM-04	OS-02, OS-05	Attestation périodique, télémétrie sécurité, corrélation flotte
Corruption des golden values ou du vérificateur	Altération d'opérations critiques	AM-02, AM-04	OS-02, OS-03	Golden values signées et versionnées, vérificateur durci et audité
Extraction de secrets sur équipement réformé ou en SAV	Extraction de secrets et d'éléments sensibles	AM-01, AM-03	OS-01, OS-03	Effacement cryptographique, quarantaine atelier, révocation
Usage d'identités valides par un équipement retiré ou volé	Usurpation d'identité de plateforme	AM-01, AM-04	OS-03, OS-05	Révocation des identités opérationnelles, inventaire de flotte à jour

TABLE 9.1 – Traçabilité des attaques du chapitre "Attester, superviser et retirer les équipements en exploitation" vers menaces, profils AM et objectifs OS

9.2.1 Modèle et preuves

L'architecture suit le modèle RATS : l'équipement (*attester*) produit une preuve signée de son état, un vérificateur la confronte aux valeurs de référence, et les services métier (*relying parties*) consomment le verdict, jamais la preuve brute [63]. Sur plateforme TPM, la preuve est une quote des PCR couvrant la chaîne de boot mesurée du chapitre GPP [30, 43] ; sur architecture SE/TEE, une preuve équivalente peut être construite à partir des mesures consolidées par la racine de confiance.

Deux propriétés sont non négociables : la **fraîcheur** (nonce du vérificateur inclus dans la preuve, pour empêcher le rejeu) et la **liaison d'identité** (la clef d'attestation se rattache cryptographiquement à l'identité de fabrication du chapitre provisioning, pour empêcher l'émulation d'un équipement par un autre) [63, 28].

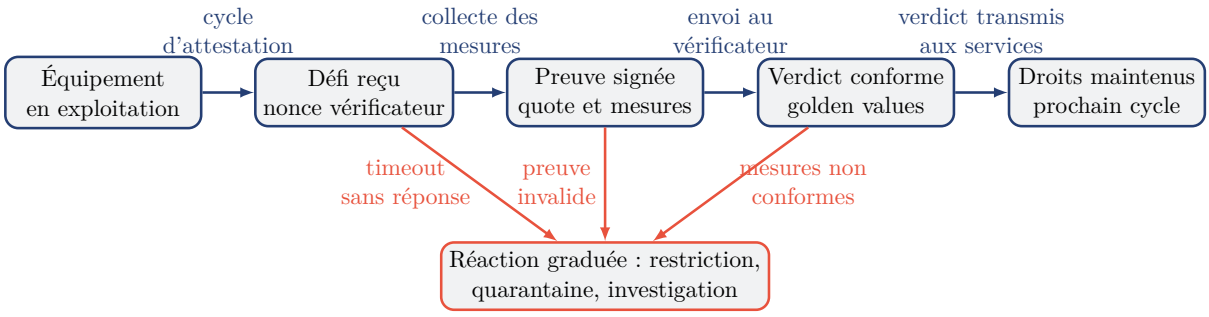


FIGURE 9.1 – Automate d'état du cycle d'attestation pour le cas d'usage "Attester, superviser et retirer les équipements en exploitation"

Descriptif synthétique

L'automate décrit le cycle d'attestation périodique : le vérificateur émet un défi avec nonce, l'équipement produit une preuve signée liée à son identité matérielle, et le verdict est rendu par comparaison aux golden values versionnées. Une preuve absente, invalide ou rejouée, comme un état non conforme, déclenche la réaction graduée (restriction de droits, quarantaine, investigation) plutôt qu'un rejet silencieux.

9.2.2 Gouvernance des valeurs de référence

Les golden values sont un actif de sécurité à part entière : signées, versionnées, produites par la chaîne CI/CD au moment du build (chaque release publie ses mesures de référence dans le registre signé), et déployées vers le vérificateur de façon synchronisée avec les campagnes OTA [43, 52]. Une mise à jour de firmware sans mise à jour des références produit de faux positifs massifs ; l'inverse, des références mises à jour sans contrôle, offre à un attaquant de la chaîne de build un moyen de faire accepter un état malveillant.

9.2.3 Politique de réaction

Un verdict de non-conformité doit déclencher une réaction graduée et prédéfinie : restriction des droits applicatifs (retour aux niveaux N0/N1 du chapitre transactions), quarantaine réseau, puis investigation. La politique doit être écrite avant le premier incident, en précisant qui décide, sur quels critères, et comment un équipement réhabilité réintègre la flotte (ré-attestation complète après reprovisioning contrôlé).

9.3 Supervision et réponse à incident

9.3.1 Télémétrie de sécurité minimale

La télémétrie remontée par équipement doit couvrir au minimum : résultats et échecs de boot, décisions `ALLOW/DENY` des opérations cryptographiques, verdicts d'attestation, valeur des compteurs anti-rollback, et événements de mise à jour (début, succès, retour arrière). Ces événements réutilisent le format d'audit défini au chapitre transactions (horodatage fiable, identité équipement, `reason_code`, chaînage) et sont exportés vers le collecteur central [31, 27].

9.3.2 Corrélation à l'échelle de la flotte

L'analyse unitaire ne suffit pas : les attaques de masse se détectent par corrélation (pic de refus anti-rejeu sur un lot, échecs d'attestation groupés après une campagne OTA, équipements d'un même site qui cessent de remonter). Le SOC doit disposer des référentiels croisés lot/site/version pour mener ces corrélations, ce qui suppose que les registres de production du chapitre provisioning soient accessibles en lecture à la supervision [53, 4].

9.3.3 Réponse à incident

Trois playbooks doivent exister et être exercés : équipement isolé compromis (quarantaine, collecte forensique, reprovisioning ou retrait), lot suspect (croisement avec les registres de production, révocation de lot), et compromission d'une clef d'infrastructure (bascule vers le

plan de ré-émission flotte défini en GPP-CL-05 et OTA-CL-05). Le CRA impose par ailleurs la notification des vulnérabilités activement exploitées dans des délais courts, ce qui suppose une chaîne de qualification interne déjà en place [24, 53].

9.4 Fin de vie, retour SAV et transfert de propriété

Le retrait d'un équipement est une opération de sécurité, pas une opération logistique :

- **Effacement des secrets :** l'effacement cryptographique (destruction de la clef de chiffrement dans le SE/HSE) est la méthode de référence pour les données chiffrées au repos ; les secrets résiduels en mémoire non chiffrée suivent les recommandations de sanitisation des supports [44] ;
- **Révocation :** les identités opérationnelles (LDevID) sont révoquées à la sortie d'inventaire ; l'identité de fabrication est conservée pour la traçabilité (RMA, litiges, analyse post-mortem) [28] ;
- **Retour SAV :** un équipement revenant du terrain est traité comme non fiable : quarantaine atelier, diagnostic sur banc isolé, et re-personnalisation complète avant toute réinjection en flotte [4] ;
- **Transfert de propriété :** le nouvel exploitant ré-enrôle une identité opérationnelle propre après purge des données et des politiques de l'ancien ; le processus s'appuie sur la double identité du chapitre provisioning, sans exposer les secrets d'usine.

Ces exigences doivent être définies dès l'architecture : un effacement cryptographique n'est possible que si le chiffrement au repos a été conçu pour, et une révocation de masse n'est réaliste que si l'inventaire de flotte est fiable [24, 3].

9.5 Checklist industrielle et suite de tests synthétique

9.5.1 Checklist des actions côté industriel

9.5.2 Suite de tests synthétique

ID	Action à réaliser	Responsable	Preuve attendue
MCO-CL-01	Déployer le vérificateur d'attestation et la politique de réaction graduée	Architecture sécurité + exploitation	Dossier d'architecture + politique validée
MCO-CL-02	Intégrer la publication des golden values signées dans la chaîne CI/CD, synchronisée avec l'OTA	DevSecOps	Registre de références versionné
MCO-CL-03	Définir la télémétrie sécurité minimale et son export vers le SOC	Exploitation + SOC	Schéma de télémétrie + preuve de collecte
MCO-CL-04	Rédiger et exercer les trois playbooks (équipement, lot, clef d'infrastructure)	RSSI + SOC/CSIRT	Comptes rendus d'exercices datés
MCO-CL-05	Formaliser les procédures de fin de vie, retour SAV et transfert de propriété	Qualité + exploitation	Procédures approuvées + traçabilité inventaire

TABLE 9.2 – Checklist industrielle pour "Attester, superviser et retirer les équipements en exploitation"

ID	Test	Méthode	Criticité	Critère de réussite
MCO-T-01	Rejeu d'une preuve d'attestation ancienne	Test protocolaire	Haute	Preuve rejetée (nonce invalide) + audit
MCO-T-02	Attestation d'un équipement au firmware modifié	Test d'intégration bout en bout	Haute	Verdict non conforme, réaction graduée appliquée
MCO-T-03	Campagne OTA sans mise à jour des golden values	Test négatif de gouvernance	Moyenne	Divergence détectée avant diffusion, campagne bloquée
MCO-T-04	Lecture des données d'un équipement après effacement de fin de vie	Analyse de support en atelier	Haute	Aucune donnée ni secret exploitable récupéré
MCO-T-05	Connexion d'un équipement aux identités révoquées	Test d'infrastructure	Haute	Accès refusé sur tous les services + alerte

TABLE 9.3 – Suite de tests synthétique pour "Attester, superviser et retirer les équipements en exploitation"

10

Exécution d'algorithmes cryptographiques sur SE, TPM et TEE

Dans un système embarqué, l'exécution d'une primitive cryptographique ne se résume pas à un choix de bibliothèque ou de fonction. Elle traduit en pratique la frontière entre le code applicatif exposé à des attaques logicielles et les secrets qui doivent rester maîtrisés, attestés et, si possible, isolés dans un environnement distinct. La décision d'exécuter une opération sur un Secure Element (SE), un TPM ou un TEE dépend donc du modèle de menace, de la criticité des clefs, du besoin d'attestation et des contraintes de performance et de coût [11, 51, 10].

En pratique, trois motifs justifient souvent ce déplacement : protéger des clefs racines ou des secrets critiques, obtenir une attestation fiable de l'état du dispositif, et réduire l'exposition du cœur de calcul principal aux attaques locales ou aux fuites de mémoire. Cette séparation est particulièrement pertinente pour les produits industriels, les équipements de santé ou les dispositifs connectés à longue durée de vie, où la robustesse du chain-of-trust et la capacité de mise à jour sécurisée sont aussi importantes que la simple exécution d'un algorithme [27, 31, 24].

10.1 Choisir entre SE, TPM et TEE

Le choix d'une architecture de confiance doit être guidé par le niveau d'isolation requis et par la nature de l'opération à exécuter. Un SE est souvent retenu pour la protection de clefs racines et les opérations sensibles à forte valeur, un TPM pour l'attestation, le stockage de mesures et la gouvernance des politiques d'usage, tandis qu'un TEE est adapté lorsqu'on souhaite exécuter un sous-ensemble de code sécurisé avec un bon compromis entre isolation logicielle et performance [25, 30, 9].

Dans une conception réaliste, on ne place pas toutes les opérations cryptographiques dans le même composant. Les clefs racines et les fonctions de signature/attestation sont souvent déléguées à un composant distinct, alors que le microcontrôleur hôte conserve des clefs de session, des handles ou des éléments de politique plus légers. Ce découpage améliore la résilience face à la compromission du cœur applicatif et simplifie la mise à jour de la pile cryptographique sans re-concevoir tout le produit [10, 57].

10.2 Accélération matérielle vs logicielle

L'offload vers un composant sécurisé réduit en général l'exposition des clefs et la charge CPU, mais ajoute une latence d'IPC/bus. Le bon choix dépend du profil (débit, énergie, criticité des secrets, coût BOM) [11, 51].

10.2.1 Benchmarks performance/consommation : SE intégré vs librairie MCU

Comparer au minimum :

- temps moyen et performance de signature, authentification et chiffrement
- énergie par opération
- coût mémoire (RAM/Flash)
- impact sur les tâches temps réel

Ces mesures doivent être interprétées à la lumière du contexte d'intégration : un composant sécurisé peut coûter plus en latence et en consommation d'énergie par transaction, alors même qu'il apporte une meilleure protection des secrets et une meilleure capacité d'attestation. Pour cette raison, les campagnes de benchmark doivent être réalisées en conditions représentatives, avec des charges réalistes et des scénarios de reprise, afin de ne pas favoriser à tort une solution trop optimiste sur papier [52, 11].

10.2.2 Critères de décision pour le placement des opérations

Un choix pertinent repose souvent sur quatre questions simples : la clef doit-elle être protégée contre l'extraction depuis le cœur applicatif ? faut-il une attestation de l'état du dispositif ou simplement une exécution rapide ? l'opération est-elle critique en termes de sécurité, ou peut-on tolérer une faiblesse locale ? enfin, le système doit-il permettre des mises à jour de firmware ou de bibliothèques cryptographiques sans re-conception complète ? Ces questions orientent vers un SE lorsque la priorité est la protection matérielle, vers un TPM lorsque l'attestation et la politique sont centrales, et vers un TEE lorsqu'un compromis plus flexible est recherché [25, 30, 9, 10].

10.2.3 Impact sur autonomie batterie

Sur équipement contraint, le rapport énergie/sécurité doit être mesuré : une clef protégée mais inaccessible en régime veille peut pénaliser la QoS. L'architecture doit donc séparer opérations fréquentes et opérations racines [3, 11].

10.2.4 Latence introduite par les échanges inter-puces

L'impact des bus (I2C/SPI) sur la latence transactionnelle doit être quantifié, notamment en charge et en cas de pertes/retransmissions. Les timeouts et retries doivent être bornés pour éviter le deadlock sécuritaire [25, 53].

10.3 Opérations supportées nativement

10.3.1 Chiffrement symétrique

AES-GCM est à privilégier pour la protection combinée confidentialité/intégrité ; les modes non authentifiés sont à restreindre aux cas legacy avec encapsulation adéquate. En pratique, le choix d'un mode authentifié n'est pas seulement une question de conformité : il évite les erreurs de conception classiques où une intégrité manquante ouvre la voie à des attaques par manipulation de flux ou de messages [41, 50]. Cette recommandation s'applique aussi bien à des échanges de commande industriels qu'à la protection locale de données sensibles au bord du dispositif.

L'emploi d'AES-GCM impose toutefois une contrainte absolue : l'unicité du couple clef/IV. La réutilisation d'un nonce détruit à la fois la confidentialité et l'authenticité du mode, et le risque est réel en embarqué, où un compteur d'IV peut se réinitialiser après une coupure d'alimentation. Conformément aux recommandations de l'ANSSI (RGS, annexe B1), la génération des IV doit être soit déterministe et persistante (compteur monotone sauvegardé), soit aléatoire avec un volume de chiffrement borné par clef [1, 41].

10.3.2 Chiffrement asymétrique

ECC est couramment retenu en embarqué pour limiter la charge calculatoire à niveau de sécurité équivalent selon les tailles de clef recommandées. Les tailles de clefs doivent suivre les recommandations de transition, car le passage à des niveaux de sécurité plus élevés influe à la fois sur la taille des messages, la consommation et la durée de vie des mécanismes de signature ou d'échange de clef [51, 1]. Dans un contexte d'embarqué, ce choix doit être aligné avec la politique de cryptoagilité du produit et avec les contraintes de mise à jour du firmware.

10.3.3 Fonctions de hachage

SHA-256/384 restent les références pratiques ; SHA-1 est proscrit pour les usages de sécurité. La sélection dépend des contraintes de performance et de conformité, mais aussi de la robustesse attendue vis-à-vis de l'intégrité des données, de la signature de messages et de l'authentification de firmware [45, 50]. Sur un microcontrôleur contraint, il est fréquent de privilégier une fonction de hachage plus légère en microarchitecture, à condition de conserver un niveau de sécurité conforme au profil de risque du dispositif.

10.3.4 Génération de nombres aléatoires

La qualité RNG conditionne la sécurité cryptographique : un RNG biaisé peut compromettre l'ensemble des mécanismes. En pratique, cette étape est souvent sous-estimée alors qu'elle conditionne la sécurité de toutes les opérations suivantes, de la génération de clef à la protection des nonces et des protocoles d'authentification. Il est donc nécessaire d'exiger une validation statistique et une conception documentée, typiquement fondée sur un DRBG associé à une source d'entropie qualifiée [46, 48, 18].

10.4 Protéger les données au repos

La protection des données au repos est une exigence transverse : la checklist du chapitre 3 refuse la mise en production sans mécanisme d'intégrité, et l'effacement cryptographique du chapitre 9 n'est possible que si le chiffrement au repos a été conçu pour. Le retour d'expérience WooKey rappelle le piège classique : un chiffrement de volume complet protège la confidentialité

mais pas l'intégrité, et une donnée chiffrée peut être altérée ou rejouée hors ligne sans détection [7].

10.4.1 Architecture de chiffrement recommandée

- **Chiffrement authentifié par défaut :** AEAD (AES-GCM ou équivalent) pour les données applicatives ; si un mode non authentifié est imposé (FDE legacy), l'intégrité doit être portée par une couche supérieure (condensés applicatifs, journal signé) et l'écart tracé dans l'analyse de risque [41, 7] ;
- **Hiérarchie de clefs :** clef maître de stockage non exportable en SE/HSE, clefs de volume ou de fichier dérivées par KDF et protégées par enveloppement (*key wrapping*) ; la rotation de la clef d'enveloppement se fait alors par ré-enveloppement, sans re-chiffrement complet du support ; la compromission avérée d'une clef de données impose en revanche le re-chiffrement des données concernées [51, 49] ;
- **Liaison à l'état de plateforme :** le déverrouillage des clefs de stockage n'est autorisé qu'après un boot valide (scellement TPM sur PCR, ou politique d'usage du SE), afin qu'un firmware compromis ne puisse pas monter les volumes [30, 43] ;
- **Anti-rejeu des données :** un compteur monotone ou une référence de version protège contre la restauration hors ligne d'un état ancien du stockage (rollback de données), symétrique de l'anti-rollback du firmware.

10.4.2 Effacement et fin de vie

Si tout le stockage sensible est chiffré dès la conception, la destruction de la clef maître dans le SE/HSE vaut effacement des données (*crypto-erase*), ce qui rend le retrait réaliste à l'échelle d'une flotte ; les supports contenant des résidus non chiffrés relèvent de la sanitisation classique [44]. Les procédures opérationnelles associées sont définies au chapitre 9.

10.4.3 Règles d'acceptation

- lecture directe du support hors ligne ⇒ aucune donnée exploitable ;
- modification hors ligne d'un bloc chiffré ⇒ détection au montage ou à la lecture, refus tracé ;
- restauration d'une image ancienne du stockage ⇒ refus (référence de version violée) ;
- boot non conforme ⇒ clefs de stockage non déverrouillées.

10.5 APIs et interfaces de programmation

10.5.1 GlobalPlatform : commande APDU pour Java Card SE

Les APDU et SCP03 structurent un modèle portable et auditable pour des opérations sécurisées sur SE [25, 59].

10.5.2 TPM 2.0 Library

Les piles TPM offrent des commandes standard (quote, seal, unseal, policy) facilitant l'interopérabilité avec des infrastructures d'attestation [30, 66].

10.5.3 PSA Crypto API (ARM)

PSA facilite la portabilité multi-fournisseur et l'agilité algorithmique en séparant API logique et implémentation matérielle [10, 11].

10.5.4 TrustZone-M : appels sécurisés vers TEE Cortex-M

TrustZone-M permet de partitionner proprement code sensible et non sensible, sous réserve d'un cloisonnement strict des interruptions, mémoires et périphériques [9, 67].

10.6 Patterns d'intégration logicielle

10.6.1 Wrapper abstraction layer

Introduire une couche d'abstraction cryptographique simplifie la substitution SE/TPM/TEE et réduit la dette technique lors de migration post-quantique [57, 10].

10.6.2 Gestion des erreurs et timeouts

Les erreurs de composant sécurisé ne doivent jamais dégrader silencieusement vers un mode non sécurisé. Définir des politiques fail-secure explicites [53, 27].

10.6.3 Logging et audit des opérations cryptographiques sensibles

Journaliser identité de clef, opération, résultat, contexte et référence de firmware. Cette traçabilité est essentielle pour l'investigation et la conformité CRA [24, 31].

10.7 Synthèse pratique

Au-delà des listes d'algorithmes et d'APIs, la bonne architecture consiste à faire coexister un modèle de sécurité clair, des interfaces standardisées et une capacité de mesure. Un système robuste ne se contente pas de choisir AES-GCM ou ECC : il définit où les secrets sont protégés, comment on atteste l'état du dispositif, comment on gère les timeouts et comment on rend les opérations auditables. C'est ce niveau de cohérence qui transforme une implémentation cryptographique en composant de confiance durable, au sens des référentiels industriels et des attentes réglementaires actuelles [11, 25, 30, 9].

10.8 Checklist industrielle et suite de tests synthétique

10.8.1 Checklist des actions côté industriel

10.8.2 Suite de tests synthétique

ID	Action à réaliser	Responsable	Preuve attendue
CRYPTO-CL-01	Inventorier les primitives utilisées et proscrire les algorithmes obsolètes (politique cryptographique versionnée)	Architecture sécurité	Politique cryptographique signée [50, 1]
CRYPTO-CL-02	Justifier le placement de chaque opération (SE/TPM/TEE/hôte) par l'analyse de menace	Architecture sécurité	Matrice opération/composant validée
CRYPTO-CL-03	Mettre en place la couche d'abstraction cryptographique (PSA ou équivalent)	Firmware	Dossier d'architecture + revue de code
CRYPTO-CL-04	Mesurer performance et énergie en conditions représentatives avant gel d'architecture	Firmware + hardware	Rapport de benchmark archivé
CRYPTO-CL-05	Qualifier la chaîne d'aléa (source d'entropie, DRBG, tests de santé)	Firmware sécurité	Dossier de validation RNG

Table 10.1 – Checklist industrielle pour "Exécution d'algorithmes cryptographiques sur SE, TPM et TEE"

ID	Test	Méthode	Criticité	Critère de réussite
CRYPTO-T-01	Autotests par vecteurs connus (KAT) sur toutes les primitives	Test automatisé à chaque build	Haute	100% des KAT conformes, échec bloquant
CRYPTO-T-02	Erreur ou timeout du composant sécurisé en opération	Injection de panne contrôlée	Haute	Fail-secure, aucun repli silencieux vers une implémentation non protégée
CRYPTO-T-03	Tentative d'export d'une clef marquée non exportable	Test de politique de clefs	Haute	Export refusé + événement d'audit émis
CRYPTO-T-04	Mesure de latence et d'énergie par opération vs budget alloué	Campagne de mesure en charge	Moyenne	Budgets tenus ou dérogation documentée
CRYPTO-T-05	Validation statistique de la sortie RNG et des tests de santé	Suite statistique (SP 800-90B)	Haute	Aucun biais détecté, tests de santé actifs en production [48]

Table 10.2 – Suite de tests synthétique pour "Exécution d'algorithmes cryptographiques sur SE, TPM et TEE"

A

Migration vers la cryptographie post-quantique

A.1 Menace quantique et urgence de migration

La sécurité de la cryptographie à clef publique aujourd'hui déployée (RSA, ECDSA, ECDH) repose sur la difficulté de la factorisation et du logarithme discret. L'algorithme de Shor, exécutable sur un ordinateur quantique à grande échelle (CRQC, *Cryptographically Relevant Quantum Computer*), résout ces deux problèmes efficacement et rendrait caduque l'ensemble de cette cryptographie asymétrique [62, 6, 57]. Le risque est déjà actuel via l'attaque *Harvest Now, Decrypt Later* (HNDL) : un adversaire peut capturer dès aujourd'hui des flux chiffrés pour les déchiffrer une fois un CRQC disponible [40, 6]. Pour les systèmes embarqués à longue durée de vie (industriel, santé, défense), dont la confidentialité des données doit être assurée au-delà de 5 à 10 ans, cette menace est immédiate et doit orienter les choix architecturaux dès la conception. La cryptographie symétrique est moins exposée : l'algorithme de Grover accélère seulement quadratiquement la recherche exhaustive, et son effet se contre en doublant la taille des clefs (AES-256 au lieu d'AES-128, SHA2-384 ou SHA3-384 pour les hachages) [6, 57].

A.2 Standards post-quantiques

La cryptographie post-quantique (PQC) désigne des algorithmes exécutables sur des ordinateurs classiques et dont la sécurité est conjecturalement maintenue face à un adversaire disposant d'un ordinateur quantique. Le NIST a finalisé en 2024 les premiers standards issus de son processus de standardisation [57] :

- **ML-KEM** (FIPS 203, ex-Kyber) : mécanisme d'encapsulation de clef à base de réseaux euclidiens modulaires, destiné à l'établissement de canal confidentiel [54]
- **ML-DSA** (FIPS 204, ex-Dilithium) : signature numérique à base de réseaux, profil général [55]

- **SLH-DSA** (FIPS 205, ex-SPHINCS+) : signature à base d'arbres de hachage, hypothèses mathématiques distinctes des réseaux et utilisable sans hybridation selon l'ANSSI

Ces trois algorithmes constituent la base concrète de migration identifiée par le NIST dans IR 8547 *Transition to Post-Quantum Cryptography Standards* [57], qui fixe également un calendrier d'obsolescence des algorithmes classiques (RSA, ECDSA, DH) à partir de 2030 et leur interdiction au plus tard en 2035.

A.3 Feuille de route ANSSI

Dans sa note de position [6], l'ANSSI définit trois phases de transition applicables aux produits certifiés en France, alignées avec les livrables NIST :

- **Phase 1** (situation actuelle jusqu'à la publication des premiers standards NIST) : hybridation facultative, la résistance post-quantique constitue une *défense en profondeur* optionnelle. Le mécanisme hybride doit garantir qu'aucune régression de sécurité pré-quantique n'est introduite. L'ANSSI recommande d'engager dès maintenant cette phase pour les produits dont la durée de vie s'étend au-delà de 2030
- **Phase 2** (à partir de 2026, avec standards NIST publiés) : hybridation obligatoire pour tout produit revendiquant une résistance quantique. L'ANSSI évaluera les algorithmes PQC retenus et pourra délivrer des visas de sécurité incluant une garantie de sécurité post-quantique
- **Phase 3** (probablement pas avant 2030) : PQC autonome possible sans hybridation, après accumulation suffisante de recul cryptanalytique

L'hybridation consiste à combiner un algorithme classique éprouvé et un algorithme PQC : pour l'établissement de clef, on dérive le secret final depuis les deux contributions via une KDF ; pour la signature, on exige la validité des deux signatures. Le surcoût en taille de message reste faible relativement au coût des primitives PQC elles-mêmes.

A.4 Cryptoagilité et implications pratiques pour l'embarqué

L'ANSSI et le NIST insistent tous deux sur la *cryptoagilité* : la capacité d'un produit à substituer ses primitives cryptographiques sans re-conception matérielle ni rappel en masse [6, 57]. Dans le contexte des systèmes embarqués traités dans ce guide, cela se traduit par :

- **Abstraction des primitives** : ne pas ancrer dans le code métier un algorithme ou une taille de clef spécifique ; exposer des profils cryptographiques versionnés et négociables (identifiant d'algorithme, taille, suite)
- **Mise à jour sécurisée du firmware** : la chaîne de boot sécurisé décrite au chapitre précédent est le vecteur naturel de mise à jour des bibliothèques cryptographiques ; elle doit être conçue pour supporter des remplacements de bibliothèques sans recertification complète du produit
- **Délégation vers un composant sécurisé upgradable** : sur MCU contraint, déléguer les opérations PQC à un SE ou TPM dont le firmware est signé et mettable à jour réduit le risque de re-conception matérielle

- **Priorisation HNDL** : traiter en priorité les données dont la confidentialité doit résister au-delà de l'horizon de 5 à 10 ans ; déployer des mécanismes hybrides sur ces flux avant toute autre considération d'optimisation

La veille normative NIST/ANSSI doit être organisée (cadence semestrielle recommandée) pour suivre l'évolution du calendrier d'obsolescence et les éventuels correctifs sur les algorithmes standardisés [57, 6, 54, 55].

B

Interfaçage physique et protocoles de communication

B.1 Choix du bus et surface d'attaque

Les bus typiquement utilisés pour relier un MCU hôte à un SE, TPM ou module TEE externe sont I2C, SPI, UART et SWP. I2C est simple et économique mais exposé au sniffing sur bus partagé ; SPI offre un meilleur débit et une prédictibilité utile pour TPM en activité soutenue ; UART, pratique en développement et provisionnement, doit être désactivé ou strictement authentifié en production ; SWP est propre aux contextes SIM/eUICC et s'inscrit dans des écosystèmes opérateur déjà cadrés [25, 30, 3]. Dans tous les cas, l'absence de protection logique sur le bus laisse l'interception physique accessible : l'authenticité et la confidentialité des échanges critiques doivent être assurées par des protections cryptographiques de couche applicative, indépendamment du bus retenu [61, 25].

B.2 Contraintes électriques et sécurité physique

Les niveaux logiques, timings et protections ESD ne sont pas sans incidence sécuritaire : des erreurs ou transitoires physiques peuvent ouvrir des fenêtres aux fault attacks (glitch tension, injection laser) ou contourner des vérifications de signature [12, 13]. Le canal de communication doit être défendu contre le sniffing, le rejeu, l'injection de trames et le MITM local par des nonces, compteurs monotones et binding de session sur l'identité matérielle du composant [61, 25, 30]. Les optimisations basse consommation (réduction de fréquence, wake-on-interrupt, power gating) sont compatibles avec la sécurité à condition de ne pas supprimer la ré-authentification au réveil [3, 27].

B.3 Débogage et validation

La validation d'interface doit inclure des tests d'abus : glitches, variations de tension, pertes de synchronisation, fuzzing de trames. L'objectif est de vérifier que le système échoue de façon sûre (*fail-secure*) et que les interfaces de débogage (JTAG, UART de débogage) sont désactivées ou protégées avant mise en production [47, 27].

C

Intégration système et considérations opérationnelles

C.1 Choix du microcontrôleur hôte et layout PCB

Le choix du MCU hôte doit équilibrer ressources disponibles, primitives de sécurité matérielle embarquées (TrustZone-M, flash OTP, gestionnaire de clefs), outillage certifiable et support long terme [9, 67, 52]. Chaque décision de layout doit être mise en regard d'une menace concrète : placer le SE au plus près du MCU réduit l'exposition des pistes au probing et à l'injection EMI ; filtrer l'alimentation du composant sécurisé atténue les glitches ; un blindage local augmente le coût des attaques par mesure de rayonnement ; des mesures anti-tamper mécaniques ou électriques élèvent le coût d'accès physique [34, 13, 17].

C.2 Chaîne de développement et réglementaire

Les pilotes et bibliothèques intégrés doivent être gérés dans un pipeline conforme au SSDF, avec revues de code, SBOM, contrôle de provenance et tests de régression sécurité [52, 24]. Le produit doit tracer explicitement ses obligations réglementaires : Cyber Resilience Act pour les produits connectés mis sur le marché UE [24], exigences radio-cyber via la delegated regulation RED [23], éventuelles exigences sectorielles (IEC 62443 pour les environnements OT, ISO 27001 pour le système de management), et stratégie d'évaluation selon le marché cible (Common Criteria, FIPS 140-3) [17, 49, 27].

C.3 Maintenance en condition opérationnelle (MCO)

La MCO sécurité doit couvrir la supervision des événements de sécurité, la rotation périodique des clefs, la mise à jour signée du firmware et un plan de réponse à incident activable à l'échelle de la flotte. Pour les déploiements IoT à grande échelle, les procédés de révocation de masse (certificats, clefs de provisionnement) doivent être conçus dès l'architecture initiale et non ajoutés a posteriori [53, 24, 31].

D

Stratégie de certification et traçabilité normative

Ce chapitre outille l'axe audit/conformité annoncé en introduction : choisir un schéma d'évaluation adapté au marché visé, tracer les objectifs de sécurité du guide vers les référentiels réglementaires et normatifs, puis constituer le dossier de preuves exploitable en audit.

D.1 Choisir un schéma d'évaluation

Le schéma d'évaluation doit être choisi en fonction du marché cible, du niveau d'assurance attendu et du budget, en distinguant l'évaluation du **composant** (SE, TPM) de celle du **produit** qui l'intègre. Un composant certifié ne certifie pas le produit : l'assurance se compose, elle ne se transfère pas.

Schéma	**Périmètre typique**	**Effort**	**Usage recommandé**
Common Criteria (EAL) [17]	Composant sécurisé (SE, carte à puce)	Élevé	Assurance forte sur le composant ; marchés paiement, identité, régalien
CSPN [2]	Produit fini, périmètre ciblé	Modéré (charge fixe)	Premier niveau de confiance en France ; évaluation en boîte noire à durée bornée
SESIP / EN 17927 [26]	Plateforme IoT (MCU + RoT + services)	Modéré	Réutilisation composant vers produit ; passerelles documentées vers EN 303 645 et CRA
PSA Certified [11]	Racine de confiance MCU (niveaux 1 à 3)	Faible à modéré	Assurance graduée sur la RoT des microcontrôleurs
FIPS 140-3 [49]	Module cryptographique	Élevé	Marchés nord-américains et exigences fédérales
ETSI EN 303 645 [21]	Objet connecté grand public	Faible	Socle minimal consumer ; base des exigences radio-cyber RED [23]

Table D.1 – Panorama des schémas d'évaluation applicables aux produits embarqués sécurisés

En pratique, la stratégie la plus économe consiste à composer : retenir un SE ou TPM déjà certifié (Common Criteria, FIPS 140-3) pour porter l'assurance de protection des clefs, puis évaluer le produit intégrateur à un niveau adapté (CSPN ou SESIP), l'évaluation produit se concentrant alors sur l'intégration, la chaîne de boot et les interfaces plutôt que sur la résistance

physique du composant [17, 2, 26]. Pour les secteurs réglementés, les schémas sectoriels s'ajoutent : IEC 62443 pour l'industriel [27], ISO/SAE 21434 pour l'automobile [32].

D.2 Traçabilité des objectifs de sécurité vers les référentiels

Le tableau suivant relie les objectifs de sécurité OS-01 à OS-05 du guide aux principales exigences réglementaires et normatives. Il constitue le point d'entrée du dossier de conformité : chaque contrôle mis en œuvre dans les chapitres techniques se rattache à un objectif OS, donc à une exigence externe.

Objectif	**CRA [24]**	**IEC 62443-4-2 [27]**	**ISO 27001 :2022 [31]**	**NIST CSF 2.0 [53]**
OS-01 Confidentialité des actifs critiques	Annexe I : protection de la confidentialité des données	CR 4.1 (confidentialité de l'information)	A.8.24 (usage de la cryptographie)	PR.DS
OS-02 Intégrité du système	Annexe I : protection de l'intégrité, sécurité des mises à jour	CR 3.4, EDR 3.14 (intégrité logicielle et du boot)	A.8.9, A.8.19 (configuration, installation)	PR.PS
OS-03 Contrôle d'accès aux fonctions sensibles	Annexe I : contrôle d'accès, configuration sécurisée par défaut	CR 1.1, CR 2.1 (authentification, autorisation)	A.5.15, A.8.2 (contrôle d'accès, privilèges)	PR.AA
OS-04 Résistance aux attaques physiques	Annexe I : réduction des surfaces d'attaque	EDR 3.11 (résistance au sabotage physique)	A.7 (contrôles physiques)	PR.IR
OS-05 Traçabilité et audit	Annexe I : journalisation ; gestion des vulnérabilités	CR 2.8 à 2.12 (événements auditables, horodatage)	A.8.15, A.8.16 (journalisation, supervision)	DE.CM

TABLE D.2 – Traçabilité des objectifs OS-01 à OS-05 vers les référentiels réglementaires et normatifs

Les références aux exigences sont volontairement données au niveau de la famille : la déclinaison exacte (articles du CRA, exigences détaillées IEC) dépend de la classe du produit et doit être établie dans la matrice de conformité du projet.

D.3 Constitution du dossier de preuves d'audit

Les checklists (`*-CL-*`) et suites de tests (`*-T-*`) des chapitres techniques sont conçues pour produire directement les preuves attendues en audit. Le dossier minimal comprend :

- l'analyse de risque et le modèle de menace, tenus à jour (EBIOS RM ou équivalent) [5] ;
- les politiques versionnées : politique cryptographique, politique de clefs, politique secure boot ;
- les rapports d'exécution des suites de tests CARTE-T, GPP-T, TRX-T, PROV-T et CRYPTO-T, avec leurs critères de réussite ;
- la SBOM et le registre de build signé de la chaîne CI/CD [52] ;

- les journaux probants d'exploitation et la procédure de gestion et de notification des vulnérabilités exigée par le CRA [24] ;
- pour les produits certifiés : les certificats des composants et les rapports d'évaluation, avec la démonstration que les hypothèses d'usage du composant sont respectées dans le produit.

La cohérence entre ces pièces importe autant que leur existence : un auditeur vérifiera qu'un contrôle déclaré dans la matrice de conformité correspond à une exigence implémentée, testée par une campagne identifiée, et supervisée en exploitation.

E

Références normatives et spécifications

- ISO/IEC 11889 (TPM 2.0) [30]
- ISO/IEC 15408 (Common Criteria) [17]
- ISO/IEC 27001 :2022 (management de la sécurité) [31]
- IEC 62443-4-2 (exigences techniques IACS) [27]
- NIST SP 800-193 (Platform Firmware Resiliency) [47]
- NIST SP 800-155 (BIOS Integrity Measurement) [43]
- NIST SP 800-57 (Key Management) [51]
- GlobalPlatform Card + SCP [25]
- ANSSI CSPN (certification de sécurité de premier niveau) [2]
- SESIP / EN 17927 (évaluation de plateformes IoT) [26]
- ETSI EN 303 645 (socle consumer IoT) [21]
- IEEE 802.1AR (Secure Device Identity) [28]
- FIPS 140-3, FIPS 203, FIPS 204 [49, 54, 55]
- CRA (UE 2024/2847) et RED Delegated Act (UE 2022/30) [24, 23]

Bibliographie

[1] ANSSI. Recommandations de securite relatives aux mecanismes cryptographiques. Guide ANSSI, 2020.

[2] ANSSI. Certification de securite de premier niveau (cspn) : referentiel. Schema de certification ANSSI, 2023.

[3] ANSSI. Recommandations de securite pour les objets connectes. Guide ANSSI, 2023.

[4] ANSSI. Securite de la chaine d'approvisionnement numerique. Guide ANSSI, 2023.

[5] ANSSI. Ebios risk manager. Methode de gestion des risques, 2024.

[6] ANSSI. Migration vers la cryptographie post-quantique : recommandations. Note de position ANSSI, 2024.

[7] ANSSI WooKey Project Contributors. Wookey architecture and design rationale. Project documentation, 2019.

[8] Arm. Arm security technology : Building a secure system using trustzone technology. Documentation technique Arm, 2019.

[9] Arm. Armv8-m security extensions : Requirements on development tools. Documentation technique Arm, 2021.

[10] Arm. Psa cryptography api specification. Specification, 2024.

[11] Arm PSA Certified. Psa certified security model and guidance. Program documentation, 2024.

[12] Eli Biham and Adi Shamir. Differential fault analysis of secret key cryptosystems. In *CRYPTO*, 1997.

[13] Dan Boneh, Richard A. DeMillo, and Richard J. Lipton. On the importance of checking cryptographic protocols for faults. *Journal of Cryptology*, 2001.

[14] Eric Brier, Christophe Clavier, and Francis Olivier. Correlation power analysis with a leakage model. In *CHES*, 2004.

[15] Jo Van Bulck et al. Foreshadow : Extracting the keys to the intel sgx kingdom with transient out-of-order execution. In *USENIX Security*, 2018.

[16] CISA, NIST, and NSA. Post-quantum cryptography roadmap and transition guidance. US Government guidance, 2025.

[17] Common Criteria Recognition Arrangement. Common criteria for information technology security evaluation (iso/iec 15408). Norme internationale, 2022.

[18] Debian Security Team. Dsa-1571-1 : Predictable random number generator in openssl. Security advisory, 2008.

[19] Arnaud Ebalard, Arnaud Fontaine, David Diallo, Loic Flori, Karim Khalfallah, Mathieu Renard, and Ryad Benadjila. Eurisko : developpement d'une carte electronique securisee. Actes SSTIC 2016, 2016.

[20] Maik Ender, Amir Moradi, and Christof Paar. The unpatchable silicon : A full break of the bitstream encryption of xilinx 7-series fpgas. In *USENIX Security*, 2020.

[21] ETSI. En 303 645 : Cyber security for consumer internet of things : Baseline requirements. ETSI European Standard, 2024.

[22] ETSI. Migration strategies and recommendations for quantum-safe cryptography. ETSI technical report, 2024.

[23] European Commission. Commission delegated regulation (eu) 2022/30 supplementing directive 2014/53/eu. Official Journal of the European Union, 2022.

[24] European Union. Regulation (eu) 2024/2847 (cyber resilience act). Official Journal of the European Union, 2024.

[25] GlobalPlatform. Globalplatform card specification v2.3.1. Specification, 2018.

[26] GlobalPlatform. Security evaluation standard for iot platforms (sesip), en 17927. Evaluation methodology, 2023.

[27] IEC. Iec 62443-4-2 : Technical security requirements for iacs components. Norme internationale, 2019.

[28] IEEE. Ieee std 802.1ar-2018 : Secure device identity. IEEE Standard, 2018.

[29] Intel. Intel software guard extensions (intel sgx) sdk developer reference. Documentation Intel, 2023.

[30] ISO/IEC. Iso/iec 11889 : Trusted platform module. Norme internationale, 2015.

[31] ISO/IEC. Iso/iec 27001 :2022 information security management systems. Norme internationale, 2022.

[32] ISO/SAE. Road vehicles – cybersecurity engineering (iso/sae 21434). Norme ISO/SAE, 2021.

[33] Paul Kocher et al. Spectre attacks : Exploiting speculative execution. In *IEEE S&P*, 2019.

[34] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In *CRYPTO*, 1999.

[35] Linaro and OP-TEE Contributors. Op-tee documentation. Project documentation, 2025.

[36] Moritz Lipp et al. Meltdown. In *USENIX Security*, 2018.

[37] lowRISC and OpenTitan contributors. Opentitan documentation and hardware specification. Project documentation, https://opentitan.org/, 2025.

[38] Roel Maes. Physically unclonable functions : Constructions, properties and applications. *Springer*, 2013.

[39] Amir Moradi, Alessandro Barenghi, Timo Kasper, and Christof Paar. On the vulnerability of fpga bitstream encryption against power analysis attacks. In *ACM CCS*, 2011.

[40] Michele Mosca. Cybersecurity in an era with quantum computers : Will we be ready ? *IEEE Security & Privacy*, 2018.

[41] NIST. Sp 800-38d : Recommendation for block cipher modes of operation : Galois/counter mode (gcm). Special Publication, 2007.

[42] NIST. Sp 800-147 : Bios protection guidelines. Special Publication, 2011.

[43] NIST. Sp 800-155 : Bios integrity measurement guidelines. Special Publication, 2012.

[44] NIST. Sp 800-88 rev. 1 : Guidelines for media sanitization. NIST Special Publication, 2014.

[45] NIST. Fips 180-4 : Secure hash standard (shs). Federal Information Processing Standard, 2015.

[46] NIST. Sp 800-90a rev. 1 : Recommendation for random number generation using deterministic random bit generators. Special Publication, 2015.

[47] NIST. Sp 800-193 : Platform firmware resiliency guidelines. Special Publication, 2018.

[48] NIST. Sp 800-90b : Recommendation for the entropy sources used for random bit generation. Special Publication, 2018.

[49] NIST. Fips 140-3 : Security requirements for cryptographic modules. Federal Information Processing Standard, 2019.

[50] NIST. Sp 800-131a rev. 2 : Transitioning the use of cryptographic algorithms and key lengths. Special Publication, 2019.

[51] NIST. Sp 800-57 part 1 rev. 5 : Recommendation for key management. Special Publication, 2020.

[52] NIST. Sp 800-218 : Secure software development framework (ssdf). Special Publication, 2022.

[53] NIST. Cybersecurity framework (csf) 2.0. Framework, 2024.

[54] NIST. Fips 203 : Module-lattice-based key-encapsulation mechanism standard. Federal Information Processing Standard, 2024.

[55] NIST. Fips 204 : Module-lattice-based digital signature standard. Federal Information Processing Standard, 2024.

[56] NIST. Fips 205 : Stateless hash-based digital signature standard. Federal Information Processing Standard, 2024.

[57] NIST. Nist ir 8547 : Transition to post-quantum cryptography standards. NIST Interagency Report, 2024.

[58] NIST. Post-quantum cryptography standardization project – status reports. Project documentation, 2025.

[59] Oracle. Java card platform specification, version 3.1. Specification, 2022.

[60] OWASP. Threat modeling cheat sheet. OWASP Cheat Sheet, 2024.

[61] Eric Rescorla. The transport layer security (tls) protocol version 1.3. RFC 8446, 2018.

[62] Peter W. Shor. Algorithms for quantum computation : Discrete logarithms and factoring. In *FOCS*, 1994.

[63] Ned Smith, Laurence Lundblade, et al. Remote attestation procedures (rats) architecture. RFC 9334, 2023.

[64] The Update Framework. The update framework specification. Specification, 2024.

[65] Tropic Square. Tropic square. Company website, https://www.tropicsquare.com/, 2025.

[66] Trusted Computing Group. A practical guide to tpm 2.0 and tcg tpm 2.0 library overview. TCG documentation, 2023.

[67] TrustedFirmware.org. Trusted firmware-m documentation. Project documentation, 2025.

[68] Uptane Alliance. Uptane standard for design and implementation. IEEE-ISTO Standard, 2023.

[69] Yuanzhong Xu, Weidong Cui, and Marcus Peinado. Controlled-channel attacks : Deterministic side channels for untrusted operating systems. In *IEEE S&P*, 2015.

Licence