Solutions de confiance pour la sécurité dans l'Internet des Objets

Louis Moreau

Thèse de Doctorat

Université de Limoges
ED 653 : SCIENCES ET INGENIERIE – XLIM
Faculté des Sciences et Techniques – Institut de Recherche XLIM

Thèse pour obtenir le grade de
Docteur de l'Université de Limoges
Spécialité Informatique

Présentée et soutenue par
Louis MOREAU
Le 12 Décembre 2022

SOLUTIONS DE CONFIANCE POUR LA SÉCURITÉ DANS L'INTERNET DES OBJETS

Thèse dirigée par Damien SAUVERON et Emmanuel CONCHON

JURY :

Président du jury
M. Yacine GHAMRI-DOUDANE, Professeur – L3I – Université de La Rochelle

Rapporteurs
Mme Samia BOUZEFRANE, Professeure – CEDRIC – Le Cnam
M. Serge CHAUMETTE, Professeur – Labri – Université de Bordeaux

Droits d'auteurs

Table des Matières

Table des Figures

Liste des Tableaux

Glossaire

Agrégation C'est le mécanisme de récupération des attestations au niveau du réseau. L'agrégation peut remonter jusqu'à un appareil unique ou être propagée dans tout le réseau.

Appareils, nœuds Ce sont deux façons interchangeables de nommer des objets connectés dans cette thèse. Le terme appareil est plus proche d'une considération physique, réelle de l'objet tandis que le terme nœud est plus proche d'une considération orientée réseau.

Attaque ToCToU (Time of Check Time of Use) Cette attaque repose sur le délai entre le moment où une vérification est effectuée, et le moment où elle est utilisée. Dans le cas de l'attestation, un logiciel malveillant pourrait se désinstaller ou se déplacer dans des portions de mémoire de façon à ne pas être détecté au moment de l'attestation.

Attaque *wormhole*, trou de ver Dans cette attaque, un attaquant sert de relais pour permettre un échange entre deux appareils qui n'auraient pas pu communiquer ensemble en temps normal.

Attaque *Man in the Middle* C'est une attaque dans laquelle un attaquant se place entre deux appareils communiquant ensemble, et peut écouter leurs échanges. Cet attaquant peut parfois également modifier les messages.

Attestation Il s'agit d'un élément de preuve permettant de valider l'état d'un appareil par rapport à un état correct connu par le *Verifier*. Les attestations sont souvent transmises sous forme de hachés.

Challenge-réponse Mécanisme durant lequel un appareil en interroge un autre de façon répétée, afin d'augmenter les probabilités de détecter une anomalie. En effet, un appareil sain aura toujours la bonne réponse dans un délai raisonnable, alors qu'il suffit d'une erreur pour identifier un appareil corrompu.

Cluster C'est un regroupement de plusieurs appareils qui interagissent entre eux et appartiennent à un réseau plus large, composé de plusieurs clusters.

Confiance C'est le fait que le réseau ou les appareils voisins considèrent un appareil comme étant dans un état normal. Cette confiance est réévaluée en fonction des résultats d'attestation.

Consensus Dans un processus d'attestation, il y a consensus lorsque l'attestation d'un appareil repose sur plusieurs autres appareils. En général, si une seule attestation échoue, l'appareil fautif sera exclu du réseau.

Control flow verification Il s'agit d'un mécanisme de vérification de l'exécution des différentes instructions du code d'un objet connecté, par comparaison à un arbre d'exécution connu.

Firmware C'est le code inclus dans un objet connecté, qui en assure les différentes fonctionnalités (communication, mesures,...)

Framework C'est un outil haut-niveau permettant d'englober plusieurs protocoles, et fournissant lui-même des messages et des processus haut-niveau.

Heartbeat, beat Il s'agit d'un message échangé de manière régulière pour s'assurer qu'un appareil est toujours connecté au réseau. Cela permet d'identifier un comportement anormal, et d'exclure un appareil qui a pu être compromis.

MCU (Memory Control Unit) C'est une partie de processeur garantissant l'utilisation de zones mémoires par des instructions autorisées.

Microcontrôleur Élément matériel généralement présent dans les objets connectés, dans lequel sont intégrés processeur, mémoire, et différents types d'entrées sorties (ADC, contrôleurs de bus,...).

Mobilité Les appareils prenant part à un réseau peuvent être fixes et ne jamais se déplacer, ou bien mobiles et réaliser des mouvements. Les appareils mobiles peuvent devenir fixes par moment, et vont souvent changer de voisins dans le réseau suite à leur mouvements.

Protocole C'est un ensemble de messages et de processus de communication défini pour que des appareils puissent échanger entre eux, ici leurs attestations.

Prover Appareil qui doit prouver son état à l'aide d'une attestation, régulièrement ou sur demande d'un *Verifier*.

Publish/Suscribe C'est un mécanisme d'envoi de messages dans lequel des *Publisher* envoient des messages sur un sujet donné (une information) à des *Subscriber* au

travers d'un système intermédiaire appelé *Broker*. C'est le cas avec le protocole MQTT par exemple.

PUF (Physically Unclonable Function) c'est un élément matériel qui ne peut être dupliqué, et répond toujours de la même manière à un challenge donné, permettant d'identifier de manière unique un appareil.

ROM (Read-Only Memory) Il s'agit de mémoire embarquée qui, une fois fabriquée en usine, ne peut plus être modifiée, garantissant que le code ou les données stockés ne seront pas modifiables.

TEE (Trusted Execution Environment) C'est une sous-partie d'un processeur qui garantit l'exécution sécurisée de code critique.

TPM (Trusted Platform Module) Il s'agit d'un microprocesseur sécurisé dédié à l'utilisation de clés cryptographiques.

Verifier Appareil qui reçoit l'attestation d'un ou plusieurs autres appareils (*Provers*) lors d'une phase d'attestation. Dans certains cas, c'est également l'appareil qui déclenche la phase d'attestation.

1 Introduction

Sommaire

1.1 L'Internet des Objets

N'importe quel objet faisant jusqu'à aujourd'hui partie du quotidien se voit désormais offrir la possibilité de communiquer sur Internet. Four, montre, drone, moissoneuse-batteuse, voiture, thermomètre, caméras, pacemaker, ces objets deviennent connectés dans l'objectif d'offrir de nouvelles fonctionnalités ou de se distinguer des produits existants. De par leur nombre sans cesse croissant, passé de 1,1 milliards à 11,3 milliards entre 2011 et 2021 [1][2], ces objets faisant partie de l'Internet of Things (IoT) dépassent même en nombre les objets électroniques non connectés.

1.1.1 Domaines d'application

Les domaines d'applications des objets connectés sont nombreux et en innovation constante. Parmi ceux-ci, on peut par exemple citer la smart industry, les smart cities ou encore la smart health.

La smart industry, qui est inclue dans ce qui est appelé industrie 4.0 ou quatrième révolution industrielle, repose sur l'augmentation de la connectivité et des automatismes intelligents dans le domaine de l'industrie. L'augmentation du nombre d'objets connectés dans ces milieux industriels permet des processus plus automatisés grâce à la surveillance continue de paramètres à l'aide de capteurs connectés, et à l'automatisation des réponses par le biais d'algorithmes dédiés. Un sous-exemple de la smart industry est le smart farming [3], qui est rendu possible grâce à des capteurs pouvant indiquer l'hygrométrie, la température, l'exposition à la lumière, pouvant mesurer la surface des feuilles et contrôler l'arrosage ou l'approvisionnement en engrais en retour.
Un autre avantage de ces nombreux capteurs connectés est l'arrivée de la maintenance prédictive [4] sur de nombreuses machines, permettant de réduire leur temps d'arrêt dû à des pannes et donc d'en maximiser la rentabilité. Par exemple, un capteur de vibrations embarqué sur un tractopelle peut permettre d'en étudier le comportement nominal à l'aide d'algorithmes d'intelligence artificielle, et de détecter une usure avancée des dents de la pelle, qui seront alors remplacées avant qu'un problème n'apparaisse.

De même, une smart city va utiliser de nombreux capteurs connectés pour accroître la quantité d'informations disponibles, et permettre d'agir en retour. À l'échelle d'une ville, les éléments mesurés sont notamment l'approvisionnement en électricité, celui en eau et les systèmes de circulation et de transport.
Un meilleur contrôle de l'alimentation électrique permet d'accroître la flexibilité du réseau et l'intégration de technologies de production d'énergies renouvelables comme l'éolien ou le solaire.
Le contrôle de l'approvisionnement en eau permet de son côté une meilleure qualité de l'eau ainsi que la réalisation d'économies d'échelle en optimisant les approvisionnements et en détectant les fuites sur le réseau. En ce qui concerne les systèmes de transport, la connectivité permet d'améliorer la fluidité du trafic en répondant de manière plus précise aux variations de flux d'usagers, ou encore en pilotant de manière intelligente la signalisation lumineuse des intersections.

La smart health, ou encore e-santé, repose également sur cette interface entre le réel et le virtuel. Cela peut s'appliquer à domicile, par exemple au travers de balances, de thermomètres ou de tensiomètres connectés, ou encore de détecteurs de chutes à destination des personnes âgées. Mais cela peut également s'appliquer directement dans les hôpitaux avec du matériel tels que les dosimètres connectés, les pousses-seringues, ou encore des robots permettant de réaliser des opérations chirurgicales à distance.

Les exemples d'objets connectés et leurs domaines d'applications sont ainsi très divers, mais des caractéristiques communes permettent de définir ce qui constitue un objet connecté.

1.1.2 Caractéristiques d'un objet connecté

Un objet connecté, comme représenté schématiquement figure 1.1, est un objet physique qui vient se placer à l'interface entre le monde réel et le monde informatique. Souvent, un objet connecté est un objet du quotidien pour lequel l'ajout de la connectivité vient apporter des fonctionnalités de remontée d'information ou de contrôle à distance qui en améliorent l'usage.
Connecter un objet permet d'apporter deux usages principaux, la récupération d'informations à distance, et le déclenchement d'actions à distance. Récupérer des informations à distance à l'aide de capteurs permet d'accroître fortement la quantité de données disponibles via des mesures plus régulières (on peut ici parler de *Big-Data*), et ainsi d'assurer un suivi automatique et intelligent de ces mesures, permettant par exemple de déclencher des alertes en temps réel. Le déclenchement d'action peut se

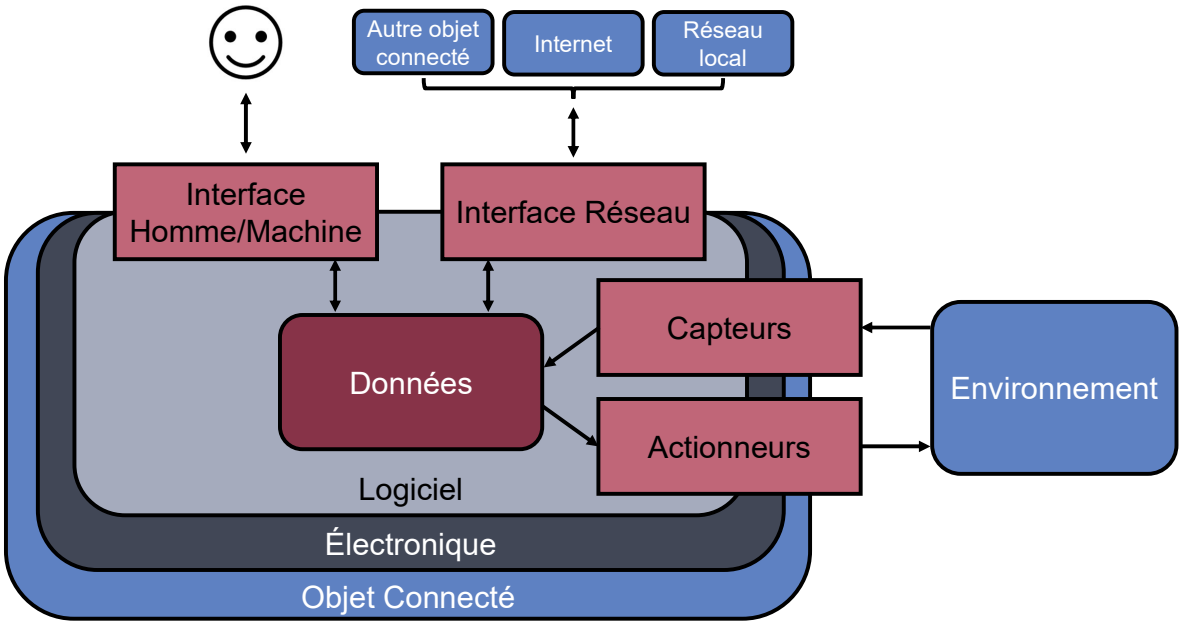


FIGURE 1.1 – Représentation schématique d'un objet connecté

faire en réponse automatique à ces mesures, ou sous l'action d'un opérateur distant. L'intelligence embarquée dans les objets connectés s'accroît sans cesse, allant jusqu'à l'utilisation d'algorithmes d'intelligence artificielle qui permettent des traitements très spécialisés (par exemple compter le nombre de personnes à l'aide de caméras et d'algorithmes de reconnaissance d'image).

Ces objets sont connectés sur différents types de réseaux : des réseaux locaux, par exemple dans les smart homes ; des réseaux mesh comme parfois dans le cas des drones ; des réseaux nécessitant des stations de base qui agissent comme relais entre les objets connectés eux-mêmes ; ou encore des réseaux basés sur Internet et un cloud. Les moyens de communications sont soit filaires soit sans-fil et reposent sur une large liste de technologies et protocoles existants. Parmi les technologies filaires, Ethernet [5], RS-232 ou RS-485 sont couramment utilisées : Ethernet permet des connexions locales ou vers Internet en utilisant en général le protocole IP ; RS-232 permet la connexion de deux appareils proches entre eux et RS-485 permet la connexion de plusieurs appareils proches entre eux avec un système d'adressage. Les technologies radio sans-fil se distinguent par les fréquences utilisées et leur portée de communication. Ainsi, les technologies NFC [6] ou RFID [7] vont être utilisées au contact ou à une distance très courte. Les technologies comme ZigBee [8] ou le BlueTooth [9] vont être utilisées à quelques dizaines de mètres. Le Wi-Fi [10] sera utilisé jusqu'à quelques centaines de

mètres. Enfin, des technologies comme le LTE [11] ou LoRa [12] vont être utilisées pour des distances plus grandes selon les besoins du cas d'usage.

Une autre des caractéristiques des objets connectés est le fait qu'ils possèdent en général de faibles ressources de calculs, soit par souci d'économie financière en vue de leur commercialisation sur un marché de masse, soit par souci d'économie d'énergie. Sur l'aspect financier, il faut en effet que le bénéfice apporté par la connectivité surpasse le coût additionnel, ce qui est parfois complexe lorsque les objets sont peu coûteux. Concernant l'aspect économie d'énergie, il est souvent dicté par le contexte : en effet, certains objets connectés doivent être placés loin de toute source d'alimentation pour répondre à leur usage, et doivent donc avoir une autonomie suffisante (pour certains appareils, cela peut se compter en années). Des compromis de performances et de fréquences de fonctionnement doivent alors être réalisés.
La plupart du temps, ces objets connectés vont également bénéficier d'une interface homme-machine. Parfois accessible sur le réseau ou sur une interface locale comme un écran ou un port série, cette interface permet en général de paramétrer les appareils ou d'en inspecter le bon fonctionnement en détails.
Enfin, ces objets connectés vont évoluer dans différents contextes de déploiement. Ces appareils peuvent entre autres être mobiles (drones, véhicules,...) ou fixes (réfrigérateur, station météo,...). Mais surtout, ces appareils peuvent être déployés dans des environnements hostiles, et être sujets à des attaques réseaux, logicielles ou physiques : ils doivent donc être convenablement sécurisés.

Cependant, de nombreux concepteurs d'objets connectés ne prennent pas suffisamment en compte la sécurité de ceux-ci, permettant l'exploitation de failles de sécurité par des personnes malintentionnées. Ces attaques amènent souvent à l'intégration des objets connectés dans des botnets, et conduisent à des dégâts sur le plan économique [13] comme sur le plan humain [14].

1.1.3 La sécurité des objets connectés

Afin de répondre à ce besoin de sécurité des objets connectés, de nombreuses solutions existent et sont sans cesse améliorées, avec pour objectif de répondre aux contraintes de confidentialité, d'intégrité et de disponibilité. Ces solutions doivent être adaptées aux contraintes [15] que posent les objets connectés que sont leurs faibles performances de calcul, leur grande hétérogénéité ou encore leurs limitations de consommation énergétique. De plus, comme le montre le schéma figure 1.1, la sécurité d'un objet connecté doit englober plusieurs aspects : L'interface homme/machine, la connexion aux différents types de réseaux de communication, les données traitées et leur stockage,

l'aspect logiciel que ce soit les applications ou le système d'exploitation, ou encore l'aspect physique en ce qui concerne la carte électronique et en particulier les capteurs ou actionneurs qui la composent.

Ainsi, l'interface homme/machine doit être sécurisée en ce qui concerne l'authentification des utilisateurs et la sécurité de l'interface en elle-même. L'authentification doit respecter les bonne pratiques en termes de longueur de mot de passe et de séparation des privilèges par exemple. L'interface ne doit pas contenir de failles de sécurité en elle-même. Par exemple si cette interface repose sur un serveur Web disponible sur le réseau, des attaques comme les injections SQL ou les XSS (ou celles de l'OWASP Top Ten [16]) ne doivent pas être possibles.

De son côté, la connexion aux différents types de réseaux, filaires ou non, doit également être sécurisée. La méthode de sécurité la plus efficace pour sécuriser ces communications est la cryptographie, qui permet le chiffrement des données qui transitent sur le réseau, et donc l'assurance que la confidentialité et l'intégrité de ces données sont garanties. Les protocoles de communication proposent souvent des techniques de chiffrement, c'est le cas par exemple du Wi-Fi après une phase d'authentification WPA2. Cependant, le réseau étant souvent considéré comme maîtrisé par l'attaquant, un chiffrement de bout en bout est la plupart du temps recommandé, en utilisant des primitives cryptographiques éprouvées. Cela peut également s'appliquer aux données stockées sur les objets connectés, qui peuvent être chiffrées dans la mémoire de l'appareil lorsqu'elles n'ont pas besoin d'être accédées. Dans le but d'assurer une sécurité minimale, ces données peuvent être authentifiées à l'aide de codes d'authentification de message de hachage à clé ou de signatures cryptographiques.

Les logiciels utilisés par l'objet connecté sont également des surfaces d'attaques potentielles, que ce soient les applications embarquées ou même l'OS. Un moyen de sécuriser ces éléments peut être l'utilisation de TEE (Trusted Execution Environnement) qui permet d'isoler des tâches entre un monde sécurisé et un monde non-sécurisé, de telle sorte que les applications qui ne sont pas de confiance n'ont pas d'impact sur le reste du fonctionnement de l'objet connecté. Les applications critiques ou l'OS peuvent également être vérifiés formellement afin d'en valider les propriétés de sécurité par rapport à un modèle de sécurité établi.

Enfin, la sécurité va également reposer sur les composants physiques intégrés à l'objet connecté, certains étant développés avec de forts pré-requis de sécurité, voir uniquement dédiés à la sécurité. Par exemple, le TPM (Trusted Platform Module), comme la gamme ATECC608 [17] de chez Microchip, est fréquemment utilisé pour le stockage de

clés cryptographiques et permet l'utilisation de fonctions cryptographiques associées, implémentées de façon matérielle sécurisée (notamment, elles ne provoquent pas de fuite via des canaux auxiliaires). D'autres composants comme les MPU (Memory Protection Unit) vont permettre l'accès aux différentes zones mémoire de stockage uniquement aux processus qui en ont le droit afin de limiter l'accès aux données ou fonctions critiques. Également, les PUF [18] (Physically Unclonable Functions) permettent d'identifier de manière unique un appareil et sont donc également utilisées comme sécurité physique.

Dans cette thèse, l'accent va être mis sur l'attestation à distance, dont les différentes composantes concernent à la fois le matériel, l'applicatif et le réseau.

1.1.4 L'attestation à distance

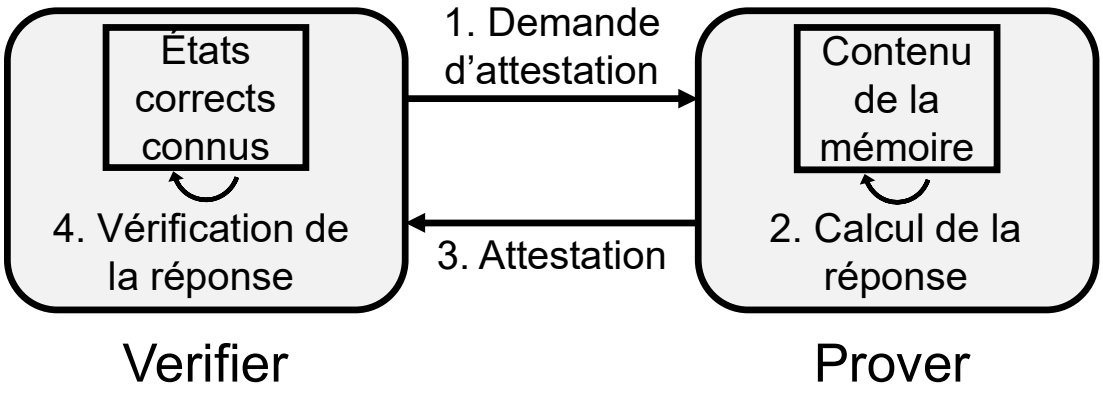


FIGURE 1.2 – Représentation haut-niveau du processus d'attestation à distance

L'attestation à distance, dont un schéma général est présenté figure 1.2, est un mécanisme qui permet d'améliorer la sécurité des réseaux d'objets connectés. L'attestation consiste à vérifier l'état actuel de la mémoire ou des processus d'un appareil en comparaison à des états connus préalablement afin de s'assurer de leur intégrité. Cette attestation est en général réalisée en réaction à la demande faite par un autre appareil distant, mais elle peut également être réalisée de manière proactive. Parmi les propositions de solutions existantes, différentes catégories de mécanismes d'attestation existent. Elles peuvent reposer sur des protocoles de communication, sur l'utilisation de composants matériels ou encore sur des briques logicielles seules, comme le présentera en détails le chapitre 2.

Les protocoles de communication vont en général reposer sur des standards existants comme TCP/IP [19][20], HTTP [21] ou encore MQTT [22], et l'objectif consiste alors à proposer des mécanismes efficaces utilisant ces protocoles et permettant la bonne communication des attestations entre les différents appareils impliqués dans le processus d'attestation à distance.
L'utilisation de composants matériels va elle se concentrer sur l'obtention en local d'une attestation permettant d'assurer au mieux la sécurité, ce qui peut passer par l'analyse détaillée de la mémoire et des fonctions en cours d'exécution.
Dans ces deux cas, l'attestation à distance repose souvent sur l'utilisation de primitives cryptographiques telles que les signatures ou les fonctions de hachage.

Les signatures font partie de la cryptographie asymétrique, également appelée cryptographie à clé publique. La cryptographie a clé publique repose sur des paires de clés, l'une publique communicable à tout le monde et l'autre privée qui doit rester confidentielle. Un appareil peut signer des données à l'aide de sa clé privée, et n'importe quel autre appareil peut vérifier cette signature à l'aide de la clé publique correspondante, ce qui assure l'authentification et l'intégrité des données signées.
Une autre solution cryptographique communément utilisée repose sur les fonctions de hachage. Les fonctions de hachage sont utilisées pour obtenir une valeur de taille fixe (un haché) à partir de données d'entrée de différentes tailles de manière déterministe. En cryptographie, on s'intéresse principalement aux fonctions de hachage unidirectionnelles sans collision. Dans le cadre de l'attestation à distance, ces fonctions sont notamment utilisées pour produire des HMAC (Hash-based Message Authentication Code) qui permettent d'authentifier des données sur la base d'une clé cryptographique partagée. Les HMAC sont moins coûteux en temps de calcul, en espace silicium et en consommation d'énergie que d'autres solutions cryptographiques comme les signatures, et sont ainsi particulièrement adaptés aux objets connectés.

Ainsi, c'est spécifiquement sur l'attestation à distance que porte ce travail de thèse, et en particulier sur une proposition de framework qui permet d'utiliser conjointement de multiples protocoles d'attestation à distance.

1.2 Contributions et plan

Cette thèse comporte différentes contributions :

— Elle propose tout d'abord un état de l'art détaillé de ce qu'est l'attestation à distance et des différents axes d'étude qui la composent.

— Cette thèse propose également à notre connaissance la première proposition de framework d'attestation continue à distance capable d'inclure différents mécanismes d'attestation afin de répondre de façon agnostique aux différents cas d'usage.

— Cette proposition de framework s'appuie particulièrement sur une définition générique d'un réseau d'objets connectés, et fait également l'objet d'une analyse de sécurité.

— Enfin, ce framework fait l'objet d'une étude de performances détaillée au travers de simulations.

Ces travaux ont fait l'objet d'un article publié dans un journal international, et de deux articles en cours de soumission :

— L. Moreau, E. Conchon, and D. Sauveron, "CRAFT : a Continuous Remote Attestation Framework for IoT," IEEE Access, vol. 9, pp. 46 430–46 447, 2021. [Online]. Available : https://ieeexplore.ieee.org/document/9382291

— L. Moreau, E. Conchon, and D. Sauveron, "An Adaptive Simultaneous Multi-Protocol extension of CRAFT"

— L. Moreau, E. Conchon, and D. Sauveron, "A comprehensive survey of remote attestation and its evolution"

Le manuscrit se découpe en cinq chapitres organisés comme suit :

— Chapitre 2 : Ce chapitre introduit un état de l'art des travaux concernant l'attestation à distance. Cet état de l'art consiste tout d'abord en une présentation générale de ce qu'est l'attestation à distance, en précisant les grandes sous-catégories qui la composent. Une chronologie est également fournie, permettant d'identifier quels travaux ont le plus influencé l'état de l'art. Les différentes sous-catégories d'attestation à distance sont ensuite présentées : attestation matérielle, logicielle et hybride. L'attestation matérielle propose des solutions qui reposent sur l'ajout de composants matériels spécialisés dans la sécurité comme les TPM ou les TEE, et sur leur bonne utilisation dans le cadre de l'attestation d'objets connectés. L'attestation logicielle, à l'inverse, repose sur des solutions uniquement logicielles. Leur base de sécurité est donc moins robuste, mais il n'y a pas de coût matériel supplémentaire. Enfin, l'attestation hybride repose sur des composants matériels plus classiques comme la ROM. Dans l'attestation hybride, deux types de travaux se distinguent : ceux qui se concentrent sur l'utilisation des composants matériels

et sur les solutions au niveau d'un seul appareil, et ceux qui proposent des protocoles permettant de diffuser l'attestation au sein d'un réseau.

— Chapitre 3 : Dans ce chapitre, la définition de ce qu'est un framework d'attestation continue est donnée. Un tel framework permet d'être utilisé dans n'importe quel réseau d'objets connectés, car il repose sur la définition large de ce que constitue un tel réseau également donnée dans ce chapitre. Un framework d'attestation continue doit notamment être capable d'utiliser n'importe quel protocole d'attestation existant et de les faire cohabiter. Il doit également ne pas ajouter de surcoût en performances trop élevé au niveau du réseau. En termes de sécurité, un framework doit également être capable de gérer l'exclusion des appareils malveillants ou attaqués. Un framework doit également être utilisable dans n'importe quel réseau malgré la grande diversité de ceux-ci. En effet, dans un réseau les appareils peuvent avoir des niveaux de performance ou de sécurité très variés (du simple thermomètre à la station de base), et peuvent également avoir différents degrés de mobilités (de l'antenne fixée sur un immeuble au véhicule connecté mobile). Les définitions précises de réseau et de framework permettent de proposer CRAFT, le premier framework d'attestation à distance continue. CRAFT se compose de deux phases, *Hors-ligne* et *En-ligne*, et repose sur un ensemble minimal de messages pour assurer le processus d'attestation continue. Parmi ces messages se retrouvent notamment ceux qui intègrent les protocoles d'attestation existants, et également des messages de type *heartbeat* qui assurent la continuité de la connexion des appareils, ce qui permet la détection rapide d'attaques sur ces appareils. En plus de ces messages minimaux, dans ce chapitre, les fonctionnalités ASMP (Adaptive Simultaneous Multi-Protocols) sont détaillées : ce sont celles qui permettent à un appareil de changer de protocole d'attestation en fonction du contexte. Par exemple, un appareil mobile qui deviendrait périodiquement stationnaire pourrait utiliser un protocole donné durant ses déplacements, et un autre protocole à l'arrêt. Dans ce chapitre, une analyse de la sécurité de CRAFT est également proposée sur la base d'un modèle de menaces adapté à l'attestation à distance. Cette analyse démontre que ce framework résiste à une large variété d'attaques utilisées contre les réseaux d'objets connectés.

— Chapitre 4 : Ce chapitre présente les évaluations des performances de CRAFT, réalisées à l'aide de simulations sur le simulateur de réseaux à événements discrets Omnet++. Ces simulations sont divisées en deux sous-parties : la comparaison de CRAFT à deux protocoles d'attestation seuls, et la comparaison de

CRAFT avec et sans les fonctionnalités ASMP à la combinaison de deux protocoles d'attestation. Ces simulations reposent sur une méthodologie statistique qui garantit la pertinence des résultats grâce au nombre de répétitions. Elles permettent de démontrer les bonnes performances du framework en termes d'impact sur le réseau, sur les performance de calcul, ainsi que sur la bonne exclusion d'appareils. En comparaison avec les protocoles existants, CRAFT s'illustre ainsi très positivement sur ces points, tout en étant utilisable sur n'importe quel type de réseau grâce à sa grande flexibilité.

— Chapitre 5 : Ce chapitre conclura la thèse en présentant un résumé des différentes contributions, ainsi qu'en décrivant des perspectives de travaux futurs et notamment les fonctionnalités qui pourraient être ajoutées dans les frameworks d'attestation tels que CRAFT.

2

État de l'art de l'attestation à distance

Sommaire

2.1 Introduction du chapitre

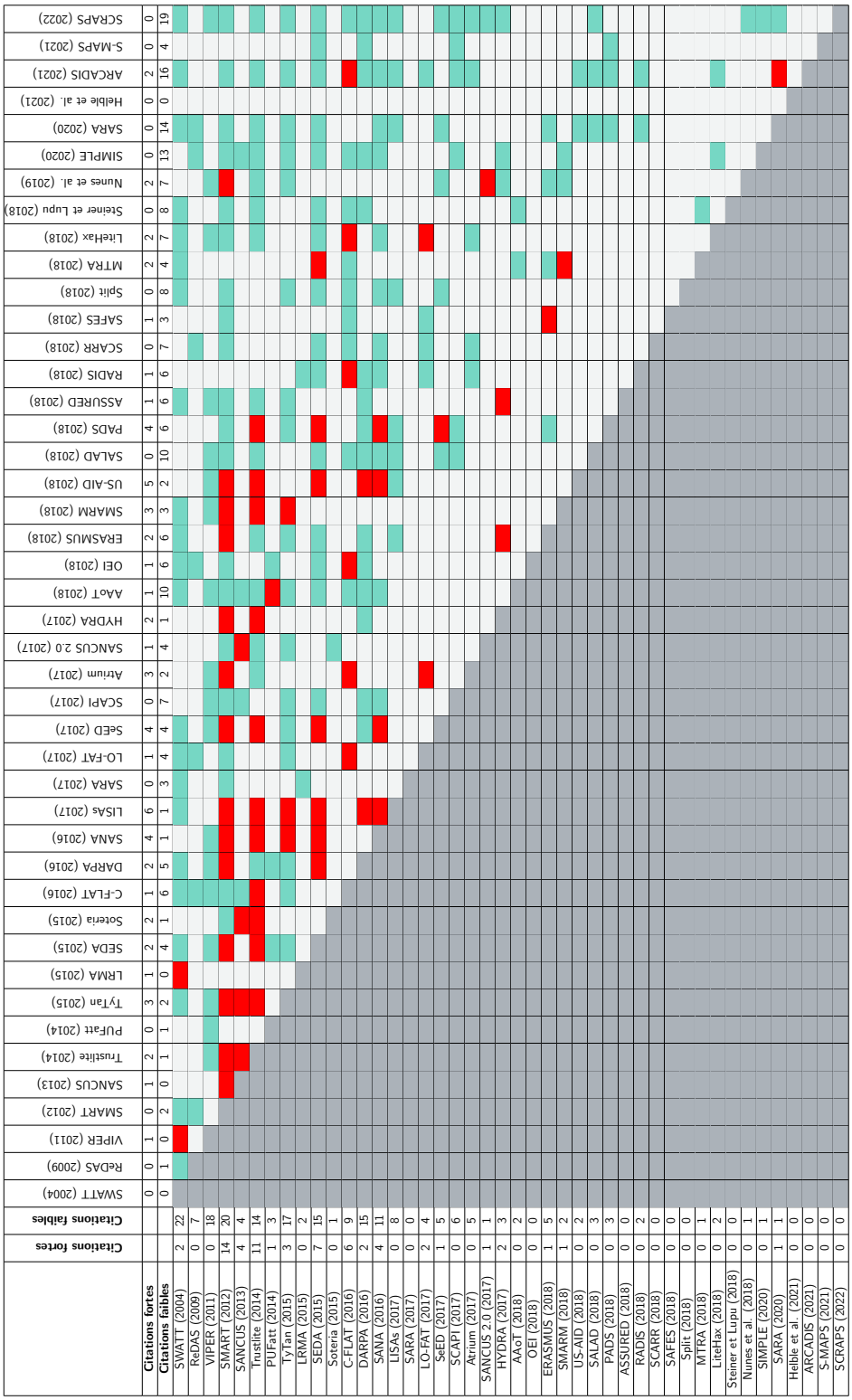
Dans ce chapitre, l'état de l'art des mécanismes d'attestation à distance est présenté. La section 2.2 décrit les généralités de ce que sont les mécanismes d'attestation. La section 2.3 décrit plus en détail les mécanismes d'attestation à distance matérielle qui peuvent être implémentés sur un appareil. La section 2.4 présente quant à elle les mécanismes d'attestation à distance logicielle qui peuvent être intégrés aux appareils d'un réseau pour en améliorer la sécurité. Enfin, la section 2.5 décrit les mécanismes d'attestation à distance hybrides, qui peuvent soit avoir une portée locale à l'appareil et utiliser les ressources matérielles disponibles, soit avoir une portée au niveau du réseau en tant que protocole de communication qui sert à l'attestation des appareils.

2.2 Présentation générale de l'attestation à distance

Plusieurs techniques d'attestation à distance ont émergé, avec pour objectif de maintenir la confiance dans un réseau. Ces solutions cherchent toutes à montrer que les appareils sont dans un état connu et correct en ce qui concerne leur logiciel et leur mémoire, de telle sorte que la confiance en ces appareils leur permette de participer aux opérations du réseau. Cependant, la majorité de ces propositions se concentrent sur des aspects spécifiques de l'attestation : un appareil seul, un réseau d'appareils, des réseaux d'appareils mobiles, ... Une solution globale, tel un framework, capable d'actionner ces différents propositions semble donc nécessaire.
Parmi les mécanismes agissant à l'échelle du réseau, divers protocoles d'attestation à distance utilisent un mécanisme d'agrégation avec un *vérificateur initial* pour diffuser une requête d'attestation au travers du réseau en utilisant un arbre de diffusion ; les

réponses à cette requête suivant le chemin inverse jusqu'au *vérificateur initial* [23]. D'autres protocoles rendent possible l'agrégation dans chaque appareil, ce qui permet alors à chaque appareil d'être utilisé comme vérificateur [24]. D'autres encore n'utilisent qu'une attestation de voisin à voisin, la confiance des appareils en leur voisins reposant alors sur un consensus local [25]. Parmi les mécanismes d'attestation au niveau d'un appareil, certains [26][27] tentent d'être efficaces tout en ayant le moins d'exigences matérielles possible afin d'être utilisables sur le plus grand nombre d'appareils quelque soit leurs performances. D'autres [28][29] au contraire cherchent à utiliser au mieux les possibilités offertes par des composants matériels tels que les TPM (Trusted Platform Module) [30] ou un TEE (Trusted Execution Environment) [31] par exemple.

La majorité des solution d'attestation à distance peut être divisée en trois catégories : l'attestation matérielle, l'attestation logicielle, et l'attestation hybride (locale ou réseau), qui sont présentées en détails dans les sections suivantes.
La figure 2.1 et la table 2.1 présentent une vision synthétique des travaux réalisés dans ces différentes catégories, et en particulier de leur influence sur l'évolution de l'attestation à distance.
Dans la figure 2.1, les articles sont listés chronologiquement, sur quatre axes représentant les quatre principales sous-catégories d'attestation utilisées ici. Pour chaque solution, le nombre d'autres solutions listées ici qu'elle a influencé est donné sous la forme "nombre de citations fortes/nombre de citations faibles" (par exemple 14/20 pour SMART). Une citation forte signifie que l'article est utilisé comme base de réflexion, comme point de comparaison, ou comme composant d'une solution en plus d'être cité dans l'état de l'art Une citation faible constitue une simple apparition dans la bibliographie ou dans l'état de l'art d'un article. Par exemple, SMART a été fortement utilisé dans 14 articles, et cité dans 20 articles supplémentaires parmi ceux présentés ici.
La table 2.1 contient des informations plus détaillées concernant les références croisées des articles. Les articles en colonne citent les articles en ligne, en distinguant les citations faibles et fortes par deux niveaux de couleur (respectivement vert et rouge). Par exemple, TrustLite se base fortement sur les retours d'expérience de SMART [32] et SANCUS [33], et cite uniquement VIPER [34]. La couleur blanche dénote l'absence de citation, et la couleur grise l'impossibilité de citer car les travaux ne peuvent se citer eux-même ou citer un article qui leur est postérieur. Ainsi, SMART ne cite aucun article car c'est le premier de la liste. En revanche, l'article d'Helble *et al.*, bien que publié en 2021, ne semble pas s'inscrire dans la continuité existante d'articles sur l'attestation à distance.
Dans les sections suivantes, ces différents travaux sont résumés par ordre chronologique de publication et regroupés selon leur catégorie d'attestation (matérielle, logicielle ou hybride). Ces différentes catégories sont également présentées plus en détails.



	Citations fortes	Citations faibles	Citations fortes	Citations faibles
SWATT (2004)	0	0		
ReDAS (2009)	0	1		
VIPER (2011)	1	0		
SMART (2012)	0	2		
SANCUS (2013)	1	0		
Trustlite (2014)	2	1		
PUFatt (2014)	0	1		
TyTan (2015)	3	2		
LRMA (2015)	1	0		
SEDA (2015)	2	4		
Soteria (2015)	2	1		
C-FLAT (2016)	1	6		
DARPA (2016)	2	5		
SANA (2016)	4	1		
LISAs (2017)	6	1		
SARA (2017)	0	3		
LO-FAT (2017)	1	4		
SeED (2017)	4	4		
SCAPI (2017)	0	7		
Atrium (2017)	3	2		
SANCUS 2.0 (2017)	1	4		
HYDRA (2017)	2	1		
AAoT (2018)	1	10		
OEI (2018)	1	6		
ERASMUS (2018)	2	6		
SMARM (2018)	3	3		
US-AID (2018)	5	2		
SALAD (2018)	0	10		
PADS (2018)	4	6		
ASSURED (2018)	1	6		
RADIS (2018)	1	6		
SCARR (2018)	0	7		
SAFES (2018)	1	3		
Split (2018)	0	8		
MTRA (2018)	2	4		
LiteHax (2018)	2	7		
Steiner et Lupu (2018)	0	8		
Nunes et al. (2019)	2	7		
SIMPLE (2020)	0	13		
SARA (2020)	0	14		
Helble et al. (2021)	0	0		
ARCADIS (2021)	2	16		
S-MAPS (2021)	0	4		
SCRAPS (2022)	0	19		

TABLE 2.1 – Table des références des travaux sur l'attestation à distance (colonne cite ligne)

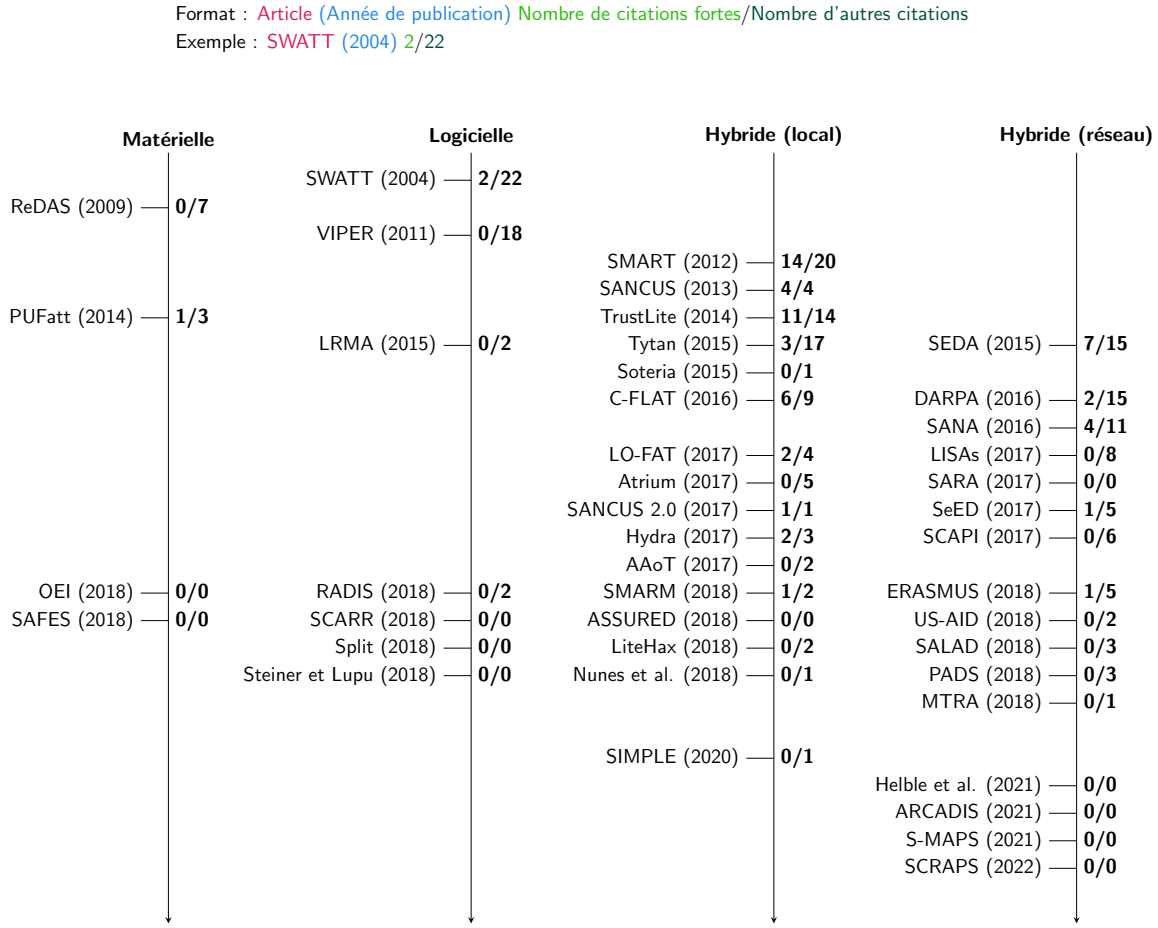


FIGURE 2.1 – Chronologie des travaux sur l'attestation à distance par catégorie d'attestation

2.3 Attestation à distance matérielle

Les solutions d'attestation matérielles sont celles qui nécessitent des fonctionnalités de sécurité matérielles spécifiques, comme un TPM (Trusted Platform Module) [30] ou un TEE (Trusted Execution Environment) [31]. Les solutions d'attestation matérielles permettent d'améliorer la sécurité des dispositifs équipés face à une plus grande diversité d'attaques que les solutions logicielles, notamment les attaques physiques contre lesquelles les composants matériels sont protégés. Cependant, ces solutions ne fournissent un mécanisme d'attestation qu'à l'échelle du dispositif et non pas à l'échelle du réseau, il convient de les utiliser conjointement avec des protocoles d'attestation.

— En 2009, Kil *et al.* ont proposé ReDAS (Remote Dynamic Attestation System) [35]. Contrairement aux solutions d'attestation précédemment proposées qui ne permettent que l'attestation de régions de mémoire statiques à l'aide de fonctions de hachage, ReDAS utilise un TPM pour attester que l'exécution de systèmes

dynamiques est sécurisée. Pour ce faire, ReDAS utilise deux propriétés dynamiques : l'intégrité structurelle du système, et l'intégrité globale des données. Ces deux propriétés permettent d'attester des applications dynamiques au niveau des dispositifs. Un des challenges posé par de tels systèmes dynamiques consiste à être capable d'identifier l'ensemble des états corrects. ReDAS permet ici d'extraire automatiquement les contraintes correspondant aux deux propriétés dynamiques utilisées. En utilisant ces contraintes, ReDAS peut ensuite détecter les attaques avec un faible coût en performance et sans faux positifs.

— "PUFatt : Embedded Platform Attestation Based on Novel Processor-Based PUFs" a été proposé par Kong *et al.* en 2014. Les PUFs (Physically Unclonable Functions) reposent sur les variations de construction d'éléments hardware comme les microprocesseurs, qui font que chaque élément est distinguable d'un autre par une série de questions-réponses. Les auteurs ont implémenté leur proposition à l'aide d'unités logiques et arithmétiques de base sur un FPGA, et des éléments identiques produisent en effet des réponses différentes à cause des variations de construction intrinsèques. Ainsi, les PUFs représentent une solution hardware peu coûteuse qui peut servir de racine de confiance sur des objets connectés peu complexes.

— "OEI : Operation Execution Integrity for Embedded Devices" a été proposé par Sun *et al.* en 2018. OEI permet de vérifier l'intégrité à la fois des flux d'exécution et des variables critiques à l'aide du système OAT (OEI Attester) sur des systèmes embarqués basés sur ARM (une carte HiKey avec un ARM Cortex-A53 équipé de TrustZone dans l'implémentation hardware). À la compilation, OAT instrumente le contrôle de flux et les variables critiques, afin de permettre la création du mécanisme de mesures qui va effectuer des contrôles sur ces deux points depuis le *Secure World* de TrustZone.

— En 2018, Kobayashi *et al.* ont proposé SAFES (Sand-boxed Architecture for Frequent Environment Self-measurement) [29]. L'objectif de SAFES est de réduire le temps entre le moment auquel est attesté le dispositif (Time of Check) et le moment où le code attesté est utilisé (Time of Use). Pour répondre à cette problématique, SAFES utilise un TEE (Trusted Execution Environment) qui est ARM TrustZone. TrustZone permet l'isolation entre un monde normal et un monde sécurisé, et permet de contrôler le code du monde normal depuis le monde sécurisé. Avec SAFES, le code peut être contrôlé à chaque événement Entrée/Sortie puis exécuté immédiatement après vérification, minimisant ainsi le délai entre les deux opérations.

Les solutions d'attestation à distance matérielles permettent ainsi de mettre en place des racines de confiance robustes au travers de mécanismes comme TrustZone ou les PUFs. Cependant, ces mécanismes sont contraignants car ils doivent être inclus dès la conception des objets connectés et ne peuvent donc pas fonctionner sur des appareils déjà déployés. Ils peuvent de plus représenter un surcoût financier en production, et une plus grande consommation énergétique liée à l'ajout des composants matériels.

2.4 Attestation à distance logicielle

Les solutions d'attestation à distance logicielles sont celles qui n'utilisent aucun composant matériel supplémentaire pour fonctionner. Cela rend ces solutions moins coûteuses, cependant elles sont plus sujettes à des attaques, en particulier au niveau physique puisque le matériel n'est pas sécurisé. Ces solutions sont principalement destinées à être utilisées dans des produits ayant de très faibles capacités comme les microcontrôleurs 8 bits, ou encore dans des dispositifs déjà existants afin de ne pas avoir à les modifier.
Les solutions d'attestation logicielles reposent souvent sur un mécanisme de question-réponse [36][37]. Ce mécanisme prend en compte le temps de réponse de l'appareil interrogé pour maintenir la confiance en celui-ci, en partant du principe qu'un appareil attaqué répondra différemment d'un appareil ne l'ayant pas été, souvent plus lentement. D'autres solutions [38][39] appliquent des principes utilisés d'ordinaire en attestation matérielle, mais en travaillant avec une racine de confiance logicielle, c'est à dire qui repose sur du code bas niveau comme le kernel.

— Seshadri *et al.* ont publié SWATT (SoftWare-based ATTestation for Embedded Devices) [36] en 2004. SWATT est l'un des premiers protocoles d'attestation à distance logicielle, dont l'objectif est de rendre la solution disponible sur le plus grand nombre de dispositifs possible, sans pré-requis matériel. SWATT fonctionne sur la base d'un protocole de question-réponse, à l'initiative d'un vérificateur externe. Le vérificateur interroge le dispositif à vérifier, qui doit lui renvoyer le haché de la partie de code demandée. Un dispositif compromis mettra plus de temps à répondre à cause des modifications apportées, et plus le vérificateur interroge un dispositif compromis de nombreuses fois, plus des écarts en temps de réponse pourront être remarqués, permettant ainsi d'exclure cet appareil du réseau. En revanche, SWATT ne permet pas de contrer une attaque physique sur le dispositif, l'ajout de mémoire additionnelle permettant par exemple de contourner SWATT en ne modifiant pas la mémoire initiale et en stockant le code malveillant dans la mémoire supplémentaire.

— VIPER (Verifying the Integrity of PERipherals' Firmware) [34] a été publié par Li *et al.* en 2011, avec pour objectif de permettre de vérifier l'intégrité des firmwares d'un périphérique à l'aide d'une solution uniquement logicielle. De manière similaire à SWATT, VIPER repose sur un système de questions réponses entre le processeur hôte et les périphériques, qui repose sur le hachage des firmwares à attester. VIPER repose sur le fait que les performances en cas d'attaque sont dégradées, et donc sont détectables. Pour confirmer l'efficacité de VIPER en matière de détection, une attaque par proxy basée sur un cas réel a été implémentée à l'aide de deux Netgear GA620, l'un étant l'appareil légitime et l'autre servant de proxy. L'utilisation de VIPER permet alors de distinguer l'attestation réalisée par l'un ou l'autres des appareils.

— En 2015, Yang *et al.* ont proposé LRMA (Towards a Low-Cost Remote Memory Attestation for the Smart Grid) [40]. L'objectif de LRMA est de diminuer le coût en calcul et en usage réseau des solutions d'attestation logicielle. LRMA se base ainsi sur SWATT, en prenant en compte les délais réseaux séparément du délai de réponse total, ce qui permet d'améliorer la performance de LRMA par rapport à SWATT. De plus, LRMA comptabilise le nombre d'échecs d'attestation, ce qui permet alors d'ajuster dynamiquement la fréquence d'attestation, et ainsi d'améliorer la probabilité de détection d'un attaquant.

— Conti *et al.* ont publié RADIS (Remote Attestation of Distributed IoT Services) [38] en 2018. RADIS permet d'attester des dispositifs dans le cadre de services distribués sur le réseau, et met en avant le fait qu'un dispositif dont le firmware n'a pas été modifié pourrait malgré tout réaliser des opérations incorrectes à la suite d'une interaction avec un autre dispositif malveillant. Lors d'une phase d'attestation, le vérificateur interroge l'ensemble des dispositifs impliqués dans une suite d'appels à des services distribués, et vérifie que cette suite d'appels est correcte, et donc légitime.

— "SCARR : A Novel Scalable Runtime Remote Attestation" [39] a été publié en 2018 par Toffalini *et al.* SCARR utilise les techniques de contrôle de flux des programmes et des données pour les appliquer de manière logicielle à des dispositifs plus complexes que les objets connectés, comme par exemple un serveur web. L'implémentation de SCARR utilise notamment un kernel Unix modifié comme racine de confiance, et la détection est instrumentée à l'aide de l'outil de débugage DynamoRIO afin de permettre la réalisations de mesures de contrôle de flux. Cette solution n'étudie pas les attaques physiques, car les auteurs partent du principe que SCARR s'applique principalement à des machines virtuelles.

— En 2018, Conti *et al.* ont publié "SPLIT : A Secure and Scalable RPL routing protocol for Internet of Things" [41]. SPLIT est une proposition de protocole de routage pour les réseaux IoT basé sur le protocole RPL (Routing Protocol for Low-Power and Lossy Networks). Ce protocole est particulièrement adapté aux réseaux à faible consommation et sujets aux pertes de paquets réseaux, et est donc bien adapté à certains réseaux d'objets connectés. SPLIT intègre les mécanismes d'attestation logicielle basés sur les questions-réponses à RPL, ce qui permet une optimisation des découvertes de routes réseau et un mécanisme de mise à l'échelle du réseau tout en vérifiant régulièrement le statut des dispositifs qui prennent part au réseau.

— "Towards more practical software-based attestation" [37], publié par Rodrigo Vieira Steiner et Emil Lupu en 2019, a pour objectif de résoudre une des problématiques que pose l'attestation logicielle, à savoir évaluer précisément le Round-Trip Time (ou RTT) durant l'exécution. Leur protocole repose sur une succession de petits challenges au lieu d'un challenge unique mais plus conséquent. Ceci permet d'améliorer l'observation du RTT et donc la précision dans l'estimation de l'état d'un dispositif. Cette solution permet de diminuer le coût en performance de calcul de l'attestation logicielle, tout en améliorant le niveau de détection de dispositifs compromis.

Les solutions d'attestation à distance logicielles sont ainsi exclusivement des solutions ne demandant aucune modification matérielle. Si ces solutions ont l'avantage de pouvoir être implémentées dans la plupart des appareils, y compris ceux déjà déployés, elles ne peuvent cependant pas prendre convenablement en compte certains types d'attaques, en particulier les attaques physiques.

2.5 Attestation à distance hybride

Les solutions d'attestation à distance hybride sont appelées ainsi car contrairement aux solutions purement logicielles, elles nécessitent des fonctionnalités matérielles, mais ces fonctionnalités restent minimes en comparaison avec les solutions d'attestation à distance purement matérielles. Ces solutions hybrides reposent souvent sur des composants matériels assez basiques comme une combinaison de ROM (Read-Only Memory) et de MPU (Memory Protection Unit), avec la MPU garantissant que le code critique et les données (telles les clés cryptographiques) sont uniquement utilisés de la manière prévue. Ces éléments sont moins coûteux qu'un TPM ou un TEE utilisés dans les solutions matérielles, tout en améliorant le niveau de sécurité en comparaison à une solution uniquement logicielle. Les articles qui traitent d'attestation à distance hybride abordent

soit l'utilisation efficace de ces composants matériels minimaux [26][32][42][43], soit les protocoles d'attestation qui utilisent ces composants [23][24][44][45].
Parmi les articles traitant d'attestation hybride au niveau d'un appareil, certains se concentrent sur l'isolation et la sécurité du code et des données critiques [26][32][46]. D'autres travaillent sur la qualité et l'efficacité de l'attestation, ainsi que sur l'élargissement du périmètre attesté [43][47][48].
Parmi les articles traitant d'attestation hybride au niveau d'un réseau, bon nombre travaillent sur l'efficacité de la communication des attestations entre les appareils du réseau et le *Verifier* [23][24][44][45]. D'autre proposent des solutions adaptées à des types de réseaux spécifiques comme les drones [49][50]. Plusieurs articles [25][51] proposent des mécanismes d'attestation continue, comme les heartbeats, qui permettent de monitorer plus finement un réseau sans augmenter la fréquence d'attestation des appareils. Plus récemment, des auteurs ont proposé des solutions d'attestation basées sur des mécanismes de type *Publish/Subscribe* [28][52].

Portée locale

— La première solution d'attestation hybride à avoir été proposée est SMART [32] (Secure and Minimal Architecture for establishing a dynamic Root of Trust), qui a été publiée en 2012 par El Defrawy *et al.* SMART utilise des éléments matériels bas-coût (tels que la combinaison d'une ROM et d'une MPU) pour stocker le code critique ainsi que les clés, et s'assurer que les clés ne sont lues que par les portions de code autorisées et que le code critique n'est exécuté que dans le but prévu. Ceci permet de réaliser une attestation de manière sécurisée car le code qui la réalise est lui même sécurisé. Les différentes applications, qui elles ne sont pas particulièrement sécurisées, sont alors attestables à tout moment.

— En 2013, Noorman *et al.* ont publié "Sancus : Low-cost trustworthy extensible networked devices with a zero-software Trusted Computing Base" [33]. Sancus est une solution d'attestation qui repose sur des modifications matérielles minimes pour constituer ce qui est appelé la "Trusted Computing Base" (TCB) sous forme d'une extension des microprocesseurs existants. Ces modifications permettent de fournir le mécanisme d'attestation à distance aux appareils équipés, ainsi qu'une bonne intégrité et authenticité de cette attestation en contrôlant les modifications de code et les accès mémoire aux clés cryptographiques. Pour que les logiciels embarqués bénéficient de la TCB de Sancus, les fichiers sources C doivent être annotés, et compilés à l'aide d'un compilateur dédié. Ces logiciels sont délivrés de manière sécurisée par des *Software Providers* (SP) agréés par un *Infrastructure Provider* (IP) à l'aide de clés cryptographiques préalablement échangées.

— "TrustLite, A Security Architecture for Tiny Embedded Devices" [26] a été proposé par Koeberl *et al.* en 2014. Trustlite est une extension de SMART, qui ajoute des règles de contôle d'accès supplémentaires pour la MPU. TrustLite prend en compte différentes sortes de mémoire physique telles que les DRAM et les mémoires Flash, ainsi que des périphériques comme les timers. TrustLite introduit également l'utilisation d'un système d'exploitation et d'états de confiance appelés "Trustlets".

— En 2015, "TyTAN : Tiny Trust Anchor for Tiny Devices" [53] a été publié par Brasser *et al.* TyTAN permet l'isolation des tâches exécutées par le processeur à l'aide d'une architecture matérielle. Ces tâches peuvent provenir de différentes parties et doivent donc être isolées les unes des autres sur la base de l'architecture de sécurité proposée par TyTAN. La sécurité de cette architecture repose sur une racine de confiance dynamique appelée *Root of Trust for Measurement* (RTM), capable d'attester le code de chaque tâche créée, et dont le résultat est garanti par une protection mémoire appelée *Execution-aware Memory Protection Unit* (EA-MPU).

— "Soteria : Offline Software Protection within Low-cost Embedded Devices" [54] a été présenté par Götzfried *et al.* en 2015. Soteria repose sur la base proposée par Sancus, avec un modèle de menaces étendu incluant le contrôle par l'attaquant des composants périphériques et logiciels (y compris l'OS). Soteria propose d'ajouter une protection de ces éléments logiciels y compris en mode offline, en chiffrant le code enregistré en ROM et en le déchiffrant au chargement du module.

— Abera *et al.* ont publié "C-FLAT : Control-Flow Attestation for Embedded Systems Software" [47] en 2016. C-FLAT permet l'attestation du flux de contrôle des applications embarquées sans avoir besoin du code source. Pour ce faire, C-FLAT utilise un outil d'analyse statique qui permet de déterminer les flux valides d'une application en cours d'exécution et les représente sous forme de hachés des chemins du flux formant un graphe, une des difficultés étant de mesurer correctement les conditions et les boucles. La mesure ainsi effectuée permettra de valider un rapport d'attestation envoyé par le module une fois déployé. C-FLAT a été implémenté par ses auteurs sur un Rasperry Pi 2 avec une architecture ARM afin d'en faire la démonstration.

— "LO-FAT : Low-Overhead Control Flow ATtestation in Hardware" [48] a été publié par Dessouky *et al.* en 2017. LO-FAT permet l'attestation du flux de contrôle des applications embarquées sans avoir besoin ni du code source ni instrumenter le code. Cela permet de ne pas ajouter le coût en performance de cette instrumentation, et également d'être utilisé sur des logiciels déjà existants. Comme C-FLAT, LO-FAT réalise des hachés des instructions connues, et a pour objectif

de traiter correctement les conditions et les boucles. En effet, le flux de contrôle de conditions ou de boucles peut amener à de très nombreux états corrects qu'il faut pouvoir valider sans en faire la liste exhaustive. LO-FAT utilise dans ce but un *Branch Filter* et un *Loop Monitor*. Les auteurs ont implémenté LO-FAT sur un microcontrôleur Pulpino basé sur une architecture RISC-V.

— "ATRIUM : Runtime Attestation Resilient Under Memory Attacks" [55] a été publié en 2017 par Zeitouni *et al.* ATRIUM est une solution qui permet d'attester le code d'un appareil ainsi que son exécution même durant une attaque (car l'attestation est réalisée à l'exécution de l'instruction), et qui est particulièrement résistant aux attaques de type *Time Of Check Time of Use* (TOCTOU) existantes sur les travaux précédents comme LO-FAT ou C-FLAT. Pour réaliser cela, ATRIUM mesure les instructions individuelles exécutées en plus de mesurer les flux de contrôle de l'application. Comme LO-FAT, ATRIUM n'instrumente pas le code, minimisant l'impact sur les performance du mécanisme d'attestation.

— En 2017, Noorman *et al.* ont proposé une seconde version de Sancus appelée Sancus 2.0 [56]. Sancus 2.0 repose sur les retours d'expérience issus du premier article (comme Soteria entre autres) afin de proposer une architecture améliorée. Les améliorations proposées permettent notamment le déploiement confidentiel de la solution à l'aide de l'instruction processeur *protect* grâce à laquelle les modules logiciels ne sont déchiffrables qu'à l'aide d'une clé elle-même chiffrée et authentifiée. Ces améliorations reposent également sur des primitives cryptographiques plus efficaces et modernes avec notamment les mécanismes d'*authenticated encryption with associated data* (AEAD). En complément de ces améliorations, cet article décrit également différentes implémentations et applications de Sancus qui ont été réalisées depuis la première publication, comme le déploiement de capteurs de mesures sur une smart grid.

— En 2017, Eldefrawy *et al.* ont publié "HYDRA : HYbrid Design for Remote Attestation (Using a Formally Verified Microkernel)" [27]. HYDRA utilise le micro-kernel vérifié formellement seL4 pour permettre de contrôler l'isolation de la mémoire et les exécutions de code avec des garanties fortes. L'utilisation de ce micro-kernel permet de se passer de composants hardware supplémentaires comme les MPU et ROM utilisés dans SMART et TrustLite. HYDRA a également pour objectif d'être implémentable sur la plupart des cartes disponibles commercialement, les auteurs en ont donc implémenté deux prototypes sur les cartes ODROID-XU4 et Sabre Lite.

— "AAoT : Lightweight Attestation and Authentication of low-resource Things in IoT and CPS" [57] a été publié en 2018 par Feng *et al.* Comme PUFatt, AAoT repose sur l'utilisation d'une *Physically Unclonable Function* (PUF). Contrairement à PUFatt, AAoT ne nécessite pas de modifier le matériel existant car la PUF est directement basée l'unicité et la répétabilité de ce matériel. De plus, AAoT combine l'authentification rendue possible par la PUF à l'attestation software de l'intégrité de l'appareil, permettant ainsi une attestation hybride avec un minimum de modifications sur le matériel existant.

— "Remote Attestation of IoT Devices via SMARM : Shuffled Measurements Against Roving Malware" [42] a été proposé par Carpent *et al.* en 2018. SMARM effectue des mesures de portions de la mémoire de manière aléatoire, et démontre que si un attaquant peut échapper à la détection une fois (par exemple en relocalisant le logiciel malveillant dans la mémoire), il est très improbable qu'il puisse éviter la détection sur une succession de mesures. SMARM détecte ainsi l'attaque malgré différentes techniques d'évasion disponibles, comme l'effacement du logiciel malveillant pour éviter la détection ou la relocalisation dans d'autres zones mémoire en fonction des zones déjà attestées. SMARM reste efficace même en cas d'interruption de son processus, ce qui permet de prioriser le fonctionnement nominal de l'appareil et de minimiser l'impact de l'attestation.

— "ASSURED : Architecture for Secure Software Update of Realistic Embedded Devices" [58] a été publié par Asokan *et al.* en 2018. ASSURED est un framework de mise à jour qui utilise les mécanismes d'attestation pour valider le bon déroulement d'une mise à jour des logiciels. ASSURED étend le mécanisme de mise à jour nommé "The Update Framework" (TUF) en y ajoutant une notion d'autorisation basée sur l'attestation à distance et l'applique en particulier sur HYDRA [27] et TrustZone.

— En 2018, Dessouky *et al.* ont publié "LiteHAX : Lightweight Hardware-Assisted Attestation of Program Execution" [43]. LiteHAX permet de monitorer à la fois le contrôle de flux ainsi que les flux de données des appareils, afin d'être robuste face aux attaques de type *data-oriented programming"* (DOP). Les attaques DOP permettent à l'attaquant d'atteindre ses objectifs tout en respectant le contrôle de flux. Les rapports de l'appareil vers le vérificateur contiennent à la fois le rapport d'attestation du contrôle de flux et du flux de données, les deux étant vérifiés pour identifier une attaque.

— Nunes *et al.* ont publié "Formally Verified Hardware/Software Co-Design for Remote Attestation" [46] en 2018, dans lequel ils construisent l'architecture appelée VRASED (Verifiable Remote Attestation for Simple Embedded Device). VRASED est vérifié formellement, et repose à la fois sur un module matériel de protection

mémoire (HW-Mod) et une implémentation logicielle (SW-Att). HW-Mod est dédié à la protection mémoire et à s'assurer de la bonne exécution de SW-Att. SW-Att réalise les opérations d'attestation à proprement parler en renvoyant un haché de la mémoire correspondant au challenge demandé par le vérificateur.

— En 2020, Ammar *et al.* ont proposé "SIMPLE : A Remote Attestation Approach for Resource-constrained IoT devices" [59]. SIMPLE n'est pas exactement un solution d'attestation hybride, mais a cependant des exigences plus fortes que les solutions d'attestation logicielles. SIMPLE ne nécessite pas de composant matériel supplémentaire car il utilise la mémoire existante, et reste donc peu coûteux tout en prenant en compte un plus grand nombre de dispositifs existants. La sécurité est apportée par l'utilisation d'un hyperviseur d'isolation logicielle de mémoire vérifié formellement, appelée Security Microvisor (SµV). Cet hyperviseur est utilisé pour séparer les logiciels applicatifs qui ne sont pas de confiance d'une partie appelée Trusted Computing Module (TCM) et ainsi permettre une attestation de confiance.

Les solutions d'attestation à distance hybrides à portée locale proposent ainsi des mécanismes permettant aux appareils de réaliser des attestations en utilisant des composants matériels minimaux. Ces composants sont notamment des ROM, des MPU, des PUFs, ou encore des modifications minimales des architectures processeurs existantes. Ces mécanismes permettent à d'autres travaux de proposer des solutions d'attestation de contrôle de flux et de données. Cependant, bien que ces mécanismes constituent une brique essentielle de l'attestation à distance hybride, ils ne répondent pas seuls à la problématique de confiance qui se pose à l'échelle d'un réseau.

Portée réseau

— Publié en 2015 par Asokan *et al.*, "SEDA : Scalable Embedded Device Attestation" [23] a été une des première solutions à définir un protocole d'attestation hybride à destination des flottes d'appareils. L'implémentation de SEDA utilise SMART et TrustLite. Dans SEDA, l'attestation est déclenchée par un vérificateur qui émet la requête auprès d'un premier appareil, qui transmet l'attestation à ses voisins, et ainsi de suite de façon à créer un arbre de communication. Les appareils en bout de branche vont répondre à la demande d'attestation, et à la fin l'appareil initial va recevoir la somme de toutes les réponses et la transmettre au vérificateur. Les réponses peuvent avoir différents niveaux de détail, que ce soit uniquement le nombre d'appareils dans lesquels le réseau a confiance, jusqu'à la liste complète des attestations de tous les appareils. Dans des propositions futures telles DARPA [51] et US-AID [25], SEDA a été critiqué par ses propres auteurs comme n'étant pas une solution adéquate pour une application réelle

car SEDA se concentre uniquement sur des attaques logicielles et sur un seul modèle de réseau, celui en essaim. Cependant, SEDA est utilisé comme base de comparaison pour de nombreuses autres études à cause de son antériorité et de son large champ d'application. SEDA sera notamment utilisé dans le cadre des simulations présentées dans le chapitre 4, car c'est un protocole d'attestation à distance qui s'adapte bien à différents réseaux et est souvent utilisé comme point de comparaison.

— "DARPA : Device Attestation Resilient to Physical Attacks" [51], publié en 2016 par Ibrahim *et al.*, utilise un mécanisme de heartbeat qui envoie périodiquement des messages aux dispositifs voisins afin de s'assurer qu'ils sont encore présents dans le réseau. Ce mécanisme permet la détection d'attaques physiques, car les auteurs considèrent que pour réaliser une attaque physique sur un appareil, l'attaquant doit le déconnecter du réseau pour une durée conséquente (à l'exception d'attaques par canaux auxiliaires).

— En 2016, Ambrosin *et al.* ont publié "SANA : Secure and Scalable Aggregate Network Attestation" [44]. SANA est un protocole d'attestation dans lequel l'attestation est déclenchée par un *Verifier* qui émet une requête à destination d'*Aggregators*, qui vont eux même émettre un requête au reste des appareils. L'attestation de l'ensemble du réseau sera ainsi agrégée jusqu'au *Verifier*. L'avantage notable de SANA et que ce protocole fonctionne également pour des réseaux contenant de nombreux appareils grâce à l'utilisation du *Novel Optimistic Aggregate Signature Scheme*, les simulation réalisées dans Omnet++ indiquant des durées d'exécution pour un million d'appareils ne dépassant pas la douzaine de secondes selon les scénarios étudiés. Un des autres avantages de SANA est que ce protocole est résiliant au fait que des nœuds d'agrégation soient compromis.

— "Lightweight Swarm Attestation : a Tale of Two LISA-s" [49] a été publié par Carpent *et al.* en 2017. Cet article définit une métrique appelée *Quality of Swarm Attestation* (QoSA) qui a pour objectif de comparer l'efficacité de protocoles d'attestation. Cet article propose également deux protocoles d'attestation, LISAα et LISAs, dédiés aux réseaux d'appareils mobiles, et qui ont différents niveaux de QoSA, permettant d'illustrer l'utilisation de cette métrique et donc de valider qu'elle permet une comparaison plus objective des protocoles d'attestation à distance.

— En 2017, Po-Hung Yang et Sung-Ming Yen ont proposé "SARA : Sandwiched attestation through remote agents for cluster-based wireless sensor networks" [60]. SARA cherche à empêcher que, dans le cas de réseaux en clusters, l'entièreté d'un cluster ne soit contrôlé par un attaquant. Pour ce faire, SARA utilise des dispositifs équipés de TPM afin d'isoler et de sécuriser les sous-parties du réseau global. SARA vise à attester les têtes de clusters de deux manières combinées :

les nœuds qui font partie du cluster vérifie régulièrement la tête de leur cluster (approche bottom-up) et en cas de suspicion de compromission, les stations de base doivent vérifier la tête de cluster incriminée (approche top-down). SARA permet ainsi aux dispositifs qui sont à la tête de clusters d'agir comme des pare-feux réseau, tout en étant lui-même attesté régulièrement, le tout avec un coût en stockage mémoire des attestations faible ainsi qu'une augmentation mineure de la consommation d'énergie liée à la communication et au traitement des messages.

— "SeED : Secure Non-Interactive Attestation for Embedded Devices" [61] a été publié par Ibrahim *et al.* en 2017. SeED propose un protocole d'attestation non-interactif, qui est déclenché à intervalle régulier par le *Prover*, et simplement reçu par le *Verifier*. Ceci permet de diminuer l'impact du protocole sur les communications et la consommation d'énergie. Ceci permet également d'empêcher les attaques par déni de service sur le *Prover* en utilisant les requêtes d'attestation, qui sont généralement authentifiées et donc nécessitent de la puissance de calcul. Ainsi en comparaison à SEDA, SeED économise ainsi environ 50% d'énergie dans le cas d'implémentations réalisées sur TelosB et MICAz.

— Kohnhäuser *et al.* ont proposé "SCAPI : A Scalable Attestation Protocol to Detect Software and Physical Attacks" [62] en 2017. SCAPI permet d'attester un réseau à l'aide d'un nombre minimal de messages, réduisant fortement l'impact du protocole d'attestation en comparaison à d'autres protocoles, et permet de maintenir l'attestation tant que moins de la moitié du réseau est compromise. Le mécanisme principal de SCAPI qui permet d'exclure les appareils correctement est la phase de mise à jour de clé de session (*Session Key Update Phase*). Durant cette phase, une clé de session est générée et partagée dans tout le réseau, et doit être utilisée par tous les appareils durant les prochains échanges d'attestation. Ainsi, un appareil physiquement compromis, qui a été déconnecté pendant une durée trop longue, ne possédera pas cette clé et ne pourra plus interagir avec le réseau. La robustesse et les performances de SCAPI sont démontrées à l'aide de simulation d'un réseau maillé dans Omnet++.

— "ERASMUS : Efficient Remote Attestation via Self-Measurement for Unattended Settings" [63] a été proposé en 2018 par Carpent *et al.* ERASMUS est un protocole dans lequel les appareils réalisent eux-même des attestations de manière périodique, comme SeED. Ces attestations sont ensuite récupérées par le *Verifier* plus tard, et servent à vérifier l'historique de l'appareil. Ceci permet en particulier d'éviter que les attestations, si elles étaient déclenchées par le *Verifier*, ne viennent perturber les tâches principales des appareils. Les mesures sont stockées dans la mémoire du *Prover* en attendant d'être récupérées par le *Verifier*, et ne sont pas particulièrement protégées : elles peuvent être modifiées ou supprimées par un

attaquant, mais cela conduirait à une détection de l'attaque. Ces mesures sont en revanche basées sur un MAC, et la clé utilisée est, elle, sécurisée. Un attaquant ne peut donc pas forger de toutes pièces une mesure correcte. Un autre avantage d'ERASMUS est la flexibilité dans la fréquence de mesure, qui permet de détecter du code malveillant qui ne serait pas persistant.

— "US-AID : Unattended Scalable Attestation of IoT Devices" [25] a été proposé par Ibrahim *et al.* en 2018 et est une solution d'attestation à distance axée sur les réseaux dynamiques autonomes, comme un essaim de drones par exemple. Un des apport principaux d'US-AID est l'utilisation de PONAs (Proofs Of Non-Absence, ou Preuves de Non-Absence), similaires aux heartbeats de DARPA : ces PONAs servent à montrer que les dispositifs sont toujours connectés et n'ont pas été compromis par une attaque physique. Utiliser les PONAs permet aux dispositifs de se déplacer et de modifier la configuration du réseau à condition que leur futur voisin soit directement à côté d'un de leurs précédents voisins. Les PONAs sont conservées et permettent également d'avoir un historique de l'état des voisins d'un appareil. US-AID sera utilisé comme protocole de comparaison dans le cadre des simulations présentées dans le chapitre 4, notamment pour son utilisation d'un mécanisme de heartbeat (les PONAs) et son bon fonctionnement dans le cas des réseaux d'appareils mobiles.

— "SALAD : Secure and Lightweight Attestation of Highly Dynamic and Disruptive Networks" [24] a été proposé par Kohnhäuser *et al.* en 2018. SALAD est un protocole d'attestation dans lequel l'attestation est réalisée localement de voisin à voisin, et agrégée ensuite. L'agrégation se fait au fur et à mesure, de telle sorte qu'à la fin l'ensemble des appareils du réseau possède le même niveau d'information sur l'ensemble des autres appareils. Ceci permet à un *Verifier* de récupérer l'attestation depuis n'importe quel appareil du réseau, et permet également d'éviter qu'un lien de communication défaillant ne viennent faire échouer l'attestation du réseau. Cette aggrégation est réalisée sur la base d'un mécanisme d'agrégation existant dont les auteurs réalisent une implémentation nommée MACSimple. Ils proposent ensuite deux extensions de ce mécanisme, MACSmart et MACGreedy. MACGreedy sert à minimiser l'attestation stockée localement en agrégeant les attestations le plus tôt possible. MACSmart au contraire sert à minimiser le volume de données transmises. Bien que MACSmart soit très efficace, les auteurs montrent que, pour des scénarios réalistes, accepter une diminution de la sécurité théorique en tronquant simplement MACSimple semble plus efficace.

— En 2018, "PADS : Practical Attestation for Highly Dynamic Swarm Topologies" [45] a été publié par Ambrosin *et al.* PADS est un protocole d'attestation qui permet d'attester des réseaux comprenant de nombreux appareils et qui ne sont pas

particulièrement structurés à l'aide d'un mécanisme de consensus. Les différents *Provers* du réseau s'attestent eux-mêmes et partagent régulièrement leur attestation et la connaissance qu'ils ont des autres appareils (Bonne santé, Compromis, ou État inconnu). Le mécanisme de consensus consiste à conserver l'état le plus significatif possible connu d'un appareil à l'arrivée de nouvelles informations avant de le retransmettre (ainsi, un appareil restera en État inconnu tant que son attestation n'a pas été reçue, et restera Compromis si un seul autre appareil le reporte en tant que tel). Dans son implémentation, PADS repose sur SeED qui est un autre protocole non-interactif, et est comparé à SANA en ce qui concerne le temps d'exécution par nombre d'appareils dans le réseau car c'est un protocole similaire.

— En 2018, "MTRA : Multi-Tier randomized Remote Attestation in IoT Networks" [64] a été publié par Tan *et al.* MTRA est un protocole d'attestation qui vise notamment les réseaux hétérogènes composés d'un côté d'appareils équipés de TPM, et donc considérés très sécurisés, et d'un autre côté d'appareils sans sécurité matérielle. L'attestation des appareils sécurisés repose alors sur la sécurité matérielle des TPM, tandis que les autres appareils sont attestés à l'aide d'un protocole challenge-réponse qui est basé sur une chaîne de hachés, permettant une défense efficace contre différentes attaques. La double approche apportée par MTRA permet ainsi de considérer des réseaux hétérogènes, plus similaires à ceux existant réellement.

— "SARA : Secure Asynchronous Remote Attestation for IoT Systems" a été publié en 2020 par Dushku *et al.* SARA permet une attestation asynchrone des différents appareils du réseau en se basant sur un mécanisme *publish/suscribe*. Les différents services des appareils vont agir comme *publisher*, *suscriber* ou les deux. Lors d'une phase d'attestation, les publisher ayant reçu une requête d'attestation vont diffuser leur attestation à leurs subscribers, qui peuvent ensuite agir également comme publishers à leur tour, ou répondre à la requête d'un *verifier* qui viendrait récupérer les attestations. Ceci permet de rendre l'attestation asynchrone en se reposant sur les protocoles de communication *publish/subscribe* existants, tel MQTT.

— Helble *et al.* (2021) [65] font la promotion de mécanismes d'attestation flexibles, ce qui est plus proche d'une proposition de framework que des protocoles d'attestation à distance précédemment cités. Helble *et al.* proposent des fonctionnalités comme la flexibilité dans le choix du protocole utilisé, ainsi qu'un mécanisme de politique de négociation des protocoles d'attestation. Pour ce faire, ils utilisent Copland de Ramsdell *et al.* [66], un langage de spécification dédié, afin de proposer plusieurs exemples de protocoles d'attestation.

— En 2021, Halldorson *et al.* ont publié "ARCADIS : Asynchronous Remote Control-Flow Attestation of Distributed IoT Services" [67]. ARCADIS permet l'attestation asynchrone d'appareil sur un modèle similaire à celui de SARA (2020), mais sur la base d'attestation dynamique de flux de contrôle, comme peuvent par exemple le proposer C-FLAT ou LO-FAT. Cela permet de combiner les avantages de ces deux mécanismes en terme de sécurité. ARCADIS présente en revanche l'inconvénient de perdre en performance rapidement avec l'augmentation du nombre d'appareils, d'autant plus dans le cas d'appareils ayant une large surface (code, données) à attester.

— Gaurang Bansal et Biplab Sikdar ont proposé "S-MAPS : Scalable Mutual Authentication Protocol for Dynamic UAV Swarms" [50] en 2021. S-MAPS permet l'attestation d'essaims de drones de tailles variées. Le mécanisme proposé repose sur une PUF pour authentifier les appareils vis-à-vis de la station de base, afin de garantir l'intégrité physique des appareils. L'attestation suit un arbre de communication de nœud à nœud, suivant un mécanisme similaire à celui de SEDA, ce qui permet dans ce cas au protocole de fonctionner malgré le dynamisme de la topologie réseau, et ce pour des grands nombres d'appareils.

— En 2022, Petzi *et al.* ont publié "SCRAPS : Scalable Collective Remote Attestation for Pub-Sub IoT Networks with Untrusted Proxy Verifier" [52]. SCRAPS permet, en utilisant la blockchain comme procuration, de réaliser une attestation *many-to-many* entre des *Verifiers* et des *Provers*. Le mécanisme d'attestation suit une logique *publish/suscribe* : les *Provers* publient leur attestation dans la blockchain, tandis que les *Verifiers* font office de *subscribers* et reçoivent les attestations. SCRAPS permet ainsi la coexistence de plusieurs *Verifiers*, et son caractère asynchrone permet la récupération des attestation à la demande, tout en minimisant la consommation énergétique des appareils.

Les solutions d'attestation à distance hybride à portée réseau consistent ainsi généralement à permettre la diffusion des attestations individuelles des appareils au sein d'un réseau, que ce soit à un *Verifier* unique, à plusieurs voisins, ou à l'ensemble du réseau. Bien que les solutions proposées soient variées, elles visent toutes à permettre cette diffusion de façon efficace tout en répondant aux contraintes posées par les réseaux d'objets connectés. En revanche, l'inconvénient de ces solutions est qu'elles sont spécifiques à des types de réseaux, d'appareils, ou répondent à une problématique trop précise : une solution idéale permettrait d'utiliser plusieurs de ces mécanismes de façon conjointe.

2.6 Conclusion du chapitre

Parmi les différentes solutions d'attestation à distance se distinguent donc différents mécanismes, qui partagent le même objectif d'amélioration de la sécurité des objets connectés et des réseaux auxquels ils participent. Ces différents mécanismes sont souvent amenés à coexister au sein d'un réseau, dans lequel les appareils sont hétérogènes, et pour lesquels les solutions logicielles, matérielles ou hybrides seront plus ou moins adaptées.
La proposition d'établir un framework d'attestation continue se base sur ce constat. En effet, ces nombreuses solutions existantes peuvent alors réellement être intégrées de façon agnostique dans une solution plus globale, paramétrable, et qui peut facilement être adaptée selon la réalité du contexte de déploiement du réseau, tout en y intégrant la notion de continuité par défaut afin de prendre systématiquement en compte les attaques physiques.
Parmi les articles présentées, deux seront principalement utilisés comme point de comparaison : SEDA [23] et US-AID [25]. Ces deux protocoles sont hybrides à portée réseau, catégorie à laquelle appartient nécessairement un framework généraliste. De plus, SEDA a grandement influencé les autres travaux sur l'attestation à distance, et est un point de comparaison souvent utilisé dans les travaux du domaine. De son côté, US-AID présente l'intérêt d'être particulièrement adapté aux réseaux mobiles, et de proposer un mécanisme de heartbeat. De plus, SEDA comme US-AID reposent déjà sur des racines de confiance matérielles telles que proposées dans SMART [32] ou TrustLite [26].

3

Un framework d'attestation à distance, continue, agnostique et adaptable au contexte

Sommaire

3.1 Introduction du chapitre

Dans ce chapitre, la proposition de framework d'attestation continue à distance est proposée, sur la base d'une définition de ce qu'est un tel framework, et de la définition générale d'un réseau.
La section 3.2 présente ainsi la définition d'un framework d'attestation continue à distance, avec notamment des exigences fonctionnelles et de sécurité en ce qui concerne le mécanisme d'attestation.
La section 3.3 donne la définition générale d'un réseau, afin d'appuyer l'aspect universel du framework proposé.
Enfin, la section 3.4 contient les détails de CRAFT, le premier framework d'attestation continue à distance. Les différents messages relatifs à l'attestation utilisés dans le cycle de vie d'un appareil y sont présentés, avec notamment les heartbeats qui permettent d'assurer l'aspect continu de l'attestation. Des fonctionnalités de CRAFT plus avancées, les fonctionnalités ASMP (Adaptive Simultaneous Multi-Protocols), sont également présentées. Ce sont elles qui permettent aux appareils l'usage adaptatif en fonction du contexte de multiples protocoles d'attestation.

3.2 Définition et exigences d'un framework d'attestation continue à distance

Un framework d'attestation continue vient en complément des protocoles d'attestation existants présentés dans le chapitre précédent. En effet, ces protocoles répondent souvent à des problématiques d'attestation spécifiques, comme l'attestation d'un certain

type de réseau (plus précisément des différents appareils qui le composent) ou celle d'un appareil en particulier (par exemple en utilisant du matériel spécifique comme un TEE ou une MCU). Un framework est agnostique des problématiques spécifiques au réseau et permet d'utiliser de multiples protocoles d'attestation afin de répondre à un grand nombre de cas d'usage avec une même spécification.
Ainsi, l'utilisation d'un framework d'attestation continue est plus flexible que celle d'un protocole unique. Les frameworks permettent également d'apporter plus de sécurité au réseau en intégrant par défaut des fonctionnalités telles que les heartbeats présentés en section 3.4, qui renforcent grandement la continuité de l'attestation tout en ayant un impact bien moindre sur les performances. La force des frameworks d'attestation vient également du fait qu'ils peuvent être adaptés aux spécificités (par exemple les contraintes en puissance de calcul ou en consommation d'énergie) de chaque réseau à l'aide d'une simple configuration basée sur des paramètres de fonctionnement, qui peuvent même évoluer en fonction du contexte.
Un framework d'attestation continue doit répondre à un certain nombre d'exigences, qui se trouvent listées ci-dessous. Ces exigences se divisent en deux catégories : les exigences fonctionnelles et les exigences de sécurité. Les exigences fonctionnelles décrivent les fonctionnalités de base qu'un framework d'attestation continue doit inclure afin d'assurer efficacement le mécanisme d'attestation. Les exigences de sécurité définissent quant à elles les fonctionnalités de sécurité qui doivent être implémentées pour maintenir la confiance que les appareils portent en leurs voisins, et par extension pour garantir la sécurité du réseau.

Exigences fonctionnelles

- **(EF1)** Un framework d'attestation continue doit pouvoir supporter n'importe quel protocole d'attestation (à condition que le protocole puisse être utilisé dans le contexte de déploiement visé).
- **(EF2)** Un framework d'attestation continue doit pouvoir adapter ses paramètres de configuration en fonction du contexte de déploiement visé, ledit contexte incluant notamment la mobilité du réseau, le niveau de sécurité intrinsèque des objets connectés, les performances matérielles des objets connectés, ainsi que le modèle de menaces spécifique au contexte de déploiement.
- **(EF3)** Un framework d'attestation continue doit réaliser aussi peu d'échanges de données que possible afin de minimiser l'utilisation du réseau et la consommation d'énergie, tout en atteignant le niveau de sécurité requis dans le cadre du déploiement.

— **(EF4)** Un framework d'attestation continue doit autant que possible minimiser les appels à des primitives cryptographiques afin de réduire le temps de calcul et la consommation d'énergie, tout en atteignant le niveau de sécurité requis dans le cadre du déploiement.

— **(EF5)** Un framework d'attestation continue doit permettre l'ajout futur d'extensions, afin de pouvoir répondre à des problématiques réseau ou de sécurité spécifiques. C'est notamment le cas des fonctionnalités ASMP qui viennent compléter la version initiale de CRAFT, présentées en section 3.4.3.

Exigences de sécurité

— **(ES1)** Un framework d'attestation continue doit rejeter tout appareil malveillant qui tente de faire partie du réseau.

— **(ES2)** Un framework d'attestation continue doit être capable d'exclure un appareil du réseau lorsque l'attestation n'est pas maintenue en continu, c'est-à-dire lorsque l'appareil ne répond pas correctement à une demande d'attestation ou lorsque l'appareil ne peut pas être contacté pendant un certain temps.

3.3 Définition générale des réseaux et cas d'usage

Dans cette section, une définition générale des réseaux est donnée en section 3.3.1. Cette définition permet d'avoir une description commune d'un réseau, et sert de base à l'explication du framework CRAFT. Des exemples plus spécifiques de cas d'usages basés sur cette définition sont ensuite fournis en 3.3.2, permettant de visualiser concrètement les applications de cette définition.

3.3.1 Définition générale d'un réseau

Les réseaux considérés sont composés de manière générale de deux catégories d'appareils ainsi qu'illustré figure 3.1. Ces deux catégories dépendent du niveau de sécurité intrinsèque des appareils, et sont les suivants : les appareils de catégorie K bénéficiant d'un bon niveau de sécurité qui appartiennent au *Coeur de réseau*, et les appareils de catégorie L moins sécurisés qui appartiennent au *Réseau externe*. Tous les appareils peuvent se connecter au réseau et communiquer avec leurs voisins. Ils peuvent également être mobiles et donc changer de voisins. Les appareils de catégorie K comme de catégorie L peuvent également être bannis du réseau si la confiance envers eux n'est plus suffisante, en particulier s'ils échouent lors d'une phase d'attestation ou s'ils ne répondent plus aux requêtes pendant une trop longue période

(à définir en fonction du contexte). Bannir les appareils dans lesquels le réseau (c'est-à-dire les autres appareils) n'a pas une confiance suffisante permet d'éviter, si ces appareils sont compromis, que le réseau entier ne devienne compromis à son tour.

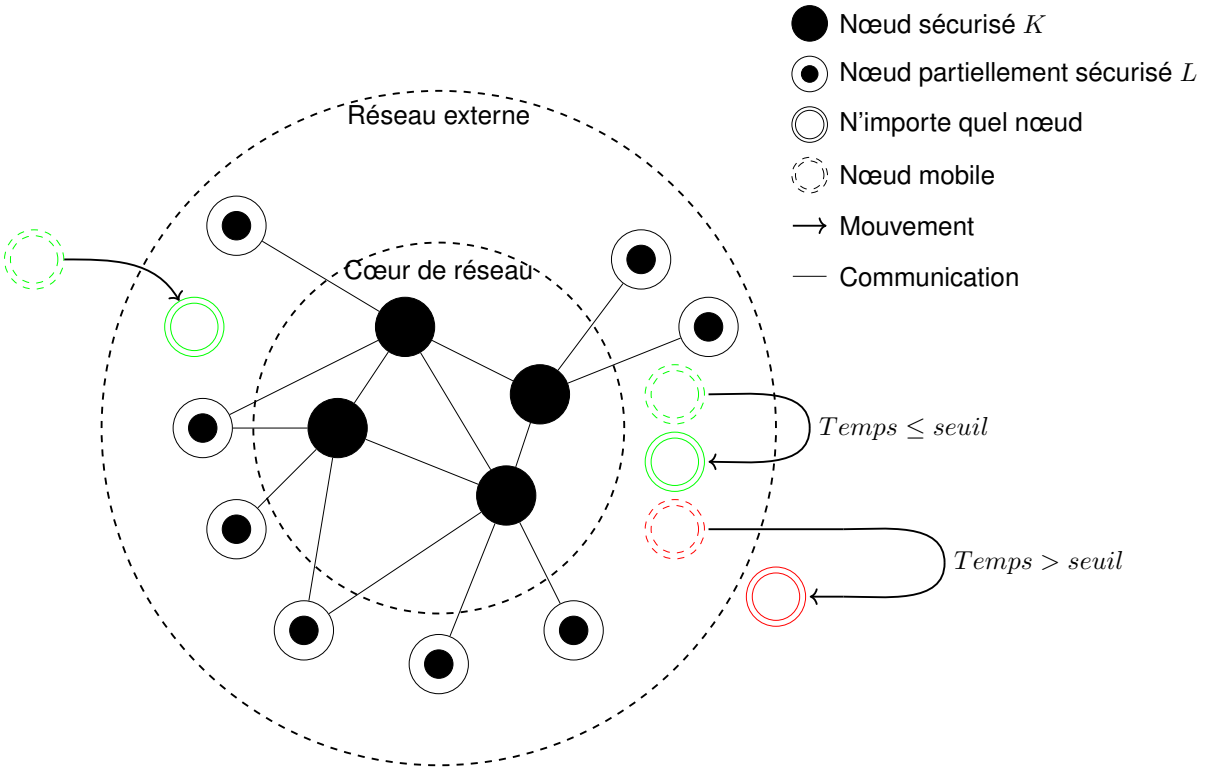


FIGURE 3.1 – Schéma général d'un réseau

3.3.2 Cas d'usage

La définition générale d'un réseau peut s'appliquer à divers cas d'usage, dont certains sont présentés ici. Ces exemples s'inspirent des cas d'usages rencontrés dans l'entreprise Icohup qui développe des capteurs de radioactivité connectés et au sein de laquelle cette thèse a été réalisée. Nous nous intéressons en particulier à trois cas d'usage :

Flotte de drones

Dans le cadre de leurs interventions, les forces de police ou les pompiers peuvent vouloir réaliser une levée de doute sur la présence de matériel radioactif dans la zone d'intervention. Pour ce faire, ils sont équipés d'une flotte de plusieurs drones en pilotage automatique qui vont quadriller la zone, ainsi que d'un ou de plusieurs drones pilotés manuellement. Ces drones sont équipés de capteurs de radioactivité connectés, et communiquent entre eux, en temps réel, les informations de mesure

et de vol, et communiquent également avec une base d'affichage de données au sol. Afin de s'assurer de l'authenticité de ces données, les drones utilisent un framework d'attestation continue, qui permet de vérifier à tout instant que les drones n'ont pas été compromis, et donc que les informations concernant la radioactivité et le plan de vol sont correctes. Dans ce cas d'usage, l'opérateur O chargé de la mise en place initiale du système peut configurer la base d'affichage au sol comme un nœud K car elle est moins limitée en ressources et en autonomie qu'un drone. La majorité des drones sont des nœuds L et communiquent régulièrement en direct avec la base pour assurer le processus d'attestation continue. Dans le cas où la mission nécessiterait une distance d'intervention trop longue, des drones dotés de meilleures puissances de calculs et pouvant faire office de nœuds K sont déployés. Ce sont eux qui agrègent alors les attestations des nœuds L tout en s'attestant mutuellement, formant ainsi un réseau maillé autonome.

Ville intelligente

Avec pour perspective les Jeux Olympiques de Paris 2024, les sites sportifs tels que les stades ou le village olympique s'équipent de solutions permettant la détection de radioactivité, afin de surveiller le risque terroriste ainsi que pour pouvoir agir en cas de crise. Différents types de portiques de détection sont installés aux accès du site, pour contrôler les véhicules, les foules, ainsi que les personnes. Des agents sont équipés de capteurs individuels pour des contrôles spécifiques. L'ensemble de ces capteurs est connecté sur le même réseau, et remonte les données au niveau de la sécurité du site, mais également au niveau des organisateurs et des services de l'État. Pour s'assurer du bon fonctionnement de l'ensemble des capteurs du réseau et de la fiabilité des données qu'ils transmettent, ceux-ci utilisent un framework d'attestation continue. L'opérateur O va déployer l'ensemble de ce système de surveillance. Les capteurs légers portés par du personnel sont des nœuds L car ayant de faibles puissances de calcul. Les portiques de détection possèdent un bon niveau de sécurité, mais étant placés au milieu de la foule et étant donné que leur accès est difficile à contrôler, l'opérateur O les désigne également comme des nœuds L. En revanche, les différents appareils situés au niveau du centre de sécurité sont des nœuds K, et agrègent l'attestation des autres appareils en plus des données de mesure.

Clinique connectée

Dans les cliniques et hôpitaux, certains soins réalisés nécessitent l'utilisation de la radioactivité, et des mécanismes de radioprotection appropriés doivent être mis en

place pour éviter la surexposition du personnel soignant. Ainsi, des capteurs de radioactivité connectés sont placés dans les salles où ces interventions médicales sont réalisées, et ils sont reliés à un système de signalisation lumineuse à l'entrée de ces salles. Certains soignants sont également équipés de capteurs portables, dit dosimètres opérationnels, afin de surveiller de façon plus fine leur exposition. Tous ces capteurs prennent part au même réseau, et leurs données sont centralisées à destination de la Personne Compétente en Radioprotection (PCR) de l'établissement. L'ensemble de ces appareils utilisent un framework d'attestation continue pour garantir leur fiabilité et donc la qualité de la surveillance mise en place. L'opérateur O configure les capteurs fixes dans les salles comme étant des nœuds K, leur niveau de sécurité étant jugé suffisant au regard du modèle de menace du contexte. Ces nœuds récupèrent l'attestation des appareils de signalisation lumineuse qui leur sont directement connectés, qui est quant à elle un nœud L. Les capteurs dont le personnel est équipé sont également des nœuds L, étant à la fois peu sécurisés et facilement accessibles. L'appareil récupérant les données de mesure à destination de la PCR est un nœud K qui récupère l'ensemble des autres attestations.

3.4 Le framework CRAFT : le premier framework d'attestation continue à distance

Dans cette section, le framework CRAFT est détaillé. Dans un premier temps, les fonctionnalités de base et les phases de CRAFT sont présentées en sections 3.4.1 et 3.4.2. Des fonctionnalités plus avancées de CRAFT permettant d'utiliser plusieurs protocoles d'attestation simultanément sont décrites en section 3.4.3. Enfin, une analyse de sécurité est détaillée en section 3.4.4.

3.4.1 Présentation générale de CRAFT

Notations

Les notations présentées dans la table 3.1 sont utilisées dans l'ensemble de ce travail pour décrire CRAFT. Cette table se compose de quatre parties : les *définitions générales*, les *paramètres de nœuds*, les *paramètres de réseau* et les *fonctions*.
Les *définitions générales* établissent la base nécessaire aux autres notations. Les *paramètres de nœuds* sont spécifiques à chaque nœud et sont soit des paramètres de configuration soit des états internes. Les *paramètres de réseau* sont utilisés pour établir le réseau et organiser les nœuds. Enfin, les *fonctions* permettent une meilleure description des algorithmes utilisés.

TABLE 3.1 – Notations

Définitions générales	
O	Opérateur du réseau
N	Nombre total de nœuds
D	Ensemble de tous les nœuds tels $D = \{D_i; 1 \leq i \leq N\}$
Paramètres des nœuds	
id_i	Identité d'un nœud D_i
s_i	Niveau de sécurité d'un nœud D_i
h_i	Mobilité maximale d'un nœud. Un nœud D_i situé à une distance réseau h_i de D_j peut se déplacer physiquement à proximité de $D_{j \neq i}$ sans communication supplémentaire entre D_i et D_j. Si $h_i = 0 \; \forall i$, alors les positions relatives des nœuds sont fixes.
T_{a_i}	Durée sans heartbeat échangé après laquelle D_i est considéré compromis
T_{b_i}	Durée maximale entre deux heartbeats envoyés par D_i après laquelle un message défini en section 3.4.2 est diffusé jusqu'au h_i ème voisin, avec $T_{b_i} < T_{a_i}$
SK_i, PK_i	Clés privée et publique de D_i
k_{ij}	Clé partagée par deux nœuds (par exemple D_i et D_j)
$OCert_i$	Certificat de D_i délivré par O
H_i^t	Heartbeat du nœud D_i reçu au temps t par $D_{j \neq i}$
H_i^{list}	Liste des heartbeats de $D_{j \neq i}$ connus par D_i
P_i	Paramètres de D_i, $P_i = \{id_i, s_i, h_i, T_{a_i}, T_{b_i}\}$
$Sign_O(P_i, PK_i)$	Signature par O des paramètres et de la clé publique de D_i
Paramètres de réseau	
st_L	Niveau de sécurité minimal pour lequel un nœud peut participer au réseau
st_K	Niveau de sécurité minimal au-dessus duquel un nœud est considéré sécurisé
K	Ensemble des nœuds sécurisés avec $K = \{D_i \,\vert\, st_K \leq s_i; 1 \leq i \leq N\}$
L	Ensemble des nœuds partiellement sécurisés avec $L = \{D_i \,\vert\, st_L \leq s_i < st_K; 1 \leq i \leq N\}$
Fonctions	
$Auth(input, key)$	Authentifie une entrée $input$ donnée à l'aide d'une clé key donnée
$Verify(auth, key)$	Vérifie une authentification $auth$ donnée à l'aide d'une clé key donnée
$Exist(value, list)$	Vérifie qu'une valeur $value$ donnée existe dans une liste $list$ donnée
$Hash(input)$	Calcule le haché d'une entrée $input$ donnée en utilisant des fonctions de hachage (par exemple SHA1, SHA-256, ...)
$Enc(input, key)$	Chiffre une entrée $input$ donnée à l'aide d'une clé key donnée
$Dec(input, key)$	Déchiffre une entrée $input$ donnée à l'aide d'une clé key donnée

Description

L'objectif de CRAFT est d'améliorer la sécurité des réseaux IoT en rendant possible l'attestation continue des nœuds du réseau. Les réseaux sont contrôlés par un opérateur O et sont composés de N appareils D_i, chacun identifié par un identifiant id_i.
Chaque appareil est défini par plusieurs paramètres afin de rendre le framework fonctionnel pour n'importe quelle configuration réseau. Les nœuds sont séparés en trois sous-catégories en fonction de leur niveau de sécurité s_i. Ce paramètre est fonction du contexte de déploiement et des caractéristiques de l'appareil, et est assigné par O à l'initialisation du réseau. Les nœuds sont rangés dans l'une des trois catégories en accord avec les seuils st_L et st_K, également fixés en fonction du contexte de déploiement. Les nœuds sécurisés avec $st_K \leq s_i$ sont des nœuds K et font partie du *Cœur de réseau*, tandis que les nœuds partiellement sécurisés avec $st_L \leq s_i < st_K$ sont des nœuds L et font partie du *Réseau externe*. Les autres appareils n'appartenant pas à ces deux catégories sont exclus du réseau.
Moins un appareil est sécurisé, plus ses autres paramètres de sécurité doivent être stricts. Par exemple, les délais maximaux entre deux heartbeats T_{a_i} et T_{b_i} ou la mobilité maximale h_i auront des valeurs plus restrictives (plus ces valeurs sont faibles, moins un appareil a de liberté). Une fois ces paramètres fixés, les appareils sont configurés par O avec $P_i = \{id_i, s_i, h_i, T_{a_i}, T_{b_i}\}$. Chaque nœud D_i possède également une paire de clés (SK_i, PK_i) pour initialiser une connexion sécurisée avec un autre nœud D_j en créant une clé de session k_{ij}. Pendant l'étape de *setup*, O signe également chaque ensemble de paramètres des appareils à l'aide de $Sign_O(P_i, PK_i)$. Ceci permet aux appareils de montrer qu'ils sont authentiques lors de leur première connexion avec d'autres appareils, car ils peuvent effectuer une vérification de la signature à l'aide de la clé publique de O.

3.4.2 Phases de CRAFT

Durant le cycle de vie du réseau tel qu'illustré figure 3.2, le framework d'attestation continue à distance comprend deux phases (*Phase Hors-ligne* et *Phase En-ligne*), qui se composent chacune de différentes étapes. Ces étapes sont illustrées en détail figure 3.3, sur laquelle elles sont représentées du point de vue d'un appareil au sein de CRAFT. Chacune des phases et étapes est détaillée dans le reste de cette section à partir de la vue d'ensemble fournie par le paragraphe suivant.
Sur ces deux figures, on observe notamment la configuration de l'appareil par l'opérateur O et son insertion dans le réseau (étapes *setup* et *init*) durant la *Phase Hors-ligne*. La *Phase Hors-ligne* démarre par une étape de *setup* durant laquelle les appareils sont catégorisés et leurs paramètres sélectionnés. Ensuite, une étape d'*init* a lieu

durant laquelle les paramètres précédemment choisis sont envoyés individuellement à l'ensemble des appareils par l'opérateur O.
À son arrivée dans le réseau, au début de la phase *Phase En-ligne*, l'appareil envoie un message `connect` à ses voisins. Ceux-ci lui répondent également par un message `connect`. Différents événements (l'arrivée d'un message ou l'expiration d'un timer) peuvent alors avoir lieu auxquels l'appareil va réagir, comme le montre la figure 3.3. Selon ses paramètres, l'appareil va envoyer des messages `beat` et `attest` à intervalles réguliers. Si il s'agit d'un nœud K, il va également envoyer un `lost` lorsqu'un de ses voisins ne communique plus depuis trop de temps. L'appareil va également recevoir différents messages dans le cadre du processus d'attestation, soit en réponse à ses propres messages, soit dans le but d'y répondre lui-même. Enfin, si l'appareil se déplace, il devra envoyer un `connect` à ses nouveaux voisins avant de reprendre son cycle d'attestation.

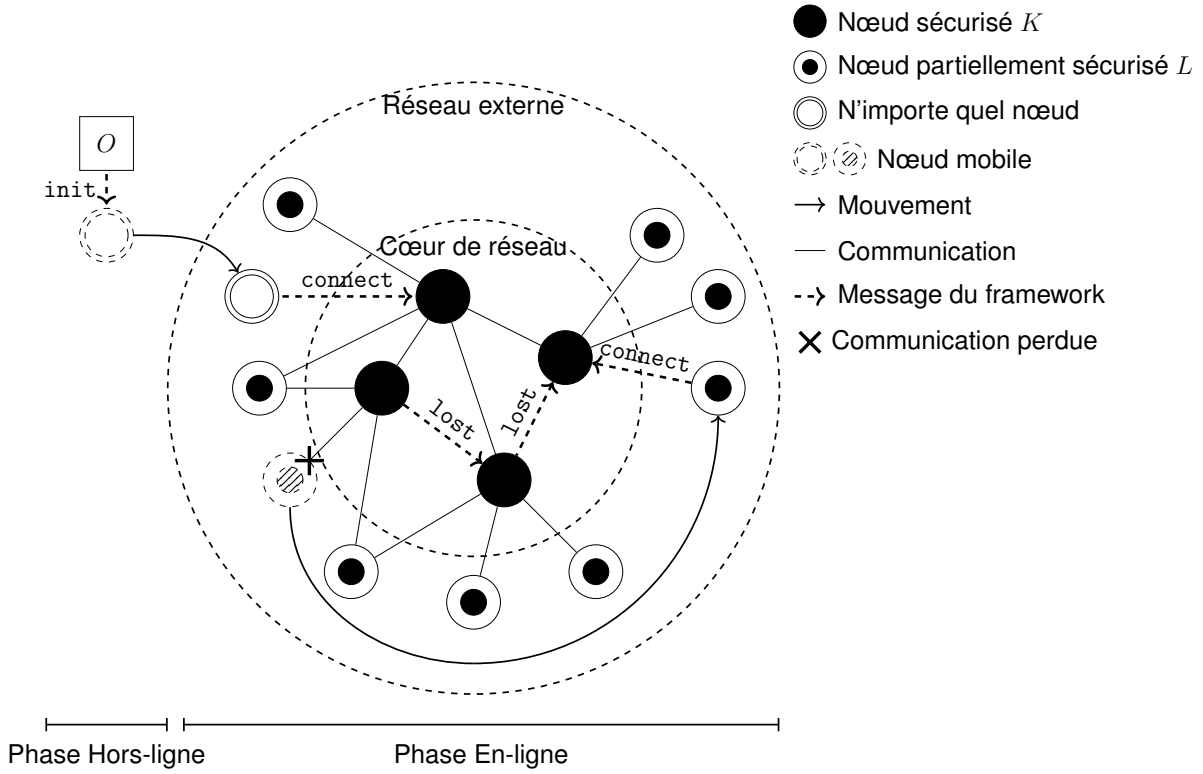


FIGURE 3.2 – Phases principales du cycle de vie de CRAFT

Phase Hors-ligne

Setup L'étape de *setup* consiste à catégoriser un appareil avant qu'il ne soit introduit dans le réseau. Comme présenté dans la section 3.3, deux classes d'appareils sont considérées, appelées K et L. La limite entre ces deux classes est choisie par

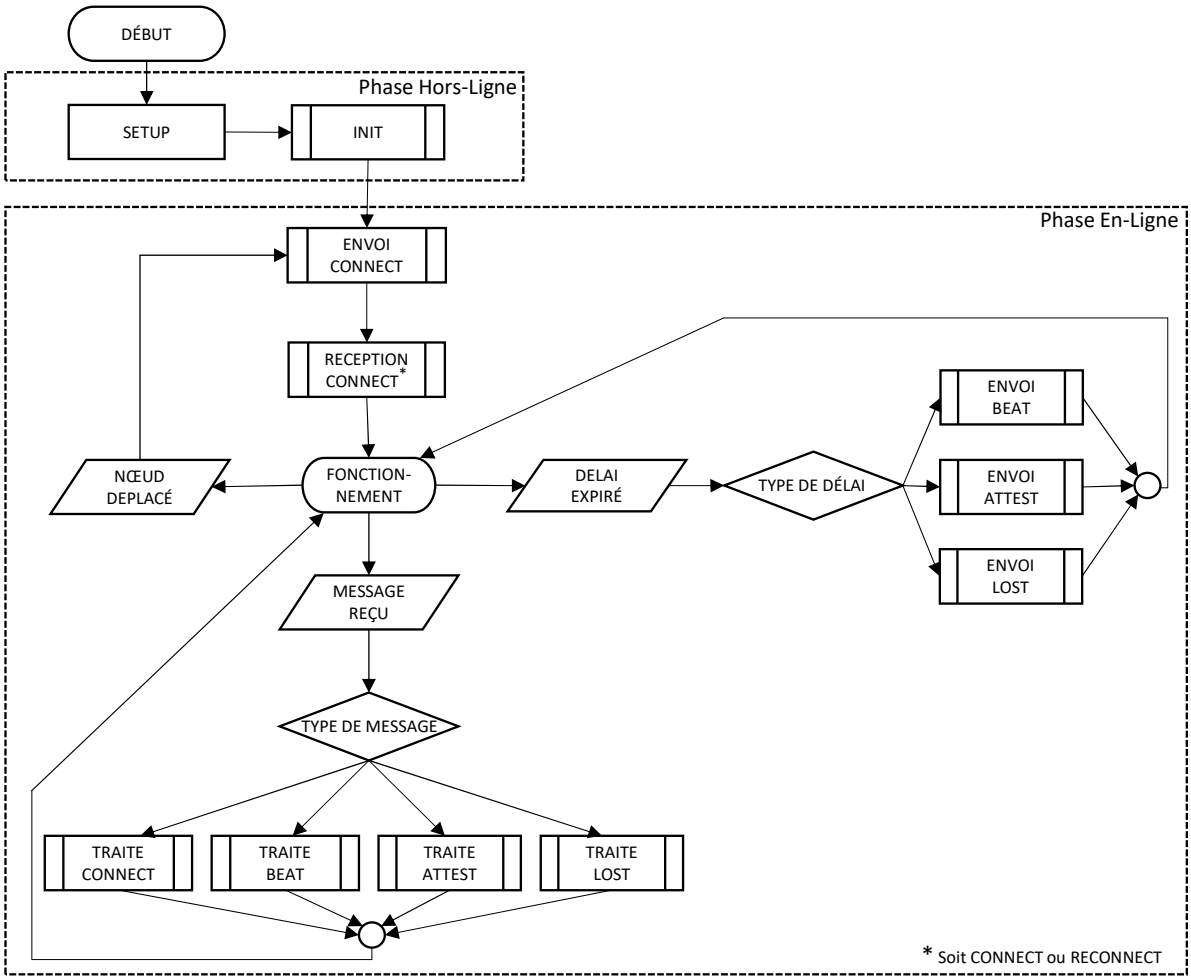


FIGURE 3.3 – Différents états d'un appareil dans CRAFT

l'opérateur O et doit être basée sur le contexte de déploiement du réseau et le niveau de sécurité intrinsèque des nœuds.
Ainsi que présenté figure 3.4, les nœuds D_i peuvent appartenir soient à la classe K soit à la classe L, en fonction du niveau de sécurité du nœud. Certains appareils peuvent avoir un niveau de sécurité insuffisant et donc ne pas faire partie de l'ensemble des nœuds L et K : ces appareils ne sont pas inclus dans le réseau. Le paramètre de niveau de sécurité s_i d'un nœud sert à définir deux autres caractéristiques de ce nœud : son rôle dans le réseau, et l'ensemble des paramètres de l'appareil.

Le rôle d'un nœud dans le réseau est soit un *node* soit un *endpoint*. Les *nodes* sont des intersections réseau et communiquent avec d'autres *nodes* ainsi qu'avec des *endpoints*. Les *endpoints* sont situés en bordure de réseau, et ne communiquent qu'avec des *nodes*.

L'ensemble des nœuds K ont la possibilité d'être aussi bien un *node* qu'un *endpoint*, car ils ont un niveau de sécurité suffisant pour être aux intersections du réseau. Ils peuvent donc communiquer entre nœuds K et avec les nœuds L. De leur côté les nœuds L peuvent uniquement être des *endpoints* et communiquer avec les nœuds K voisins. Les nœuds L font ainsi partie du *Réseau externe* tel qu'illustré figure 3.2, par opposition aux nœuds K qui constituent le *Cœur de réseau*.
Un *endpoint* communique uniquement avec un *node* pour transmettre des données applicatives ou des données d'attestation. Les *nodes* quant à eux communiquent aussi bien avec les autres *nodes* qu'avec les *endpoints*. Ils participent à l'attestation du réseau et au routage de l'ensemble des données.
Un exemple de déploiement de réseau est détaillé plus loin dans cette section.

Un appareil D_i peut être sécurisé à l'aide de différents mécanismes matériels (TEE, TPM, ...) ou logiciels (partitionnement mémoire, ...) et chacun peut avoir une solution différente, avec divers degrés de sécurité. L'ensemble des solutions qui existent dans le réseau doivent être listées et triées par niveau de sécurité, et chaque nœud doit satisfaire un niveau de sécurité minimal st_L déterminé par le contexte de déploiement. Deux appareils strictement identiques prenant part au réseau peuvent avoir différentes valeurs de s_i en fonction du contexte de déploiement. Par exemple, un appareil installé à l'intérieur d'un bâtiment avec des contrôles d'accès sera moins sujet aux attaques physiques qu'un appareil accessible par des personnes extérieures.

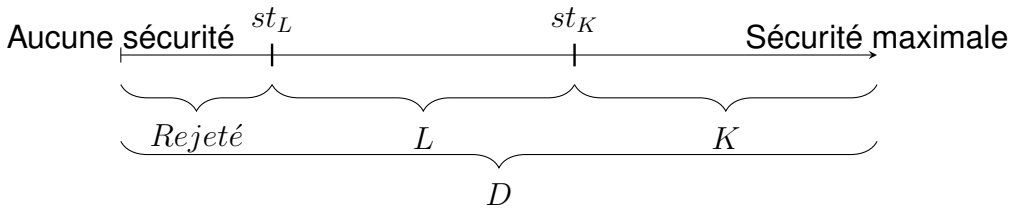


FIGURE 3.4 – Classification des nœuds par niveau de sécurité (échelle arbitraire)

Configuration des appareils Une fois que les nœuds ont été catégorisés, l'opérateur O peut leur attribuer des paramètres en fonction de s_i. Un exemple de configuration est donné plus loin dans cette section. Deux nœuds avec la même valeur de s_i devraient posséder les mêmes paramètres. En effet il n'est pas logique que pour un même niveau de sécurité au sein du déploiement un des deux appareils ait des paramètres plus stricts que l'autre. À l'inverse, des appareils avec des valeurs de s_i différentes peuvent malgré tout partager les mêmes paramètres pour faciliter le déploiement et si cela reste pertinent d'un point de vue de la sécurité globale du réseau.

Init Pendant l'étape *init*, O attribue au nœud ses paramètres de sécurité ainsi que ses paramètres fonctionnels. O fournit également à D_i différents éléments : P_i, PK_O et $Sign_O(P_i, PK_i)$, en suivant la procédure décrite dans la figure 3.5, afin de permettre à l'appareil de rejoindre le réseau. P_i contient les différents paramètres de D_i en ce qui concerne le framework. PK_O est la clé publique de O, qui permet à D_i d'authentifier les autres appareils lorsqu'ils sont insérés la première fois dans le réseau. $Sign_O(P_i, PK_i)$ représente la signature par O des paramètres P_i de D_i, ainsi que de sa clé publique PK_i, afin d'être accepté par les autres appareils lors de son arrivée dans le réseau. Cette étape peut être réalisée à l'aide d'un type de message distinct, ou bien elle peut être effectuée lors d'un paramétrage en usine. En fonction des exigences des protocoles d'attestation utilisés et des options choisies, d'autres paramètres peuvent également être fournis à cette étape.

Opérateur O PK_O, SK_O, PK_i		**Nœud** $D_{i \in K \cup L}$ PK_i, SK_i
Sélection des paramètres P_i **Génération** $\sigma \leftarrow Sign_O(P_i, PK_i)$	$\xrightarrow{\{P_i, \sigma, PK_O\}}$	**Stockage des paramètres**

FIGURE 3.5 – Étape *init*

Exemple de déploiement en monde réel Le réseau utilisé pour illustrer cet exemple est le cas d'usage de la smart city présenté page 47 comportant de nombreux appareils. Cette smart city est représentée sur la figure 3.6. Ces appareils seront séparés en différents groupes, en fonction de leur niveau de sécurité. Au moment du déploiement du réseau, l'opérateur O assigne un niveau de sécurité à chacun des appareils, selon une échelle choisie par O sur la base de ses connaissances de la smart city. Chaque appareil individuel de ces groupes peut avoir des niveaux de sécurité différents, mais dans cet exemple arbitraire seules de grandes catégories sont utilisées :

— Les antennes de dernière génération sont définies en tant que $(D_5, s_5 = 28)$

— Les antennes plus anciennes sont définies en tant que $(D_4, s_4 = 25)$

— Les véhicules appartenant à la ville sont définis en tant que $(D_3, s_3 = 21)$

— Les capteurs de pollution de l'air sont intégrés à des équipements mobiles, tels des vélos ou des drones, et sont définis en tant que $(D_2, s_2 = 14)$

— Les parkings, lampadaires, ou encore feux de signalisation connectés et intelligents sont définis en tant que $(D_1, s_1 = 12)$

— Les caméras de sécurité (bas de gamme) sont définies en tant que $(X, s_X = 5)$

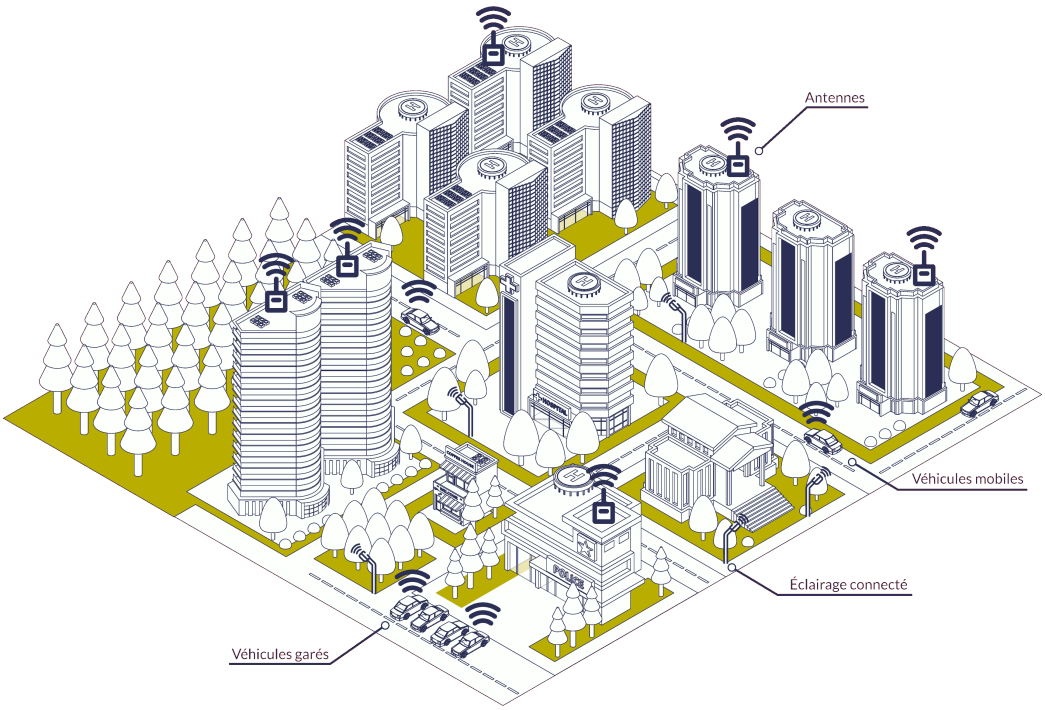


FIGURE 3.6 – Représentation d'une smart city

L'opérateur O choisit également une limite de sécurité pour les classes de nœuds K et L, avec $st_L = 10$ et $st_K = 20$. Le résultat, tel qu'illustré en figure 3.7, donne les ensembles $K = \{D_3, D_4, D_5\}$ et $L = \{D_1, D_2\}$. X, représenté ici par des caméras bas de gamme, est exclu du réseau en raison du niveau de sécurité trop faible.

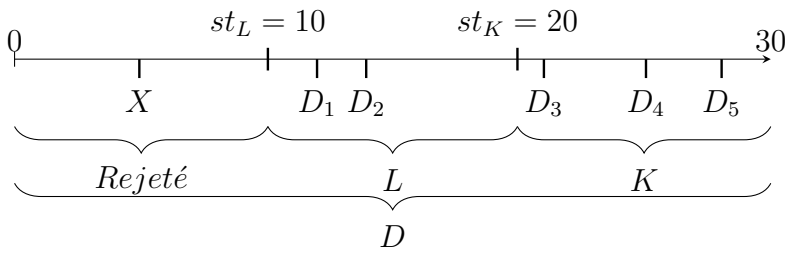


FIGURE 3.7 – Classification des nœuds par niveau de sécurité dans l'exemple de la smart city (échelle arbitraire de 0 à 30)

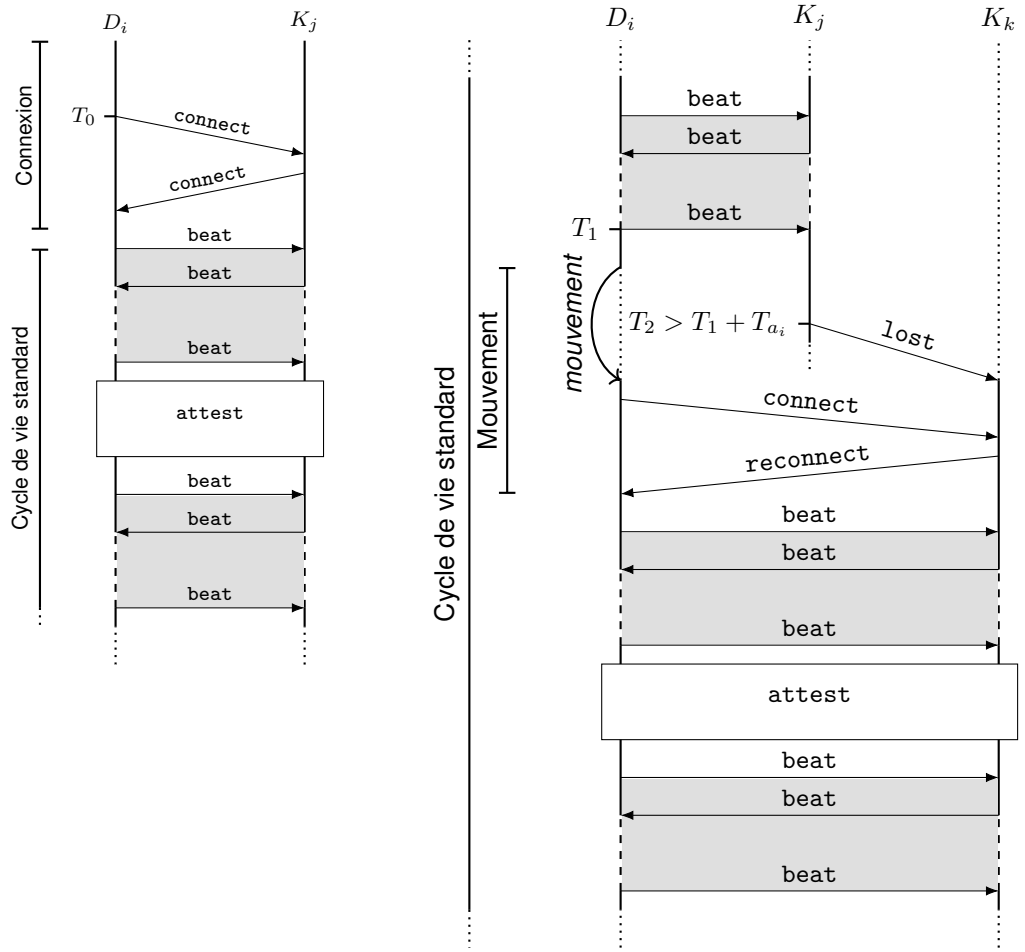
Une fois que les nœuds sont classés, l'opérateur O peut leur attribuer des paramètres. Les nœuds L se voient fixer une valeur de $T_{a_i} = 3600s$ et les nœuds K une valeur de $T_{a_i} = 7200s$, car ils sont considérés comme étant mieux sécurisés. Ces valeurs sont

basées sur le temps estimé nécessaire à un adversaire pour attaquer physiquement ces appareils. La valeur de T_{b_i} est choisie à $60s$ pour les nœuds K comme pour les nœuds L : ils doivent donc signaler leur présence au réseau avec la même régularité. Enfin, les nœuds L se voient attribuer pour h_i les valeurs $h_1 = 0$ et $h_2 = 1$, ce qui permet aux appareils mobiles des mouvements minimaux. Les nœuds K sont eux différenciés avec les valeurs $h_3 = 3$ et $h_4 = h_5 = 0$, puisque les véhicules de la ville D_3 doivent bénéficier d'une plus grande mobilité que les appareils D_2, et que les antennes D_4 et D_5 sont fixes. Ainsi, ces nœuds peuvent être instanciés à l'aide de la définition $P_i = \{id_i, s_i, h_i, T_{a_i}, T_{b_i}\}$, ce qui donne :
$P_1 = \{1, 12, 0, 3600, 60\}$
$P_2 = \{2, 14, 1, 3600, 60\}$
$P_3 = \{3, 21, 3, 7200, 60\}$
$P_4 = \{4, 25, 0, 7200, 60\}$
$P_5 = \{5, 28, 0, 7200, 60\}$

Les appareils peuvent désormais être initialisés par l'opérateur O pour rejoindre le réseau et ainsi prendre part aux échanges de messages de données et d'attestation. La *Phase En-Ligne* peut alors commencer.

En-ligne

Pendant la *Phase En-Ligne* telle qu'illustrée figure 3.8, plusieurs types de messages sont échangés.
La figure 3.8(a) montre l'insertion d'un appareil dans le réseau en utilisant le message `connect`, suivi des échanges de messages `beat` et `attest` que cet appareil réalise durant son cycle de vie standard afin de permettre l'attestation continue.
La figure 3.8(b) montre l'échange de message réalisé lorsque l'appareil D_i se déplace et change de voisins durant le cycle de vie standard du réseau. Le voisin initial de D_i, K_j, ne recevant plus de message `beat` depuis un temps supérieur à T_{a_i}. Il va alors envoyer un message `lost` à ses propres voisins, y compris K_k. Ainsi, D_i va pouvoir se reconnecter à K_k avant de reprendre l'échange standard de messages.
Les messages `connect`, `beat`, `attest` et `lost` constituent ainsi un ensemble minimal de messages requis pour répondre aux exigences fonctionnelles et de sécurité de notre framework. En fonction du protocole d'attestation utilisé et de ses spécificités (consensus, agrégation, ...) d'autres messages peuvent être nécessaires.
Seul le format général des différents paquets est défini dans ce travail, leurs longueurs sont laissées au choix de la personne chargée de l'implémentation car la majorité de ces paquets peut varier en fonction des opérations cryptographiques utilisées ou de l'inclusion de champs optionnels, ainsi que mentionné dans les exigences **PR3** à **PR5**.



(a) Connexion et début du cycle de vie du réseau

(b) Déplacement d'appareil et reconnexion

FIGURE 3.8 – Échange de messages entre des nœuds durant la *Phase En-Ligne*

Head	Size	Parameters	Device Id
Timestamp	...		

FIGURE 3.9 – Définition de l'en-tête commun aux différents paquets du framework

Comme illustré figure 3.9, chaque message se compose des champs suivants :

— Un champ appelé `Head` pour distinguer les différents messages.

— Un champ `Size`, qui représente la taille complète du message en groupes de 32 bits.

— Un champ `Parameters` qui contient des éléments de paramétrage afin que les nœuds puissent correctement communiquer et échanger des messages : par exemple, une partie du champ de bits se voyant attribuer la valeur $(01)_2$ pourrait signifier l'usage de l'algorithme de hachage SHA-1, tandis que la valeur $(10)_2$

pourrait signifier SHA-256. Ainsi, le champ `Parameters` permet de répondre à l'exigence **EF2**.

— Un champ `Device Id` qui identifie l'expéditeur du message

— Un champ `Timestamp`, utilisé pour éviter le rejeu de messages, et s'assurer que tout message trop ancien est ignoré.

— Il n'y a pas besoin d'inclure de checksum car le framework proposé se situe au niveau applicatif du modèle OSI. Il repose donc sur des couches protocolaires plus basses (*e.g.* TCP) pour assurer l'intégrité des paquets.

Les messages tels que `beat` sont authentifiés, en utilisant soit une signature soit un Message Authentication Code (MAC). Une signature nécessite plus de temps de calcul mais est plus sécurisée et liée à un unique nœud. Un MAC est plus rapide à calculer, plus flexible en termes de taille, et repose sur des clés symétriques partagées à l'aide d'une paire de clés asymétriques. Le choix entre ces deux solutions dépend des appareils qui prennent part au réseau et de leurs exigences de sécurité respectives. Cette flexibilité contribue notamment à remplir l'exigence **EF3** en ajustant le volume de données supplémentaires induit par les protocoles cryptographiques, et l'exigence **EF4** en ajustant les protocoles cryptographiques utilisés en fonction du contexte de déploiement.

Head	Size	Parameters	Device Id
Timestamp	h_i	T_{a_i}	T_{b_i}
Options Size	Other Options		
PK_i			Signature Expiration
Public Key and Device Parameters Signature by O			

FIGURE 3.10 – Définition du paquet `connect`

Ajouter des appareils au réseau L'étape de connexion telle qu'illustrée figure 3.8(a) utilise le message `connect`, représenté figure 3.10. Ce message est envoyé dans le réseau par tout nouveau nœud afin qu'il soit accepté par ses voisins, et lesdits voisins renvoient un message `connect` au nœud en cours de connexion. `connect` contient les paramètres du nœud émetteur h_i, T_{a_i} et T_{b_i}. Le champ `Other Options` permet d'ajouter à CRAFT de nouvelles fonctionnalités qui seraient nécessaires au réseau dans le futur et nécessiteraient des champs supplémentaires. Cela aurait pu être le cas pour les fonctionnalités ASMP présentées en section 3.4.3 si elles n'utilisaient pas le champ `Parameters` existant. Rendre possible cette flexibilité, sans l'ajout de paquet supplémentaire, répond à l'exigence **EF5**. Les champs `Options Size` et `Other Options`

peuvent être omis, en positionnant un bit dans le champ `Parameters`. Une clé publique PK_i est partagée pour permettre le chiffrement au travers de l'usage d'un secret partagé, créé à l'aide d'un algorithme tel que ECDH au moment où le message `connect` est reçu et validé. L'authenticité de la clé publique et celle des paramètres du nœud sont attestées par la signature de O (c'est-à-dire le champ `Public Key and Device Parameters Signature by` O). Cette signature expire en fonction de la date de validité placée dans le champ `Signature Expiration` afin de s'assurer que seul un nouveau nœud sera accepté par les autres nœuds, car un nœud qui aurait été configuré il y a trop longtemps pourrait avoir été attaqué avant de rejoindre le réseau.

Cycle de vie standard Tout au long de la vie du réseau, telle que représentée figure 3.8, les messages `beat` et `attest` sont échangés de façon régulière afin de s'assurer de l'attestation continue des appareils au sein du réseau. C'est notamment l'utilisation des messages `beat` qui permet d'assurer l'attestation continue et donc la meilleure sécurité en complétant les attestations à l'aide d'une vérification bien plus fréquente mais aussi plus légère en terme de temps de calcul et d'impact sur le réseau, car contenant peu d'informations en comparaison à une attestation.
`beat` présenté en détail figure 3.11 est un message envoyé par D_i à intervalles réguliers à ses voisins directs qui font partie du *Cœur de réseau* (soit les nœuds K) ainsi qu'illustré figure 3.12. L'échange de ce message prouve que D_i n'a pas bougé ou n'a pas été déconnecté du réseau. Afin de réduire le nombre de messages, et ainsi le volume de données et le coût en énergie ajouté par le framework, le champ `Optional Data` peut être envoyé dans le message `beat`, en accord avec les exigences **EF3** et **EF5**. Par exemple, au lieu d'utiliser un message distinct, l'attestation peut être déclenchée à l'aide du même message `beat`, réduisant ainsi l'impact du framework. L'entièreté du message doit être authentifée à l'aide de la clé k_{ij} afin d'assurer que les communications ont uniquement lieu entre appareils de confiance.

Head	Size	Parameters	Device Id
Timestamp	Optional Data		
Authentication			

FIGURE 3.11 – Définition du packet `beat`

La figure 3.12 illustre qu'à l'expiration de T_{b_i}, un nœud D_i envoie un message `beat` contenant l'identifiant id_i du nœud ainsi que l'heure d'envoi du message. Chaque nœud K_j qui reçoit ensuite ce message doit vérifier l'authentification de celui-ci en utilisant la clé partagée k_{ij} et doit vérifier que H_i^{t-1} existe dans H_j^{list}. Si ces deux conditions

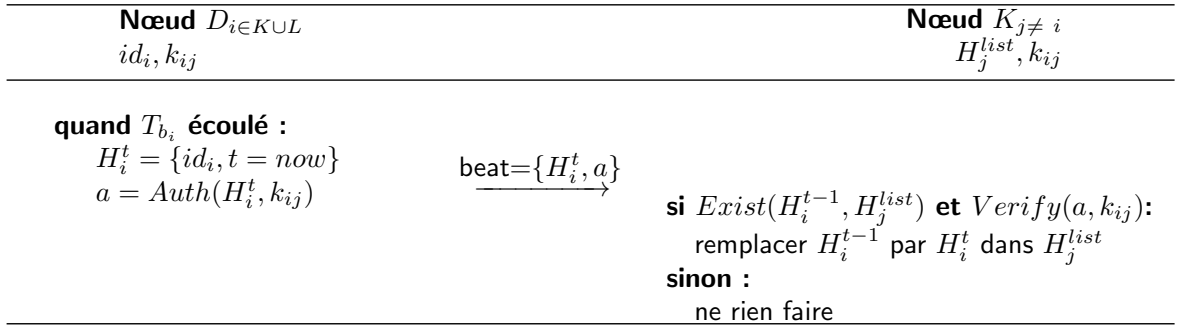


FIGURE 3.12 – Échange du message `beat` entre n'importe quel nœud et un nœud K

sont remplies, alors les nœuds peuvent mettre à jour leurs informations concernant D_i. Dans le cas contraire, ces nœuds ignorent simplement le message.

`attest` est un message (ou un groupe de messages) envoyé en fonction du protocole d'attestation utilisé. Étant donné que le fonctionnement de ces protocoles est très variable, aucune description de ces paquets n'est donnée, en dehors du fait que le format général des paquets du framework doit être respecté (voir figure 3.9). Les attestations peuvent également être broadcastées en utilisant le message `beat`, afin de minimiser le nombre de messages et donc le poids du framework dans les communications des nœuds. En effet, il est moins coûteux en performances réseau d'envoyer plusieurs informations dans un même message que d'envoyer plusieurs messages dans lesquels les en-têtes et méta-données sont répétés. Pour ce faire, le champ optionnel `Optional Data` inclus dans les messages `beat` peut être utilisé aussi bien pour déclencher ou diffuser une attestation tout en s'affranchissant d'envoyer plus de messages dont les en-têtes sont redondants. Cette flexibilité dans l'usage de CRAFT répond ainsi à l'exigence **EF1**.

Pour prendre en compte la mobilité des appareils, les paquets `lost` et `reconnect` sont également définis. La façon dont ils s'intègrent dans le cycle de vie du réseau est décrite figure 3.8(b).

Head	Size	Parameters	Device Id
Timestamp	Lost Device Id	Initial Sender Id	TTL
Expiry Time	Lost Device Configuration Hash		
Encrypted Authentication of Initial Sender			
Authentication			

FIGURE 3.13 – Définition du paquet `lost`

Le message `lost`, défini figure 3.13, est envoyé par K_j à ses nœuds voisins pour annoncer que le nœud D_i ne communique plus, et donc s'est peut-être déplacé. En complément des champs communs déjà présentés dans la figure 3.9, ce message contient l'identifiant id_i du nœud perdu dans le champ `Lost Device Id` afin d'identifier le nœud manquant. Le paquet contient également id_j dans le champ `Inital Sender Id` pour identifier le nœud à l'origine du message. Le champ `TTL` indique combien de communications réseau le message peut encore parcourir, et ce champ est décrémenté à chaque retransmission du message afin d'en limiter la portée. Par exemple, une valeur de *1* pour le champ `TTL` signifie que le message sera retransmis une dernière fois aux voisins, tandis qu'une valeur de *0* fait que le message ne sera pas retransmis vers les voisins après sa réception. Les quatre champs restants du paquet ont tous pour but de permettre au nœud perdu de se reconnecter de manière sécurisée à un autre nœud distant qui a bien reçu le message `lost` par l'intermédiaire de ses nœuds voisins :

— Le champ `Expiry Time` empêche le message `lost` d'être intercepté par un attaquant et d'être conservé pendant une longue période. Les messages `lost` reçus après l'expiration de T_{a_i} sont rejetés.

— Le champ `Lost Device Configuration Hash` contient un haché des paramètres h_i, T_{a_i}, T_{b_i} et PK_i du nœud D_i. Ainsi, quand D_i se reconnecte au réseau, son nouveau voisin K_k peut vérifier que les paramètres envoyés à la reconnexion correspondent bien au haché reçu dans le message `lost`, prouvant ainsi l'authenticité de D_i. N'importe quel nœud peut envoyer les bons paramètres durant l'échange de `connect`, mais seul D_i peut authentifier les communications suivantes. En effet, même si d'autres appareils connaissent PK_i, seul D_i connaît SK_i, et donc seul D_i peut générer une clé de session k_{ik} correcte avec son nouveau voisin K_k.

— Le champ `Encrypted Authentication by Initial Sender` permet aux nœuds perdus de savoir qu'ils se sont reconnectés à un autre nœud de confiance qui avait bien reçu le message `lost` au travers d'autres nœuds appartenant à une chaîne de confiance : le premier nœud K_j à envoyer le message authentifie les champs `Lost Device Id`, `Initial Sender Id` et `Lost Device Configuration Hash` (par exemple en utilisant l'algorithme HMAC avec la clé k_{ij}). Ces champs ne sont pas modifiés durant le transfert des messages `lost` d'un nœud à l'autre. D_i peut alors vérifier que le message `lost` a bien été émis par K_j en utilisant à son tour HMAC et k_{ij}.

  Avant d'envoyer le message `lost` à un voisin K_k, K_j chiffre cette authentification en utilisant k_{jk} dans le champ `Encrypted Authentication of Initial Sender`. Ce champ est déchiffré à chaque saut réseau par les voisins de confiance qui reçoivent le message, et chiffré à nouveau avant de transmettre le message `lost`

à leurs voisins respectifs avec le champ `Encrypted Authentication of Initial Sender` mis à jour. Ainsi, le message `lost` peut seulement suivre un chemin de confiance, et D_i peut avoir confiance dans le nœud auquel il se reconnecte puisque c'est un nœud de cette chaîne de confiance.

— Le dernier champ `Authentication` est le même que dans le message `beat`, et permet simplement à un nœud de vérifier l'authenticité du message qu'il vient de recevoir.

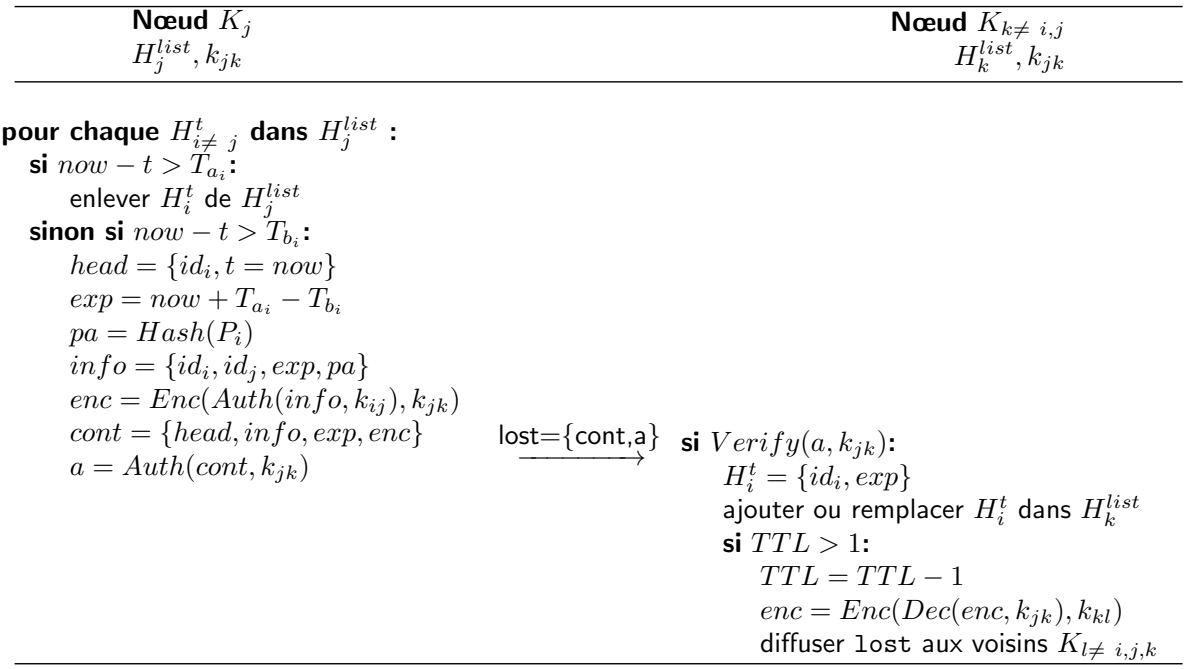


FIGURE 3.14 – Diffusion du message `lost` entre deux nœuds

La figure 3.14 illustre la façon dont le message `lost` est reçu et envoyé :

— K_j vérifie de façon régulière si un de ses voisins $D_{i\neq j}$ n'a pas envoyé de message `beat` depuis plus de T_{a_i}

— Si $D_{i\neq j}$ n'en a pas envoyé, K_j supprime $D_{i\neq j}$ de sa mémoire et considère alors que $D_{i\neq j}$ n'est plus de confiance

— Si $D_{i\neq j}$ a envoyé un `beat`, K_j vérifie si $D_{i\neq j}$ a été silencieux plus que T_{b_i}

— Ensuite, si $D_{i\neq j}$ a été silencieux plus de T_{b_i}, alors K_j diffuse un message `lost` à tous ses voisins $K_{k\neq i,j}$

— En recevant ce message, les voisins K_k ajoutent $D_{i\neq j}$ à leur liste H_k^{list}, dans l'attente d'une tentative de reconnexion

— Si la valeur du champ `TTL` est supérieure à 0, K_k décrémente cette valeur et fait suivre le message `lost` à ses propres voisins

— Si D_i ne se reconnecte pas avant que T_{a_i} ne soit écoulé, il est supprimé de la liste H_k^{list}

Lorsque D_i se reconnecte à un nœud K_k après s'être déplacé, il envoie un message `connect` à k_k mais sans les champs `Signature Expiration` et `Parameters Signature by` O. Ces deux champs ne sont en effet plus nécessaires car la confiance repose désormais sur le fait que K_k a bien reçu un message `lost`.
En réponse à cela, K_k va envoyer un message `reconnect` décrit figure 3.15. Ce paquet contient des informations à la croisée des messages `connect` et `lost`. Cela permet à D_i d'établir une connexion avec K_k et de prouver que K_k a reçu un message `lost` qui provient d'une chaîne de nœuds de confiance.

<table>
<tr><td>Head</td><td>Size</td><td>Parameters</td><td>Device Id</td></tr>
<tr><td>Timestamp</td><td>h_i</td><td>T_{a_i}</td><td>T_{b_i}</td></tr>
<tr><td>Options Size</td><td colspan="3">Other Options</td></tr>
<tr><td colspan="3">PK_i</td><td>Lost Device Id</td></tr>
<tr><td>Initial Sender Id</td><td colspan="3">Lost Device Configuration Hash</td></tr>
<tr><td colspan="4">Encrypted Authentication of Initial Sender</td></tr>
<tr><td colspan="4">Authentication</td></tr>
</table>

FIGURE 3.15 – Définition du paquet `reconnect`

L'ensemble des messages et échanges ainsi décrits permet le bon fonctionnement du framework CRAFT, et donc l'amélioration des performances et de la sécurité du réseau au travers de l'attestation continue. Sur cette base du framework, des fonctionnalités additionnelles peuvent être implémentées comme les fonctionnalités ASMP (Adaptive Simultaneous Multi-Protocols) présentées dans la section suivante qui permettent d'utiliser plusieurs protocoles d'attestation et d'en changer en fonction du contexte.

3.4.3 Description des fonctionnalités ASMP de CRAFT

Les fonctionnalités ASMP (Adaptive Simultaneous Multi-Protocols) sont un exemple de fonctionnalité qui peut venir s'ajouter en plus du framework CRAFT de base. Elles illustrent à la fois la flexibilité de CRAFT, et permettent aux appareils d'utiliser plusieurs

protocoles d'attestation à distance et en particulier d'en changer en fonction du contexte.

Afin de permettre la sélection du protocole d'attestation le plus adapté en fonction de l'état courant de l'appareil et du contexte dans lequel cet appareil se trouve, le champ `Parameters` de chaque message est utilisé. Ce champ est défini figure 3.16. L'implémentation actuelle utilise un champ de 64 bits, numérotés de 0 à 63.
Les bits les plus à gauche étiquetés `(1)` ont une valeur de 1 lorsque le protocole d'attestation correspondant est disponible sur l'appareil (par exemple, si les bits 0 et 1 sont à 1, les protocoles d'attestation P1 et P2 peuvent être utilisés, si on considère que le framework supporte ces deux protocoles).
Le même nombre de bits que dans `(1)` est utilisé pour définir le protocole d'attestation préféré de l'appareil dans `(3)`. Quand un bit a la valeur 1 dans `(3)`, le protocole d'attestation correspondant dans `(1)` est désigné comme le préféré (par exemple, quand le bit 32 a la valeur 1, l'appareil va de préférence utiliser le protocole P1). Le protocole préféré peut changer à tout moment, ce qui permet aux appareils d'utiliser le protocole d'attestation le plus approprié à chaque instant en fonction du contexte environnant.
Le bit 31 étiqueté `(2)` prend la valeur 1 lorsque l'appareil est un nœud K.
Enfin, les bits étiquetés `(r)` sont réservés pour un usage futur (par exemple pour de nouveaux protocoles d'attestation).

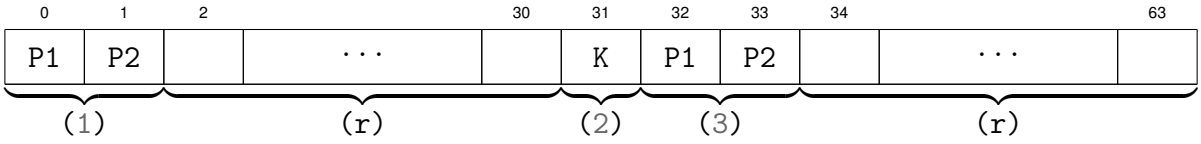


FIGURE 3.16 – Détails du champ `Parameters`

La figure 3.17 représente l'envoi des messages entre le nœud L_i et le nœud K_j. Le nœud L_i envoie des messages `beat` à intervalles réguliers, en indiquant comme protocole préféré le protocole P1. K_j déclenche donc une phase d'attestation en utilisant ce protocole. Suite à un changement de son environnement et de son contexte (par exemple, L_i peut être un nœud mobile devenu immobile), L_i envoie désormais P2 comme étant son protocole favori. A la prochaine attestation, K_j va donc utiliser le protocole P2.

3.4.4 Analyse de sécurité de CRAFT

Dans cette section, le modèle d'attaquant et les bases de la sécurité de CRAFT sont présentées. Ensuite, des méthodes par lesquelles un attaquant Adv peut tenter de contourner les exigences de sécurité **ES1** et **ES2** sont discutées concernant différents aspects de CRAFT.

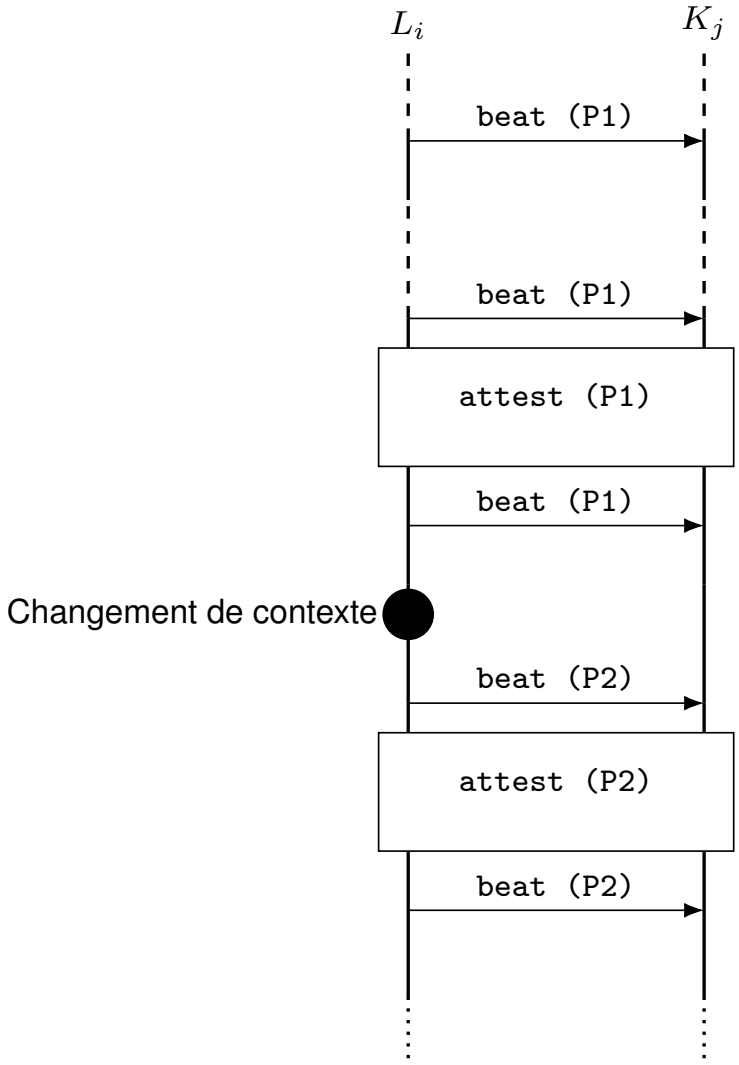


FIGURE 3.17 – Illustration des échanges de messages entre deux appareils avant et après un changement de contexte

Modèle de menace

Comme dans les travaux similaires [36][23][25], l'opérateur O qui supervise le réseau est considéré comme étant de confiance. Tous les autres appareils sont sujets aux attaques d'un adversaire Adv, avec différentes capacités d'attaques.

— Sur la base du modèle Dolev-Yao [68], Adv a le contrôle total sur les communications : il peut écouter, modifier, supprimer ou générer des messages entre les appareils.

— Adv peut également compromettre les logiciels de n'importe quel appareil et ainsi contrôler l'exécution de code et la mémoire de l'appareil.

— Adv peut réaliser des attaques physiques, et passer outre les protections matérielles en place, à condition d'avoir suffisamment de temps : il est considéré que réaliser une attaque physique nécessite de déconnecter un appareil du réseau pendant une durée significative. Adv peut alors accéder à du code et de la mémoire protégés matériellement.

Les attaques par canaux auxiliaires sont exclues du périmètre de ce travail, car elles peuvent être exécutées sur des appareils en fonctionnement, et demandent donc d'autres solutions de détection et de contre-mesures. Les attaques par déni de service (DoS) ne sont pas considérées non plus.

Bases de la sécurité de CRAFT

La sécurité de CRAFT repose sur trois éléments :

— les protocoles d'attestation que CRAFT utilise ;

— les nouveaux messages introduits par le framework CRAFT présentés précédemment ;

— la sécurité des paramètres utilisés tels que h_i, T_{a_i} et T_{b_i} définis précédemment.

La sécurité des protocoles d'attestation utilisés est exclue du périmètre de cette analyse, car chaque protocole est considéré comme étant un composant pris sur l'étagère dont le niveau de sécurité a été prouvé au préalable.
Plus ces protocoles sont sécurisés, plus le niveau de sécurité que CRAFT peut fournir est élevé. La sécurité des messages de CRAFT est détaillée dans la section 3.4.4, et repose sur la cryptographie utilisée. Enfin, la sécurité des paramètres de CRAFT repose sur la bonne adéquation de ces paramètres au regard du contexte de déploiement. Cela implique de l'opérateur O une bonne connaissance du scénario de déploiement ainsi que des appareils qui vont prendre part au réseau. L'avantage de ces paramètres est la grande flexibilité qu'ils apportent, mais leur limite repose sur le facteur humain utilisé pour l'analyse du contexte de déploiement.

Attaques sur les messages de CRAFT

Usurpation d'identité Adv peut essayer de ressembler à un appareil authentique dans le but d'être inclus dans le réseau. Cependant, les paramètres et notamment la clé publique PK_i sont signés par l'opérateur O, de tel sorte que cet appareil ne pourra pas rejoindre le réseau. Ainsi, l'exigence **ES1** est respectée en ce qui concerne l'usurpation d'identité.

Rejeu de messages En conservant un message authentique et en le renvoyant à un autre moment, Adv peut tenter d'interférer avec les appareils prenant part au réseau. Cependant, un horodatage est toujours inclus dans les messages du framework et authentifié de voisin en voisin à l'aide d'un haché et d'une clé partagée uniquement entre deux voisins. Comme les appareils sont synchronisés, tout message trop ancien est rejeté. Ainsi, l'exigence **ES1** est respectée en ce qui concerne le rejeu de messages.

Falsification de messages Adv peut tenter de construire un message de toutes pièces. Pour que ce message soit accepté, le champ d'authentification doit être correct, et donc Adv doit outrepasser la primitive d'authentification (par exemple HMAC), où obtenir une clé de la mémoire d'un appareil. Récupérer la clé nécessite alors de déconnecter l'appareil du réseau pour une durée supérieure aux limites fixées par l'opérateur O, et donc l'appareil serait exclu et la clé invalidée. Ainsi, l'exigence **ES1** est respectée en ce qui concerne la falsification de messages.

Clonage d'appareil Adv pourrait cloner un appareil dans son intégralité puis le replacer dans le réseau. Pour réaliser cela, l'appareil devrait être déconnecté du réseau pour une période de temps supérieure aux limites fixées par l'opérateur O, et serait donc considéré comme compromis et en conséquence exclu du réseau. Ainsi, un appareil cloné possédant les bonnes clés cryptographiques ne pourrait pas maintenir une communication avec le réseau, ce qui permet de respecter l'exigence **ES2**.

Prise de contrôle d'un appareil En prenant le contrôle d'un appareil, Adv peut tenter de prendre part au réseau. Cependant, cela nécessite d'outrepasser le protocole d'attestation utilisé et/ou d'attaquer physiquement l'appareil. Une attaque physique déconnecterait l'appareil pour une durée supérieure aux limites fixées par l'opérateur O et sera alors détectée. Ainsi, respecter l'exigence **ES2** permet d'éviter cela.

Attaque par trou de ver Adv peut établir un lien de communication direct entre deux appareils distants en relayant leurs messages. Cela ne fonctionnera pas, car les messages `lost` doivent être transmis en suivant un chemin de confiance pour que la connexion soit établie entre deux appareils. Ces deux appareils n'interagiront donc pas entre eux, et continueront à réaliser le mécanisme d'attestation indépendamment avec leurs voisins respectifs. Puisque l'attestation continue n'est pas impactée, l'exigence **ES2** est respectée.

3.5 Conclusion du chapitre

Dans ce chapitre, le framework d'attestation continue à distance CRAFT a été présenté. La définition de ce qu'est un framework d'attestation continue à distance a tout d'abord été donnée, permettant de comprendre qu'un framework vient compléter les protocoles d'attestation existants en répondant de manière plus générale au besoin de sécurité et de confiance des appareils et des réseaux. Au contraire des protocoles qui répondent à ce besoin pour des typologies de réseaux spécifiques ou pour des catégories d'appareils spécifiques, un framework est agnostique de ces problématiques et permet d'utiliser n'importe lequel de ces protocoles en même temps et en fonction du contexte.
Pour montrer l'aspect universel d'un framework, une définition la plus générale possible de ce qu'est un réseau a été présentée, dans laquelle les appareils sont séparés en deux catégories (les nœuds K faisant partie du *Cœur de réseau* et L appartenant au *Réseau externe*). Différents cas d'usages reposant sur cette définition ont également été détaillés afin de l'illustrer.
Une fois ces définitions établies, le framework CRAFT a été détaillé. CRAFT repose principalement sur l'échange de différents messages tout au long de la vie d'un appareil au sein du réseau. Ces échanges commencent au moment où l'appareil est configuré et continuent lors de son insertion dans le réseau et jusqu'à son fonctionnement standard ou son exclusion du réseau. CRAFT définit un format de message commun, qui permet d'inclure n'importe quel protocole d'attestation et donc de répondre à n'importe quel contexte de déploiement réseau (appareils mobiles ou non, à différents environnements, ...). Parmi ces messages, CRAFT repose notamment sur les messages `beat`, qui viennent grandement améliorer la continuité de l'attestation et donc la sécurité de l'appareil, tout en ayant un impact sur les échanges réseaux et la consommation énergétique bien plus faible qu'une phase d'attestation. Le message `lost` envoyé lors des déplacements permet également à CRAFT de se distinguer en créant une chaîne de confiance de voisin à voisin, permettant à un appareil de se reconnecter plus loin dans le réseau. Des fonctionnalités de CRAFT plus avancées, les fonctionnalités ASMP, ont également été introduites. Elles permettent, en utilisant les champs de CRAFT déjà définis, de faire varier le protocole d'attestation utilisé tout au long de la vie de l'appareil au sein du réseau. Ainsi, un appareil dont le type de mobilité change (fixe, mobile) lors de sa vie dans le réseau pourra utiliser le protocole d'attestation le plus adapté au contexte dans lequel il évolue.
Le chapitre suivant démontre au travers de simulations l'intérêt d'utiliser le framework CRAFT par rapport à des protocoles seuls en terme de performances, de sécurité et de flexibilité. Il démontre également l'intérêt des fonctionnalités ASMP dans la plupart des topologies complexes, car elles aussi permettent d'améliorer les performances et la sécurité de l'attestation à distance.

4

Évaluation du framework d'attestation continue par simulations

Sommaire

4.1 Introduction du chapitre

Dans ce chapitre, les performances et la sécurité de CRAFT sont évaluées en comparaison aux protocoles déjà existants. Dans la section 4.2, le simulateur d'événements discrets Omnet++ est présenté. Dans la section 4.3, CRAFT est comparé à des protocoles d'attestation seuls, illustrant les bénéfices d'utiliser un framework. Les protocoles utilisés dans cette comparaison sont SEDA [23] et US-AID [25] : SEDA a grandement influencé les autres travaux sur l'attestation à distance et est un point de comparaison souvent utilisé, tandis qu'US-AID est particulièrement adapté aux réseaux mobiles et propose un mécanisme de heartbeat similaire à celui que nous avons défini dans CRAFT. Dans la section 4.4, CRAFT est comparé avec et sans les fonctionnalités ASMP qui permettent l'utilisation de multiples protocoles d'attestation pour en démontrer l'utilité et les performances. Il est également comparé au framework que constitue la combinaison des protocoles SEDA et US-AID, illustrant à nouveau la meilleure efficacité d'un framework dédié tel que CRAFT qui répond à différents besoins de manière agnostique vis à vis du contexte.

4.2 Simulateur Omnet++

Afin de démontrer les performances des protocoles d'attestation, le simulateur d'événements discrets Omnet++ est souvent utilisé. C'est notamment le cas dans cette thèse pour comparer le framework d'attestation continue proposé aux protocoles existants. Omnet++ sert notamment à simuler des réseaux et les communications entre les différents composants de ce réseau. De nombreux frameworks de développement peuvent être ajoutés à Omnet++ afin de répondre à des besoins plus spécifiques. Dans

cette thèse, le framework Inet [69] est utilisé car il permet de facilement simuler des réseaux filaires ou sans fil et également des réseaux mobiles.

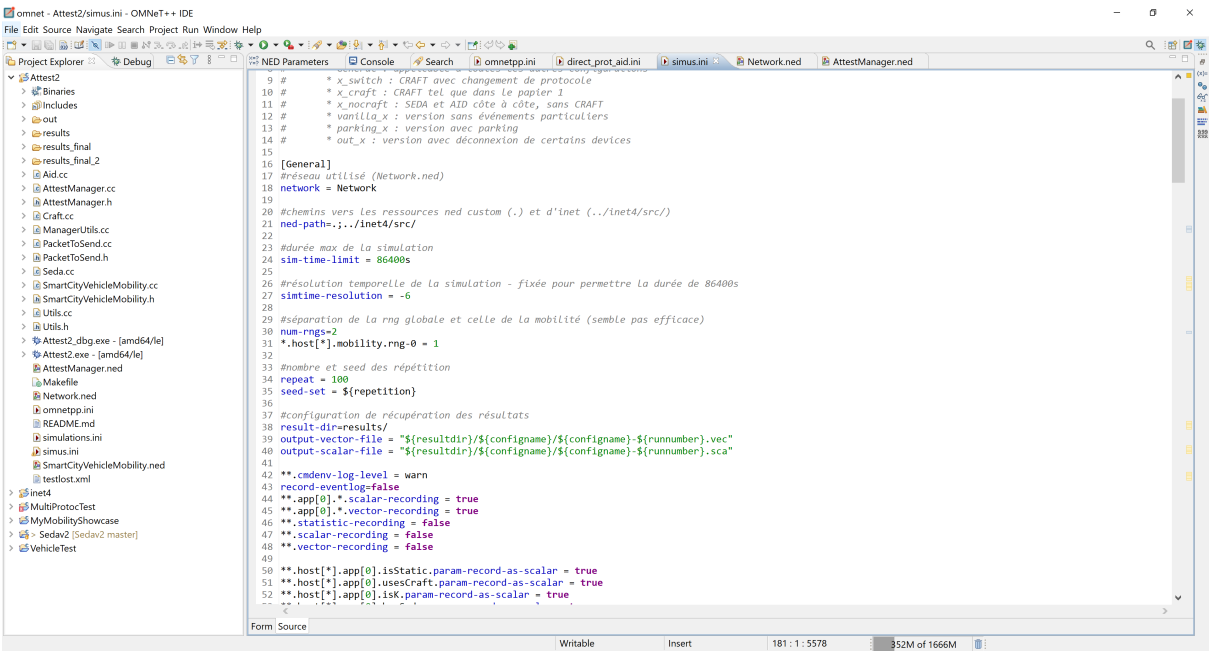


FIGURE 4.1 – Capture d'écran de l'interface principale d'Omnet++

L'architecture d'un projet Omnet++, dont l'interface est présentée figure 4.1, se compose de ces éléments principaux :

— Un fichier .ini qui contient les paramètres des simulations. Ce fichier définit quel réseau utiliser avec quels paramètres, comme le nombre d'appareils, la portée des réseaux sans-fil, la durée de la simulation, le nombre de répétitions,...

— Un ou plusieurs fichiers .ned qui définissent les réseaux et les appareils. Ce fichier définit les types d'appareils à utiliser, leurs connexions, leurs paramètres possibles et valeurs par défaut associées,...

— Des fichiers C++ qui contiennent le code qui va définir le comportement des différentes appareils de façon à répondre au besoin de la simulation.

Omnet++ permet différents modes d'utilisation, notamment un mode avec une interface comme présentée figure 4.2, qui permet d'observer le comportement des appareils et en particulier leur mobilité et le cheminement des messages. Ce mode est utile pour les phases de développement afin de s'assurer du comportement correct de la simulation. Omnet++ permet également un fonctionnement sans interface, qui permet d'obtenir les résultats finaux bien plus rapidement en réduisant les calculs nécessaires. Omnet++ dispose également d'interfaces permettant d'afficher des tableaux ou des graphes simples contenant les données des différents modules comme par exemple le

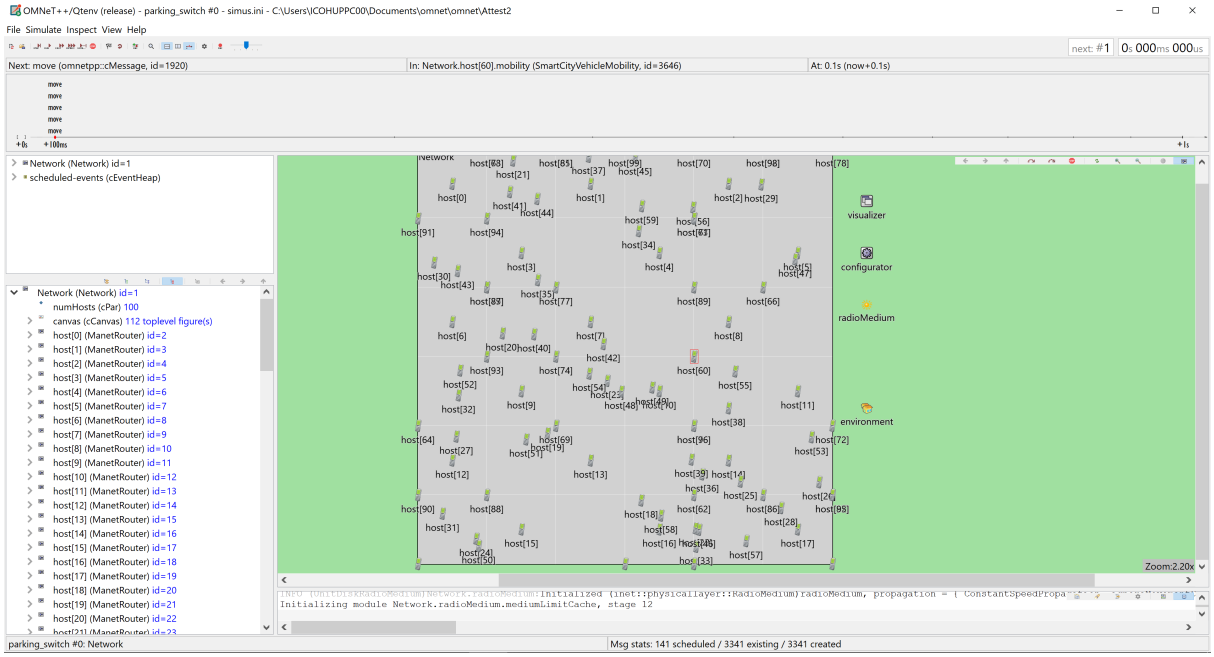


FIGURE 4.2 – Capture d'écran de l'interface graphique de simulations d'Omnet++

nombre d'octets échangés, ou toute autre paramètres déclaré dans les fichiers .ned et sauvegardés à l'aide du code du module.

4.3 Comparaison entre CRAFT et des protocoles d'attestation seuls

4.3.1 Métriques, scénarios et méthodologie

Métriques

Afin de comparer les performances de CRAFT à SEDA [23] et US-AID [25], deux métriques sont prises en compte.
La première métrique observée est le volume total de données échangées par les protocoles, car il peut avoir un impact sur le bon fonctionnement du réseau et sur la consommation d'énergie de celui-ci. La seconde métrique utilisée comme point de comparaison est le nombre d'opérations cryptographiques réalisées (qui sont ici des HMAC), car elles peuvent également impacter le temps d'exécution et la consommation d'énergie. L'évolution de ces deux métriques est étudiée dans les différent scénarios de mobilité.

Description des scénarios

Les différents protocoles ont été testés sur deux scénarios distincts (fixe et mobile) et avec un nombre de nœuds qui varie de 5 à 100 afin de comparer leurs performances dans différentes conditions. Les paramètres généraux de la simulation sont listés dans la table 4.1.

TABLE 4.1 – Paramètres de simulation d'Omnet++

Paramètre	Valeur
Densité d'appareils (appareils par km^2)	250
Portée de communication (m)	100
Placement des appareils	Aléatoire (avec seed)
Durée de simulation (s)	86400
Nombre de répétitions par scénario	100
Modèle radio	UnitDiskRadio
Modèle d'interface sans-fil	WirelessInterface
Modèle de mobilité	GaussMarkovMobility

Le premier scénario se compose d'appareils immobiles placés de manière aléatoire sur une surface carrée, qui représente un cas abstrait et général. La durée de la simulation est fixée à 86400 secondes (soit 24 heures), ce qui correspond à une période représentative d'utilisation de mécanismes d'attestation continue. La densité des appareils est quant à elle fixée à 250 appareils par kilomètre carré (c'est-à-dire que quand le nombre de nœuds augmente, la surface augmente en conséquence), ce qui permet d'observer uniquement l'influence du nombre d'appareils dans le réseau. La portée de communication est fixée à 100 m, de telle sorte que les nœuds sont en général à portée d'au moins un autre appareil, même si des déconnexions sont possibles.
Le second scénario possède les mêmes paramètres, à l'exception du fait que tous les appareils se déplacent dans la surface en suivant le modèle de mobilité Gauss-Markov inclus dans Omnet++. Ce modèle de mobilité permet à ces appareils de suivre une marche aléatoire des nœuds proche du mouvement brownien ; chaque nouveau mouvement étant réalisé dans une direction aléatoire. Une illustration du chemin suivi par un appareil en l'espace de quelques minutes de simulation se trouve figure 4.3.

Méthodologie

La méthodologie utilisée permet de fournir des mesures significatives ainsi que des intervalles de confiance autour des valeurs moyennes.
Chaque simulation d'un framework dans un scénario est répétée 100 fois en utilisant une graine (seed) différente à chaque répétition. Cette graine est utilisée par le générateur de nombres aléatoires d'Omnet++. Par exemple, le scénario fixe avec 100 appareils

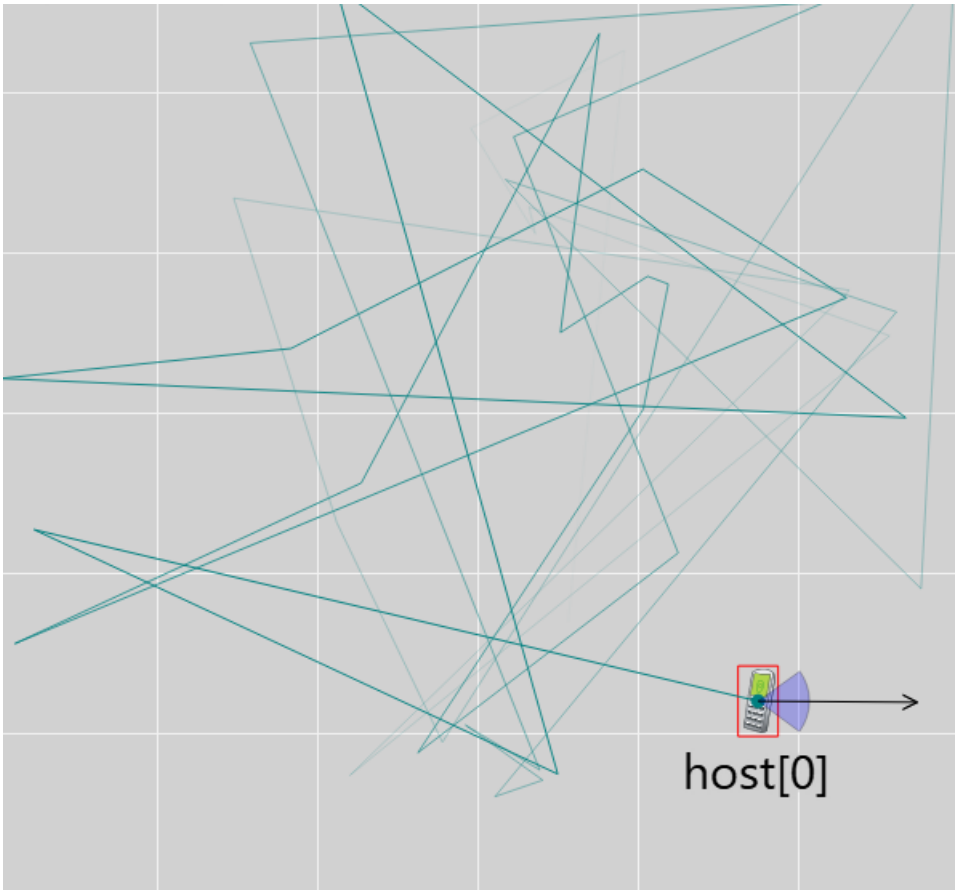
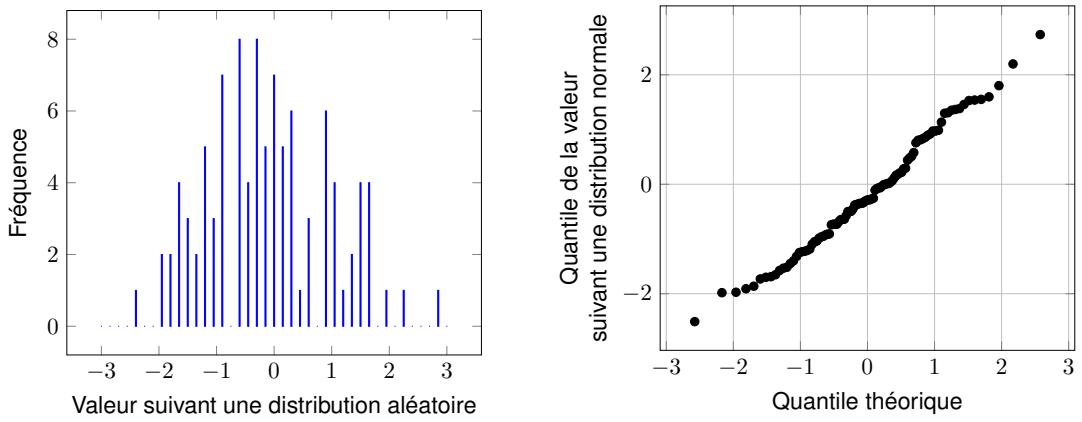


FIGURE 4.3 – Chemin suivi par un appareil utilisant le modèle de mobilité Gauss-Markov d'Omnet

qui utilise SEDA est répété 100 fois avec 100 graines différentes, et le scénario fixe avec 100 appareils qui utilise CRAFT+SEDA est répété 100 fois en utilisant les mêmes graines. Changer cette graine permet notamment de changer la position initiale des appareils ou leurs mouvements, et garantit donc que sur les 100 répétitions, ces éléments n'influencent pas le résultat.
En utilisant les résultats de simulation, l'hypothèse est faite que les mesures suivent une distribution normale.
Dans le but de valider cette hypothèse pour une métrique donnée, 100 répétitions d'un scénario donné sont évaluées à l'aide de deux méthodes graphiques : les histogrammes et les diagrammes Q-Q.
Les histogrammes montrent la fréquence des valeurs d'une métrique tandis que les diagramme Q-Q comparent les quantiles des valeurs d'une métrique aux quantiles théoriques d'une distribution normale. Si l'hypothèse est vérifiée, les histogrammes doivent grossièrement suivre la forme d'une distribution normale et les diagrammes Q-Q doivent suivre une ligne droite. Des exemples d'un histogramme et d'un diagramme

Q-Q de 100 valeurs aléatoires générées par Excel et qui suivent une distribution normale sont données en figure 4.4 à titre d'illustration des résultats attendus dans le cadre de cette méthodologie.

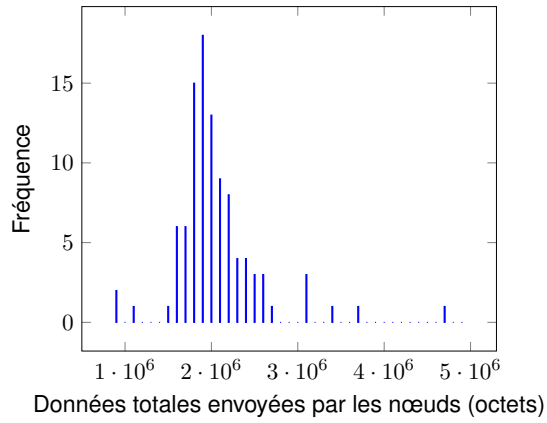


(a) Fréquence de 100 valeurs aléatoires qui suivent une distribution normale

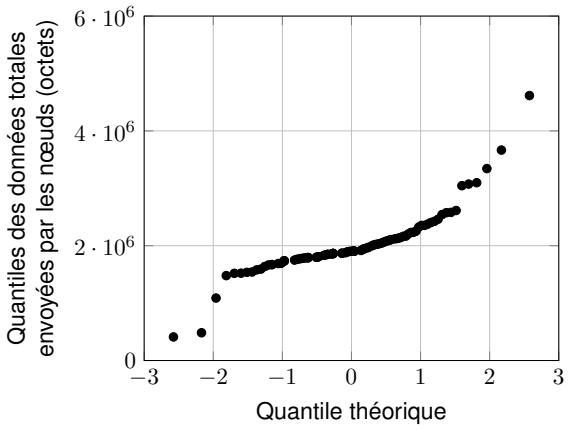
(b) Diagramme Q-Q de 100 valeurs aléatoires qui suivent une distribution normale

FIGURE 4.4 – Exemple de graphes de données aléatoires suivant une distribution normale

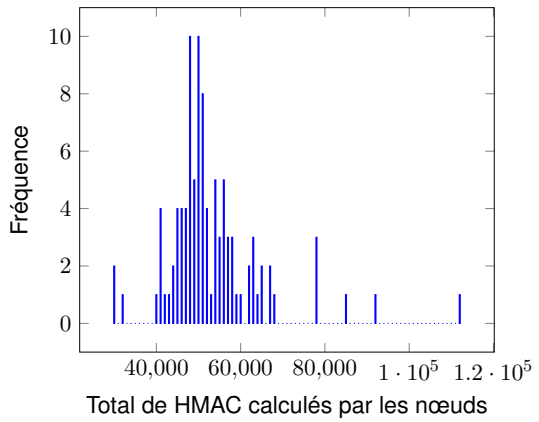
Cette méthodologie peut s'appliquer à toutes les métriques de l'ensemble des scénarios, mais pour des raisons de clarté et de concisions elle n'est présentée que pour deux métriques : le nombre total d'octets envoyés dans le réseau dans le scénario fixe avec 100 appareils qui utilise CRAFT+SEDA ainsi que le nombre total de HMAC calculés dans ce même scénario.
L'histogramme de la figure 4.5(a) est très proche d'une distribution normale, avec cependant une légère asymétrie sur le côté droit. De plus, les points du diagramme Q-Q figure 4.5(b) suivent majoritairement une ligne droite à l'exception des quelques points à l'extrémité du graphe, en particulier à droite.
De manière similaire, l'histogramme de la figure 4.5(c) montre seulement une légère asymétrie à droite, et suit principalement la forme d'une distribution normale. Le diagramme Q-Q de la figure 4.5(d) suit de son côté une ligne droite, sauf aux extrémités également. La conclusion à en tirer est que comme les données suivent approximativement une distribution normale, il est possible d'utiliser le Théorème Centrale Limite qui permet de considérer que les données suivent une loi normale à condition d'avoir un nombre de répétitions suffisant (en général supérieur à 30). Comme les simulations sont répétées 100 fois chacune, les résultats sont bien approximés par une distribution normale. Ainsi, des intervalles de confiance peuvent être calculés en utilisant la distribution t de Student qui permet ces calculs pour des échantillons de données de petite taille suivant approximativement une distribution normale. De cette manière, les résultats présentés dans la section suivante peuvent être facilement vérifiés et répétés.



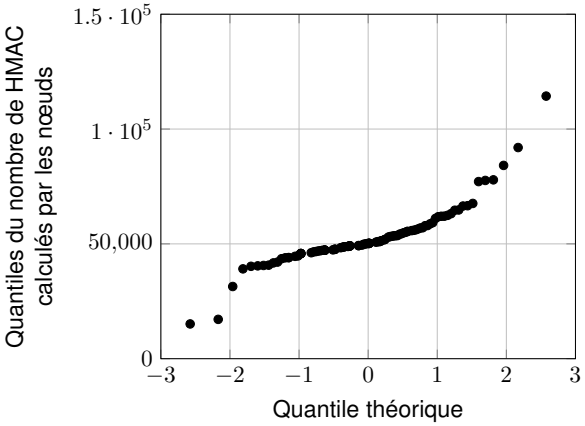
(a) Fréquence du nombre total d'octets envoyés dans le réseau sur 100 itérations



(b) Diagramme Q-Q du nombre total d'octets envoyés dans le réseau sur 100 itérations



(c) Fréquence du nombre total de HMAC calculés dans le réseau sur 100 itérations



(d) Diagramme Q-Q du nombre total de HMAC calculés dans le réseau sur 100 itérations

FIGURE 4.5 – Graphes montrant des données légèrement asymétriques

4.3.2 Détails concernant l'implémentation

Tous les nœuds implémentés dans ces simulation de CRAFT sont des nœuds K, ce qui représente le pire cas pour l'étude des performances de CRAFT étant donné que les nœuds K échangent plus de données que les nœuds L.
Les protocoles existants SEDA [23] et US-AID [25] ont été ré-implémentés dans le but de les comparer à CRAFT avec le plus de précision possible. Dans ces comparaisons, les nœuds sont fixes lorsque SEDA est utilisé, et mobiles lorsque US-AID est utilisé, afin de réaliser la comparaison dans des conditions nominales d'utilisation des protocoles tout en démontrant la flexibilité de CRAFT.
Étant donné que CRAFT est un framework et ne possède donc pas de protocole d'attestation en soit, il intègre les protocoles SEDA et US-AID. CRAFT utilise alors SEDA (formant CRAFT+SEDA) quand il est comparé à SEDA seul, et utilise US-AID

(formant CRAFT+US-AID) lorsqu'il est comparé à US-AID seul.
Ainsi, CRAFT+SEDA a le même mécanisme d'attestation que SEDA mais il permet l'utilisation des heartbeats, qui permettent soit d'accroître le niveau de sécurité du réseau en multipliant les communications soit de réduire la quantité de données ajoutées par le mécanisme d'attestation. Pour CRAFT+US-AID, le mécanisme d'attestation est le même que pour US-AID mais il apporte plus de flexibilité grâce aux heartbeats, qui remplacent alors les *Proof Of Non-Absence* (PONAs) d'US-AID. Bien que les heartbeats soient comparables aux PONAs d'US-AID dans leur utilité et leur fonctionnement, ils sont plus compact et représentent donc une charge réseau moindre.
Toutes les exécutions des opérations cryptographiques de ces scénarios sont simulées à l'aide des délais détaillés dans la table 4.2 en utilisant des valeurs tirées de SEDA [23] et des benchmarks réalisés par wolfCrypt [70]. L'ensemble des simulations sont réalisées sur une période limitée à 86400 secondes, soit 24 heures, ce qui constitue un aperçu représentatif des cycles d'attestation qui ont lieu dans le réseau.

TABLE 4.2 – Délais utilisés pour simuler les opérations cryptographiques dans Omnet++

Fonction	HMAC (SHA-256)	Chiffrement (AES)	ECDSA	PRNG
Délai (ms)	$0,4$	$0,4$	$347,2$	$3,8$

Les durées des périodes entre deux attestations ou encore deux heartbeats sont choisies arbitrairement, mais de façon à correspondre à un cas d'usage réaliste. Ainsi, les simulations de SEDA sont exécutées avec une attestation toutes les 3600 secondes (ce qui revient à 23 attestations sur la durée de la simulation). CRAFT est comparé à SEDA avec deux ensembles de paramètres appelés CRAFT+SEDA A et CRAFT+SEDA B, qui sont tous les deux représentés dans la figure 4.6. Ces paramètres sont aussi présentés dans la table 4.3.
CRAFT+SEDA A est exécuté avec une attestation toutes les 3600 secondes et un hearbeat toutes les 3600 secondes, soit une communication toutes les 1800 secondes et un doublement du nombre de fois où la présence d'un appareil est vérifiée, diminuant ainsi le risque qu'un attaquant puisse attaquer physiquement un appareil déconnecté.
CRAFT+SEDA B est exécuté avec des attestations toutes les 7200 secondes et des heartbeats toutes les 2400 secondes, soit également une communication toutes les 1800 secondes tout en réduisant le volume de données échangées significativement, mais diminuant le niveau de sécurité en comparaison.

TABLE 4.3 – Paramètres de scénarios pour SEDA et CRAFT+SEDA

Paramètres	SEDA	CRAFT+SEDA A	CRAFT+SEDA B
Période d'attestation (s)	3600	3600	7200
Période des heartbeats (s)	-	3600	2400

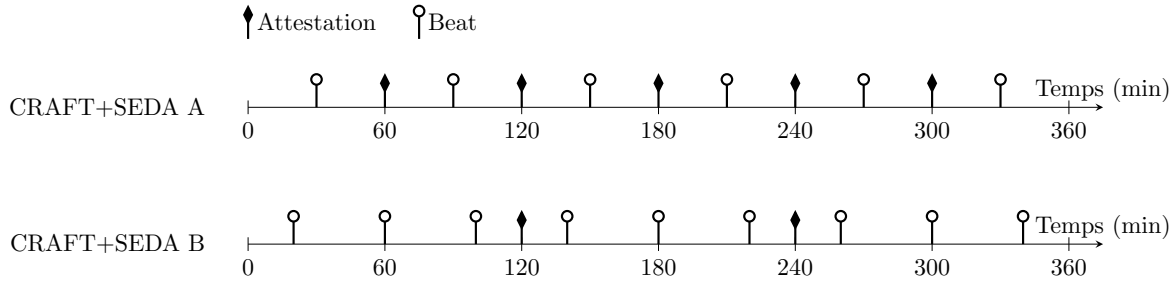


FIGURE 4.6 – Timings des attestations et heartbeats dans les simulations CRAFT+SEDA A et B

Les simulations d'US-AID sont également exécutées avec une attestation toutes les 3600 secondes (ce qui revient à 23 attestations sur la durée de la simulation) et des heartbeats toutes les 1100 secondes. CRAFT est comparé à US-AID avec deux ensembles de paramètres appelés CRAFT+US-AID A et CRAFT+US-AID B, ainsi que présenté dans la table 4.4. Les deux ensembles de paramètres de CRAFT+US-AID A et CRAFT+US-AID B utilisent les mêmes intervalles d'attestations et de heartbeats ; mais le premier a pour valeur du paramètre h_i (soit le TTL, valeur après laquelle le message ne sera plus retransmis) $h_i = 3$, tandis que le second a la valeur $h_i = 1$. La valeur 3 permet de démontrer la flexibilité de CRAFT concernant la mobilité, tandis que la valeur 1 permet de comparer US-AID et CRAFT+US-AID avec des fonctionnalités les plus équivalentes possibles afin de montrer plus précisément les différences de performance.

TABLE 4.4 – Paramètres de scénarios pour US-AID et CRAFT+US-AID

Paramètre	US-AID	CRAFT+US-AID A	CRAFT+US-AID B
Période d'attestation (s)	3600	3600	3600
Période des heartbeats (s)	1100	1100	1100
h_i (nombre de sauts)	-	3	1

4.3.3 Évaluation des performances

Dans cette section, les métriques précédemment détaillées — le volume de données et le nombre d'opérations cryptographiques — sont comparées entre le framework accompagné du protocole d'attestation et le protocole d'attestation seul. Tout d'abord, SEDA va être comparé à CRAFT+SEDA A et B. Cette comparaison démontrera que le framework proposé est bien plus flexible que SEDA. CRAFT permet soit de fournir un meilleur niveau de sécurité avec un minimum d'impact sur les performances, soit d'améliorer les performances tout en fournissant un niveau de sécurité égal ou supérieur à SEDA. US-AID va ensuite être comparé à CRAFT+US-AID A et B, démontrant que bien que les deux implémentations aient les même fonctionnalités de base et des

niveaux de sécurité et de confiance équivalents, CRAFT permet d'échanger un volume de données moindre et a donc un impact réduit sur l'utilisation du réseau.

SEDA vs CRAFT+SEDA

Les deux implémentations partagent le même mécanisme d'attestation, mais SEDA ne possède pas de heartbeat : cela signifie que CRAFT améliore la sécurité en échange d'un compromis de performances. Cependant, pour un niveau équivalent de confiance et de sécurité, la période d'attestation peut être réduite, car la continuité de l'attestation est garantie par les messages `beat`, résultant en une perte de performances moindre.

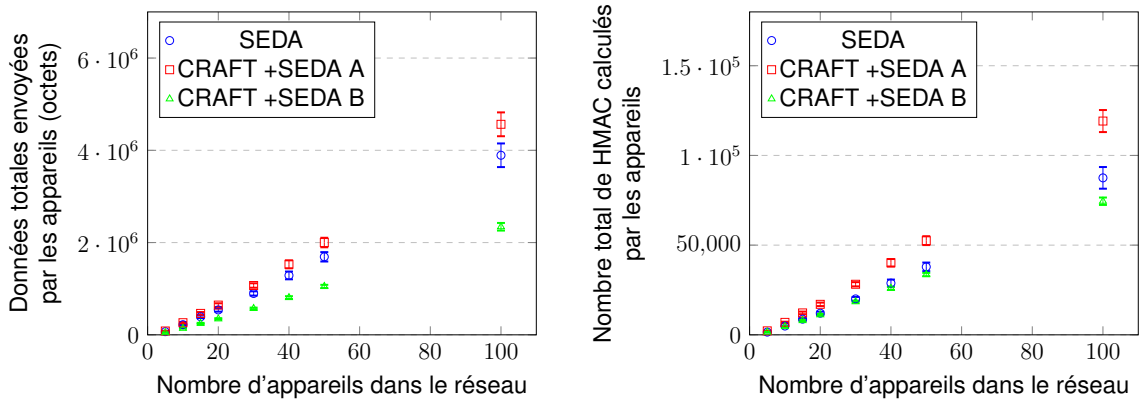
TABLE 4.5 – Comparaison des différences entre CRAFT+SEDA et SEDA

Métrique	Version de CRAFT+SEDA	% d'attestations	Différence de CRAFT
Volume de données	A	100%	$+18,3\%$
	B	50%	$-38,0\%$
Nombre de HMAC	A	100%	$+38,6\%$
	B	50%	$-11,3\%$

Comme résumé table 4.5, la figure 4.7(a) montre que, pour la même période d'attestation CRAFT+SEDA A échange un volume de données supérieur de 18,3% à SEDA. En revanche, en remplaçant la moitié des attestations par des messages `beat` dans CRAFT+SEDA B, ce volume de données est inférieur de 38,0% par rapport à SEDA. De la même manière, la figure 4.7(b) montre que, pour la même période d'attestation CRAFT+SEDA A a un nombre de HMAC supérieur de 38,6% à SEDA, mais inférieur de 11,3% pour CRAFT+SEDA B par rapport à SEDA en réduisant le nombre d'attestations de moitié.

Ainsi, en fonction du contexte de déploiement, CRAFT peut être paramétré pour obtenir un équilibre satisfaisant entre sécurité et performance.
La mobilité n'est pas évaluée dans la comparaison à SEDA, car ce protocole d'attestation seul ne possède pas de mécanisme permettant la reconnexion des nœuds mobiles au réseau.

Ces résultats démontrent que CRAFT est bien plus flexible que SEDA seul grâce à ses possibilités de paramétrage tout en étant meilleur en terme de sécurité, de performance ou des deux en fonction du contexte de déploiement et des paramètres sélectionnés pour CRAFT.



(a) Comparaison du volume de données **(b)** Comparaison du nombre de HMAC

FIGURE 4.7 – Comparaison de SEDA avec CRAFT+SEDA A et B concernant le volume de données envoyées et les HMAC calculés par les appareils

US-AID vs CRAFT+US-AID

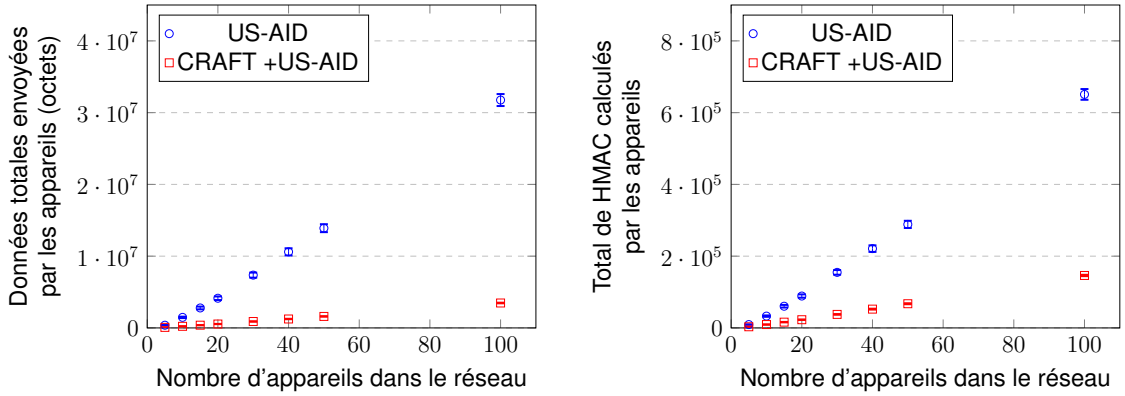
Les deux implémentations partagent le même protocole d'attestation mais ont différents mécanismes de heartbeat. US-AID fournit les PONAs, qui sont similaires aux messages `beat`. Bien que US-AID et CRAFT envoient le même nombre de messages, chaque message est bien plus petit dans CRAFT. Cependant, CRAFT utilise également les messages `lost` afin de permettre aux appareils de se déplacer plus loin, ce qui augmente le nombre de messages échangés.

TABLE 4.6 – Comparaison de la différence entre CRAFT+US-AID et US-AID

Métrique	Scénario	Différence de CRAFT
Volume de données	Statique	$-88,5\%$
	Mobile ($h_i = 3$)	$-36,1\%$
	Mobile ($h_i = 1$)	$-80,1\%$
Nombre de HMAC	Statique	$-76,6\%$
	Mobile ($h_i = 3$)	$+16,9\%$
	Mobile ($h_i = 1$)	$-66,0\%$

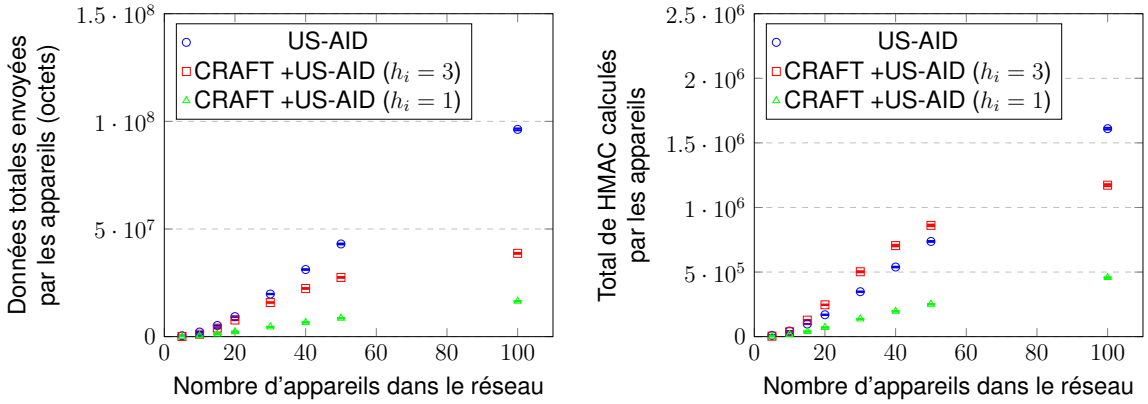
Comme résumé table 4.6, ces éléments sont vérifiés dans les figures 4.8(a)-4.9(b). La figure 4.8(a) montre que CRAFT utilise 88,5% de données en moins que US-AID dans une configuration statique. La figure 4.9(a) montre que CRAFT utilise 36,1% de données en moins que US-AID dans une configuration mobile avec $h_i = 3$ et 80,1% avec $h_i = 1$.

En ce qui concerne les HMAC, CRAFT est bien plus performant que US-AID dans un scénario où les appareils sont statiques avec 76,6% de HMAC en moins. Ce n'est pas



(a) Comparaison du volume de données **(b)** Comparaison du nombre de HMAC

FIGURE 4.8 – Comparaison de US-AID avec CRAFT+US-AID dans un réseau statique concernant le volume de données et les HMAC calculés



(a) Comparaison du volume de données **(b)** Comparaison du nombre de HMAC

FIGURE 4.9 – Comparaison de US-AID avec CRAFT+US-AID dans un réseau mobile concernant le volume de données et les HMAC calculés

le cas pour les réseaux mobiles en raison des messages `lost` car ces messages sont propagés plus loin : CRAFT réalise ainsi 16,9% de calculs de HMAC supplémentaires. Cependant, lorsque h_i est fixé à 1, les deux protocoles supportent le même niveau de mobilité et CRAFT et en réalité plus efficace, réalisant 66,0% de calculs de HMAC en moins avec la propagation réduite des messages `lost`.

Enfin, dans les cas où les appareils sont mobiles, il est possible de comparer combien d'entre eux ne parviennent pas à rester dans le réseau tout au long de la simulation, comme illustré figure 4.10. Cette figure montre que les nœuds avec US-AID sont plus souvent exclus du réseau que ceux avec CRAFT : sur le scénario comportant 50 nœuds et sur 100 simulations, US-AID a exclu $2,07 \pm 0,42$ nœuds en moyenne, tandis que CRAFT a exclu seulement $0,27 \pm 0,11$ en moyenne pour les deux valeurs de h_i (les valeurs moyennes de nœuds perdus sont les mêmes pour $h_i = 1$ et $h_i = 3$

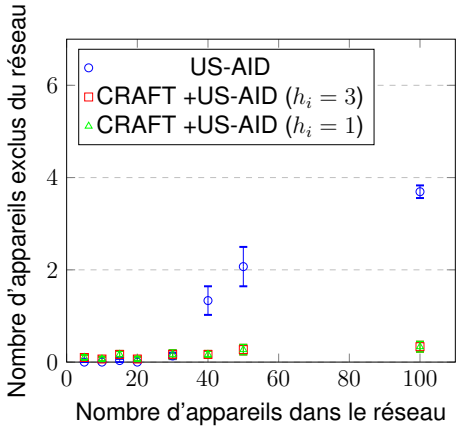


FIGURE 4.10 – Comparaison de US-AID avec CRAFT+US-AID dans un réseau mobile concernant les nœuds incorrectement exclus du réseau

dans les scénarios testés).

Dans le scénario de réseau comportant des nœuds mobiles, les données évoluent en

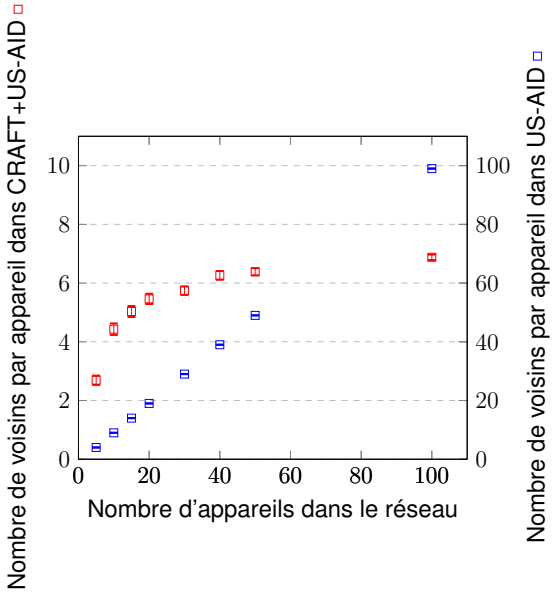


FIGURE 4.11 – Évolution logarithmique du nombre de voisins par nœud dans CRAFT+US-AID dans un réseau mobile

suivant ce qui apparaît être une courbe logarithmique qui évolue avec le nombre de nœuds observés, tandis que les valeurs de US-AID augmentent de manière linéaire, comme le montre la figure 4.9(b) avec CRAFT+US-AID ($h_i = 3$). Ceci s'explique par le fait qu'US-AID conserve la trace de tous les voisins précédents d'un nœud durant toute la durée de vie du réseau tandis que CRAFT ne conserve que les voisins actifs afin de

limiter l'empreinte mémoire globale du framework et donc le risque de déni de service. Ceci signifie que dans le cas du scénario à 50 nœuds, chaque nœud d'US-AID envoie des PONAs contenant l'information d'environ 49 autres nœuds, tandis que chaque nœud de CRAFT+US-AID n'envoie des messages `beat` et `lost` qu'à 6,4 voisins en moyenne. Pour le scénario à 100 nœuds, le nombre de nœuds évolue de manière linéaire pour US-AID (de 49 à 99) tandis qu'il évolue de manière logarithmique pour CRAFT+US-AID (de 6,4 à 6,9) tel qu'illustré figure 4.11.

Sur la base de ces résultats, il est possible d'affirmer que CRAFT est meilleur qu'US-AID en ce qui concerne les performances de calcul et la gestion de la mobilité, pour un niveau de sécurité équivalent.

Ainsi, les deux comparaisons démontrent bien que l'utilisation de CRAFT offre de multiples avantages en comparaison à l'utilisation des protocoles existants seuls, en particulier grâce à sa grande flexibilité, son niveau de sécurité, et de ses performances. Les simulations qui comparent CRAFT et SEDA mettent en lumière l'utilité des messages `beat` en ce qui concerne l'amélioration du niveau de sécurité qui se fait aux prix d'une augmentation du volume de données limitée.
Les simulations avec US-AID quant à elles montrent que CRAFT est parfaitement adapté à une utilisation dans des réseaux mobiles et permet une forte mobilité tout en n'excluant pas autant de nœuds qu'US-AID, grâce à l'utilisation de routes de confiance dans le réseau. En effet, ces routes de confiance aident les nœuds à se reconnecter au réseau à tout moment et n'importe où tandis qu'US-AID ne permet qu'une mobilité de proches en proches.

4.4 Évaluation de CRAFT dans un contexte comportant plusieurs protocoles d'attestation

Dans cette section, CRAFT est comparé dans ses versions avec et sans les fonctionnalités ASMP a un pseudo-framework composé des deux protocoles SEDA et US-AID. Ces frameworks, les appareils qui les composent, ainsi que l'environnement et les scénarios dans lesquels ces appareils évoluent sont présentés dans la section 4.4.1. Les métriques étudiées ainsi que l'application de la méthodologie déjà introduite dans la section 4.3 sont ensuite présentées dans la section 4.4.2. Enfin, les résultats des simulation sont présentés et analysés dans la section 4.4.3.

4.4.1 Description des frameworks, de l'environnement, des appareils et des scénarios

Frameworks

Trois frameworks différents sont définis pour les simulations :

— CRAFT-ASMP : il s'agit de la nouvelle version de CRAFT qui inclue les fonctionnalités ASMP (Adaptive Simultaneous Multi-Protocols) décrites en section 3.4.3. Grâce à ces fonctionnalités, certains nœuds peuvent utiliser différents protocoles d'attestation en fonction du contexte. Toutes les fonctionnalités de CRAFT sont disponibles pour tous les appareils.

— CRAFT : il s'agit du framework CRAFT présenté précédemment, sans les fonctionnalités ASMP. Contrairement à CRAFT-ASMP, tous les nœuds n'utilisent qu'un seul protocole d'attestation tout au long de la simulation, bien que plusieurs protocoles soient supportés en parallèle.

— SEDA+AID : les attestations de SEDA et d'US-AID coexistent au sein d'un même réseau, mais il n'y a pas de connexion directe entre les deux. Les nœuds du *Cœur de réseau* peuvent utiliser les deux protocoles, mais les nœuds avec des protocoles différents ne peuvent pas interagir. SEDA+AID est considéré comme un framework pour permettre son évaluation et sa comparaison à CRAFT-ASMP et CRAFT.

Environnement de simulation

L'environnement dans lequel les scénarios prennent place est une smart city, cette fois représentée par une grille. Cet environnement se veut plus représentatif d'un cas réel dans les déplacements des appareils et dans leur paramétrage, contrairement à l'environnement décrit dans la section précédente qui se veut plus général.
La smart city utilisée est similaire à celle représentée figure 4.12. Ici, seule une partie représentative de l'environnement de simulation est illustrée. La vue du monde réel représente les appareils physiques tels que les bâtiments, les véhicules ou encore les éclairages intelligents. Ces mêmes appareils et leurs canaux de communication sont représentés dans la vue logique.
L'utilisation d'une telle représentation de smart city permet de valider par la pratique les forces de CRAFT, et en particulier des fonctionnalités ASMP. En effet, les nœuds sont très hétérogènes comme ils le seraient dans la réalité, tant par leur mobilité que par leurs performances en terme de calcul et de sécurité.
Les caractéristiques complètes de l'environnement de simulation sont décrites dans la table 4.7. En particulier, la smart city est représentée par un carré de 6 par 6 blocs

carrés plus petits. Chaque bloc fait 100 mètres par 100 mètres, ce qui donne une taille de simulation de 600 mètres par 600 mètres.

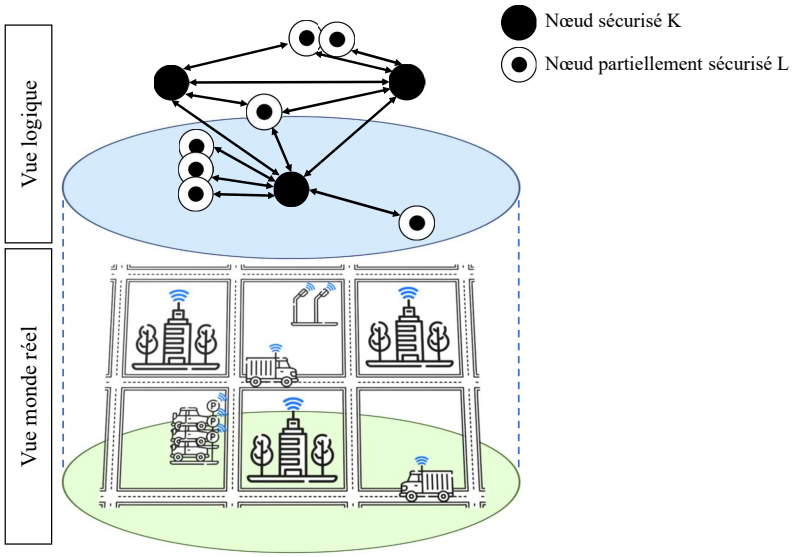


FIGURE 4.12 – Environnement de simulation représenté avec une vue du monde réel et une vue logique

Appareils Dans tous les scénarios, les nœuds placés dans la smart city sont de trois types :

— Des **antennes**, qui ne se déplacent pas sur la grille et sont placées de sorte à couvrir la totalité de la grille en terme de champs de communication. Dans CRAFT-ASMP et CRAFT, ces antennes sont des nœuds K qui forment le *Cœur de réseau*.

— Des **appareils statiques**, placés de manière aléatoire sur la grille. Ils illustrent des objets connectés tels que des éclairages intelligents ou des capteurs de pollution d'air connectés. Dans CRAFT-ASMP et CRAFT, ces appareils sont des nœuds L qui prennent part au *Réseau externe*.

— Des **appareils mobiles**, qui suivent les lignes de la grille (les routes). Ils sont placés à des intersections aléatoires au début de la simulation, se déplacent le long des lignes, puis choisissent une direction aléatoire (avant, arrière, gauche ou droite) lorsqu'ils arrivent à la prochaine intersection. Ces appareils simulent des véhicules. Dans CRAFT-ASMP et CRAFT, ces appareils sont aussi des nœuds L qui prennent part au *Réseau externe*. Ce déplacement diffère de celui présenté en section 4.3 et présente l'intérêt d'être une meilleure approximation

d'un environnement smart city, dans lequel les véhicules se concentrent sur des axes de circulation.

Scénarios

Afin de tester différents aspects des frameworks, trois scénarios sont utilisés, et leurs paramètres de simulation dans Omnet++ sont listés dans la table 4.7. Les paramètres de scénarios spécifiques sont listés dans la table 4.8. Les trois scénarios sont :

— Le scénario *Basique* consiste uniquement en la smart city, dans laquelle tous les appareils observent leur comportement standard en étant soit fixes, soit mobiles, et en communiquant en continu. Ce scénario est utilisé comme point de comparaison avec les autres scénarios.

— Le scénario *Parking* est similaire au scénario *Basique* à l'exception d'une phase de parking qui est ajoutée. Durant cette phase, les nœuds mobiles cessent de se déplacer sur la grille pendant un certain temps pour simuler le fait que des véhicules soient garés.

— Le scénario *Parking+Out* ajoute un temps d'exclusion au scénario *Parking* : certains nœuds mobiles et statiques cessent de communiquer avec le réseau, et tentent de se reconnecter plus tard. La moitié de ces nœuds a un temps d'exclusion court de $2000s$, choisi inférieur à $T_{a_i} = 3600s$, qui simule une déconnexion involontaire et sont supposés se reconnecter correctement au réseau ensuite. L'autre moitié a un long temps d'exclusion de $7200s$ qui simule une attaque des appareils, et ces nœuds ne sont pas supposés pouvoir se reconnecter au réseau car cette durée excède la valeur de $T_{a_i} = 3600s$.

En résumé, dans cet environnement smart city, chacun des trois frameworks sera simulé sur chacun des trois scénarios, amenant à un total de neuf simulations et donc neuf ensembles de résultats à comparer dans le cadre de l'évaluation du framework CRAFT.

4.4.2 Métriques et méthodologie

Afin de démontrer l'ensemble des bénéfices apportés par CRAFT-ASMP, cinq métriques sont observées :

— **Exclusion de nœuds** : un bon framework a une détection suffisamment sensible pour exclure tous les nœuds compromis en maximisant le taux de vrais positifs. L'exclusion doit également être spécifique, et les nœuds qui ne sont pas compromis doivent être maintenus dans le réseau en minimisant le taux de faux positifs.

TABLE 4.7 – Paramètres de simulation d'Omnet++

Paramètre	Valeur
Hauteur et largeur de la grille (m)	600
Hauteur et largeur d'un bloc (m)	100
Densité des nœuds (nœuds par km^2)	278
Nombre d'antennes	18
Nombre de nœuds statiques	42
Nombre de nœuds mobiles	40
Portée des antennes (m)	300
Portée des autres nœuds (m)	200
Position des antennes	Choisie
Position des nœuds fixes	Aléatoire
Position initiales des nœuds mobiles	Aléatoire, aux intersections de la grille
Modèle de mobilité des nœuds mobiles	MovingMobilityBase personnalisée
Durée de la simulation (s)	86400
Modèle radio	UnitDiskRadio
Modèle d'interface sans-fil	WirelessInterface

TABLE 4.8 – Paramètres de scénario d'Omnet++

Paramètre		Valeur	Scénario		
			Basique	Parking	Parking+Out
	Période d'attestation - T_{a_i} (s)	3600	✓	✓	✓
	Période des heartbeats - T_{b_i} (s)	1100	✓	✓	✓
	Délai entre deux phases de parking (s)	14400		✓	✓
	Durée des phases de parking (s)	10800		✓	✓
Nœuds exclus sensés se reconnecter	Durée d'exclusion (s)	2000			✓
	Nombre de nœuds statiques exclus	10			✓
	Nombre de nœuds mobiles exclus	10			✓
Nœuds exclus sensés NE PAS se reconnecter	Durée d'exclusion (s)	7200			✓
	Nombre de nœuds statiques exclus	10			✓
	Nombre de nœuds mobiles exclus	10			✓

— **Nombre d'attestations** : les attestations dépendent du protocole d'attestation utilisé et le nombre d'attestations doit être en corrélation avec la fréquence définie dans les paramètres. Pour SEDA, le nombre d'attestations doit toujours être le même (par exemple 23 dans une simulation de 24 heures avec des attestations toutes les heures). Avec US-AID, les appareils envoient une attestation par voisin à chaque phase d'attestation, le nombre d'attestations envoyées dépend donc du nombre de nœuds voisins, et des différences entre appareils ou répétitions peuvent alors s'expliquer. CRAFT et US-AID n'ayant pas la même définition de nœud voisin, ceci peut également expliquer des différences de valeur. Chaque nœud doit avoir au moins une attestation par période d'attestation indépendam-

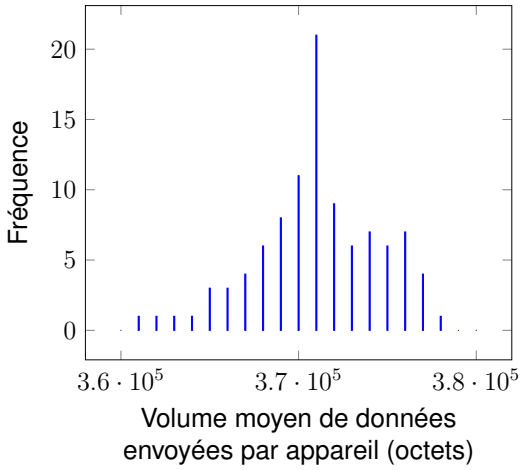
ment du protocole d'attestation.

— **Nombre de heartbeats** : les heartbeats aident à améliorer l'attestation continue en augmentant la fréquence de communication mais avec un impact moindre que les messages d'attestation. Comme expliqué précédemment section 3.4, plus il y a de heartbeats par appareil, plus l'attestation continue est renforcée et donc la confiance dans la sécurité du réseau améliorée.

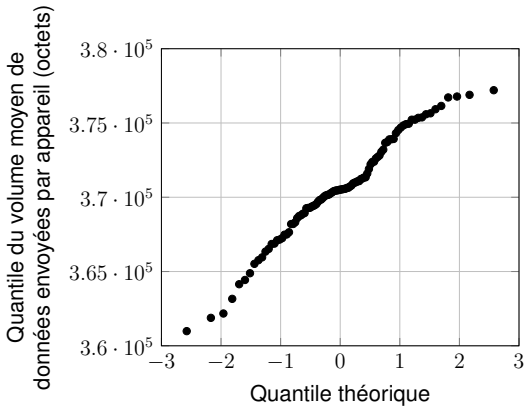
— **Volume de données moyen** : tous les frameworks doivent minimiser leur impact sur les appareils, et le volume de données échangées par le framework est un point de comparaison. Envoyer et recevoir des données est une source de consommation d'énergie et les frameworks doivent minimiser ces coûts.

— **Calculs de HMAC** : comme pour le volume de données, la cryptographie a un impact sur les performances des appareils et leur consommation d'énergie. Ces calculs doivent être minimisés, mais pas au détriment de la sécurité.

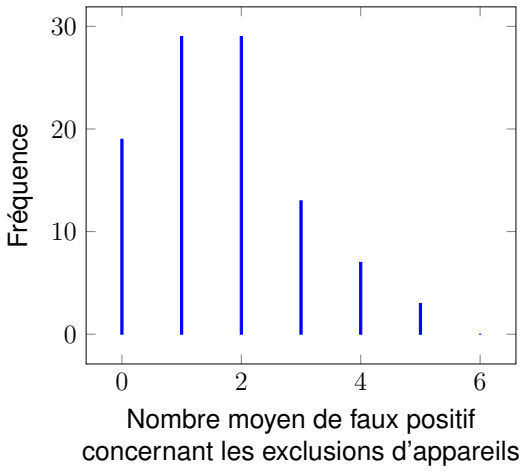
La même méthodologie que présentée en section 4.3.1 page 73 peut être appliquée ici à l'ensemble des résultats présentés pour chaque métrique des neuf simulations. À nouveau, pour des raisons de clarté et de concision, l'application de cette méthodologie n'est présentée que pour deux métriques : le volume de données envoyées moyen par les nœuds dans le scénario *Basique* utilisant le framework CRAFT-ASMP, et le taux de faux positifs pour les exclusions de nœuds mobiles dans le scénario *Parking+Out* utilisant le framework CRAFT-ASMP.
Ces mesures sont représentées figure 4.13.
L'histogramme de la figure 4.13(a) est très proche d'une distribution normale, et est seulement légèrement asymétrique. De plus, le diagramme Q-Q de la figure 4.13(b) suit majoritairement une ligne droite avec seulement quelques irrégularités qui n'empêchent pas l'application de la méthodologie.
De la même manière, l'histogramme de la figure 4.13(c) est proche d'une distribution normale. Cependant, le fait que cette métrique soit discrète avec un minimum de 0 rend ce fait moins évident car l'histogramme n'est pas symétrique à gauche. Le diagramme Q-Q de la figure 4.13(d) montre la même information avec une ligne droite, à l'exception du premier point.
Ici aussi, il est donc possible d'utiliser le Théorème Centrale Limite, car les données suivent de façon proche une distribution normale : comme les simulations sont répétées 100 fois chacune, les résultats sont bien approximés par une distribution normale, et des intervalles de confiance peuvent être calculés en utilisant la distribution t de Student, et les résultats présentés dans la section suivante peuvent être facilement vérifiés et répétés.



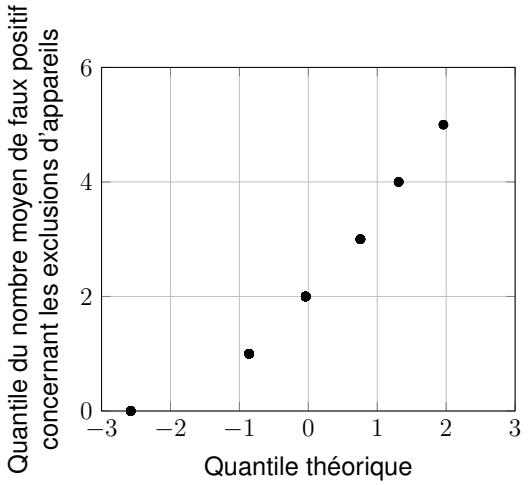
(a) Fréquence du volume moyen de données envoyées par les appareils dans le scénario Basique utilisant le framework CRAFT-ASMP sur 100 répétitions



(b) Diagramme Q-Q du volume moyen de données envoyées par les appareils dans le scénario Basique utilisant le framework CRAFT-ASMP sur 100 répétitions



(c) Fréquence du taux de faux positifs concernant les exclusions d'appareils mobiles dans le scénario *Parking+Out* utilisant le framework CRAFT-ASMP sur 100 répétitions



(d) Diagramme Q-Q du taux de faux positifs concernant les exclusions d'appareils mobiles dans le scénario *Parking+Out* utilisant le framework CRAFT-ASMP sur 100 répétitions

FIGURE 4.13 – Graphes montrant que les résultats moyens des simulations suivent approximativement une distribution normale

4.4.3 Résultats

Les métriques précédemment mentionnées (c'est-à-dire l'exclusion des nœuds, le nombre d'attestations, le nombre de heartbeats, le volume de données moyen et les calculs de HMAC) sont analysées dans cette section. Ces résultats démontrent que CRAFT-ASMP améliore significativement CRAFT et SEDA+AID de plusieurs façons. En effet, CRAFT-ASMP démontre sa sécurité en excluant les appareils compromis mieux que les autres frameworks. De plus, CRAFT-ASMP démontre de meilleures performances que les autres frameworks en ce qui concerne les calculs et la communication.

Exclusion de nœuds

Scénario	Framework	Taux d'exclusion des nœuds mobiles		Taux d'exclusion des nœuds statiques	
		Vrais positifs	Faux positifs	Vrais positifs	Faux positifs
Basique	CRAFT-ASMP	-	$0,20\% \pm 0,15\%$	-	-
	CRAFT	-	$0,20\% \pm 0,15\%$	-	-
	SEDA+AID	-	$24,50\% \pm 1,37\%$	-	-
Parking	CRAFT-ASMP	-	$5,05\% \pm 0,68\%$	-	-
	CRAFT	-	$5,35\% \pm 0,73\%$	-	-
	SEDA+AID	-	$22,58\% \pm 1,51\%$	-	-
Parking+Out	CRAFT-ASMP	$100\% \pm 0\%$	$5,63\% \pm 0,85\%$	$100\% \pm 0\%$	$0\% \pm 0\%$
	CRAFT	$100\% \pm 0\%$	$5,67\% \pm 0,90\%$	$100\% \pm 0\%$	$0\% \pm 0\%$
	SEDA+AID	$100\% \pm 0\%$	$47,90\% \pm 1,23\%$	$0\% \pm 0\%$	$0\% \pm 0\%$

TABLE 4.9 – Comparaison des frameworks sur trois scénarios concernant l'exclusion des nœuds mobiles et statiques

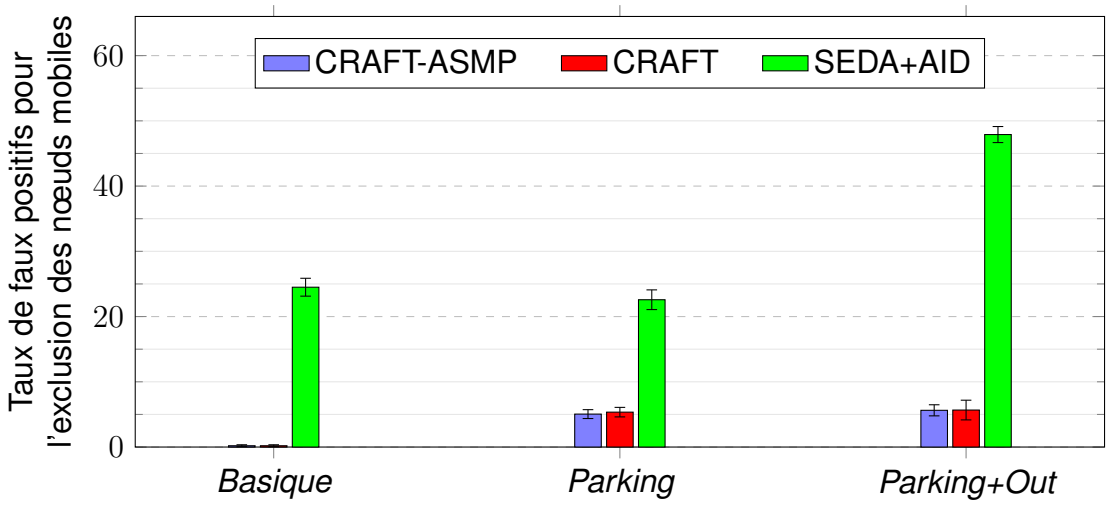


FIGURE 4.14 – Comparaison des frameworks sur trois scénarios concernant le taux de faux positifs pour l'exclusion des nœuds mobiles

Surveiller l'exclusion des appareils est critique pour assurer le niveau de sécurité d'un

framework d'attestation, car tous les nœuds compromis doivent être exclus tout en maintenant les autres nœuds dans le réseau.
Dans la table 4.9, les taux de vrais positifs et de faux positifs utilisés comme référence sont présentés.

— Un vrai positif ne peut avoir lieu que pour des nœuds exclus dans le scénario *Parking+Out* car les appareils ne simulent pas d'exclusion dans les autres scénarios. Les résultats montrent qu'en utilisant CRAFT, avec ou sans les fonctionnalités ASMP, 100% des appareils attaqués sont exclus, que ce soient les fixes utilisant SEDA ou les mobiles utilisant US-AID. En revanche en ce qui concerne SEDA+AID, étant donné que SEDA ne réalise pas d'exclusion de nœuds sur ces critères, seuls les nœuds mobiles utilisant US-AID sont exclus.

— Les faux positifs peuvent avoir lieu pour les nœuds mobiles qui se déplacent hors de portée du réseau et ne sont pas capables de se reconnecter suffisamment rapidement (c'est-à-dire avant le seuil fixé dans le scénario). Les résultats présentés dans la table 4.9 et illustrés figure 4.14 montrent que les deux implémentations de CRAFT se comportent de manière semblable, tandis que SEDA+AID exclue incorrectement de plus nombreux nœuds.

Ces meilleurs résultats pour CRAFT-ASMP et CRAFT sont expliqués par l'introduction des paquets `lost` de CRAFT qui permettent aux appareils de rejoindre le réseau plus facilement que dans SEDA+AID.

Attestations et heartbeats

Les attestations et les heartbeats définissent à quelle fréquence l'état et la présence d'un appareil sont vérifiés par le réseau. Un appareil doit être vérifié de manière suffisamment régulière pour maintenir la confiance du réseau à son égard. Ainsi, plus il y a d'attestations et de heartbeats, meilleure est la sécurité. Cependant, ces messages ne doivent pas ajouter trop de coût en communication et en calculs.
La figure 4.15 montre combien d'attestations d'US-AID sont échangées parmi les nœuds mobiles pour tous les scénarios.

— Dans le scénario *Basique*, le framework SEDA+AID a plus d'attestations par nœud que CRAFT et CRAFT-ASMP car le protocole US-AID montre expérimentalement que chacun de ses nœuds a un plus grand nombre de voisins que ceux de CRAFT, et les attestations sont envoyées à tous les voisins utilisant US-AID.

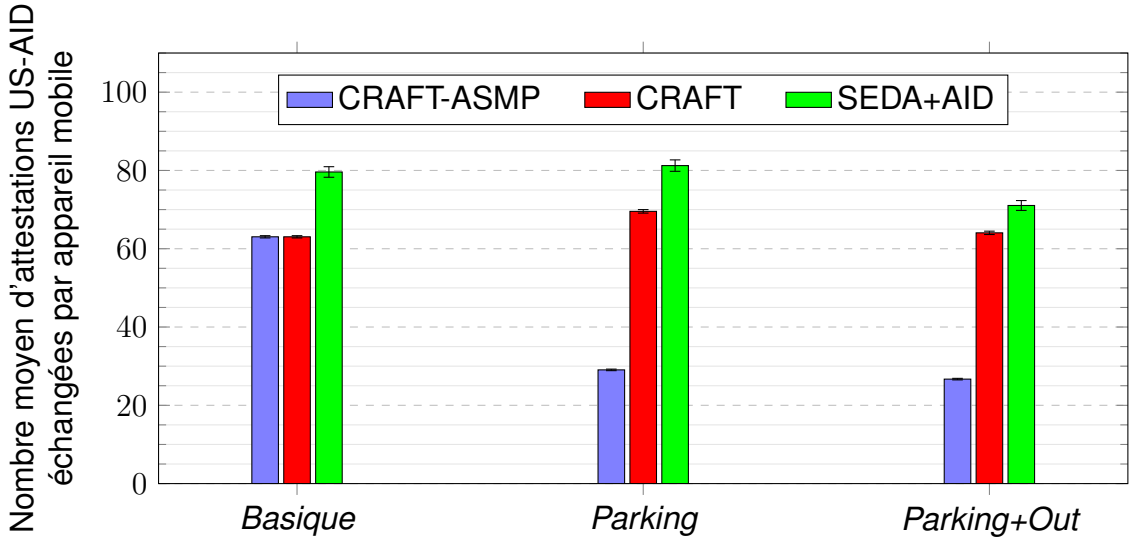


FIGURE 4.15 – Comparaison des frameworks sur trois scénarios concernant le nombre d'attestations US-AID

— Dans les scénarios *Parking* et *Parking+Out*, les nœuds mobiles de CRAFT-ASMP basculent sur le protocole SEDA quand ils sont garés, ce qui représente $37,5\%$ de la durée de simulation, ils ont donc moins d'attestations US-AID que dans le scénario *Basique*. Ceci explique pourquoi le nombre d'attestations pour CRAFT-ASMP dans le scénario *Parking* est inférieur de $46,1\%$ par rapport au nombre d'attestations pour CRAFT-ASMP dans le scénario *Basique* (de $29,05$ à $63,05$). Ceci n'a pas lieu pour les autres frameworks, car ils ne basculent pas d'un protocole d'attestation à un autre.

— Pour l'ensemble des trois scénarios, CRAFT et SEDA+AID restent à des niveaux d'attestations équivalents. Cependant, le scénario *Parking* a des différences respectives de $+10,3\%$ (de $63,05$ à $69,55$) pour CRAFT et $+2,0\%$ (de $79,61$ à $81,24$) pour SEDA+AID dans le nombre d'attestations en comparaison au scénario *Basique*. De manière similaire, le scénario *Parking+Out* a des différences de $+1,6\%$ (de $63,05$ à $64,06$) pour CRAFT et $-10,8\%$ (de $79,61$ à $71,05$) pour SEDA+AID concernant le nombre d'attestations. Cette différence est expliquée par la combinaison de nœuds garés et exclus. Les différents frameworks en eux-mêmes n'ont donc pas d'impact significatif sur le nombre d'attestations pour un scénario équivalent.

Ainsi, en comparant CRAFT-ASMP à CRAFT ou SEDA+AID en ce qui concerne les attestations, l'utilité des fonctionnalités ASMP apparaît clairement en ce qu'elles permettent d'optimiser le rapport entre la sécurité et les performances.

Scénario	Framework	Nombre d'attestation SEDA	
		Nœuds statiques	Nœuds mobiles
Basique	CRAFT-ASMP	$23,00 \pm 0,00$	$0,00 \pm 0,00$
	CRAFT	$23,00 \pm 0,00$	-
	SEDA+AID	$23,00 \pm 0,00$	-
Parking	CRAFT-ASMP	$23,00 \pm 0,00$	$6,39 \pm 0,07$
	CRAFT	$23,00 \pm 0,00$	-
	SEDA+AID	$23,00 \pm 0,00$	-
Parking+Out	CRAFT-ASMP	$21,33 \pm 0,00$	$5,89 \pm 0,07$
	CRAFT	$21,33 \pm 0,00$	-
	SEDA+AID	$21,81 \pm 0,00$	-

TABLE 4.10 – Comparaison des frameworks sur trois scénarios concernant le nombre d'attestations SEDA

La table 4.10 montre le nombre d'attestations SEDA échangées parmi les nœuds statiques et mobiles dans tous les scénarios (quand c'est applicable).

— Pour les attestations SEDA, tous les nœuds statiques du scénario *Basique* et *Parking* ont réalisé 23 attestations. Étant donné que l'interprétation de ce résultat est directe, il n'est pas représenté dans une figure. Les nœuds CRAFT-ASMP dans les scénarios *Parking* et *Parking+Out* réalisent seulement $6,39 \pm 0,07$ et $5,89 \pm 0,07$ attestations SEDA en moyenne car ils utilisent US-AID en dehors des phases de parking.

— Dans le scénario *Parking+Out*, il y a également moins d'attestations SEDA par nœud en moyenne, à cause de l'exclusion des nœuds à un moment de la vie du réseau. Ce nombre est plus élevé sans CRAFT ($21,81 \pm 0,00$ sans pour $21,33 \pm 0,00$ avec) car les nœuds statiques exclus dans SEDA+AID continuent à participer au réseau et donc réalisent plus d'attestations.

Pour résumer la comparaison des nombres d'attestations, l'ensemble des protocoles réalise un nombre similaire d'attestations. Cependant, CRAFT-ASMP démontre sa supériorité en utilisant le protocole le plus approprié en fonction du contexte dans le but d'améliorer la sécurité du réseau dans son ensemble. En effet, les nœuds qui utilisent CRAFT-ASMP peuvent utiliser SEDA lorsqu'ils sont garés, étant donné que SEDA est plus approprié qu'US-AID pour les appareils statiques et démontre de meilleurs résultats dans ce contexte.

La figure 4.16 montre le nombre moyen de heartbeats envoyés par chaque nœud durant la durée de la simulation.

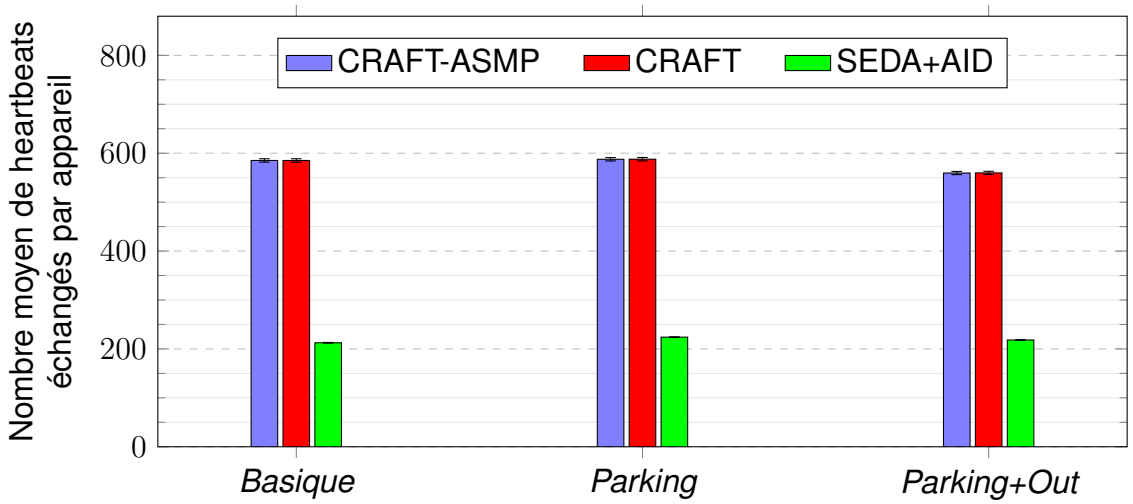


FIGURE 4.16 – Comparaison des frameworks sur trois scénarios concernant le nombre de heartbeats

— CRAFT-ASMP est aussi performant que CRAFT, car tous deux fournissent le même mécanisme de heartbeat à l'ensemble des nœuds. Fournir ce mécanisme est une fonctionnalité de sécurité essentielle qui vient compléter les attestations avec une vérification des nœuds plus légère mais aussi plus fréquente : c'est ce qui permet l'attestation continue.

— Au travers des trois scénarios, la différence entre les nombres de heartbeats de CRAFT-ASMP et CRAFT est de moins de $0,05\%$. Cela démontre que les fonctionnalités supplémentaires de CRAFT-ASMP n'ont pas d'impact sur les performances du mécanisme de heartbeat.

— En comparaison à SEDA+AID, CRAFT-ASMP montre toujours un nombre de heartbeats supérieur de plus de 60%. Ceci s'explique par le fait que seuls les nœuds US-AID (soit 40% des nœuds) réalisent des messages PONAs, qui sont similaires aux messages `beat` de CRAFT. Cela montre clairement les bénéfices de CRAFT-ASMP, car maintenir la continuité de l'attestation au travers des heartbeats permet de garder le réseau sécurisé.

Volume de données moyen et calculs de HMAC

Ces deux métriques sont corrélées avec la consommation d'énergie et le temps de calcul, car elles représentent l'essentiel de la charge de travail du framework. Elles permettent l'analyse de la performance générale des frameworks comparés.
La figure 4.17 montre le volume moyen de données envoyées par chaque nœud.

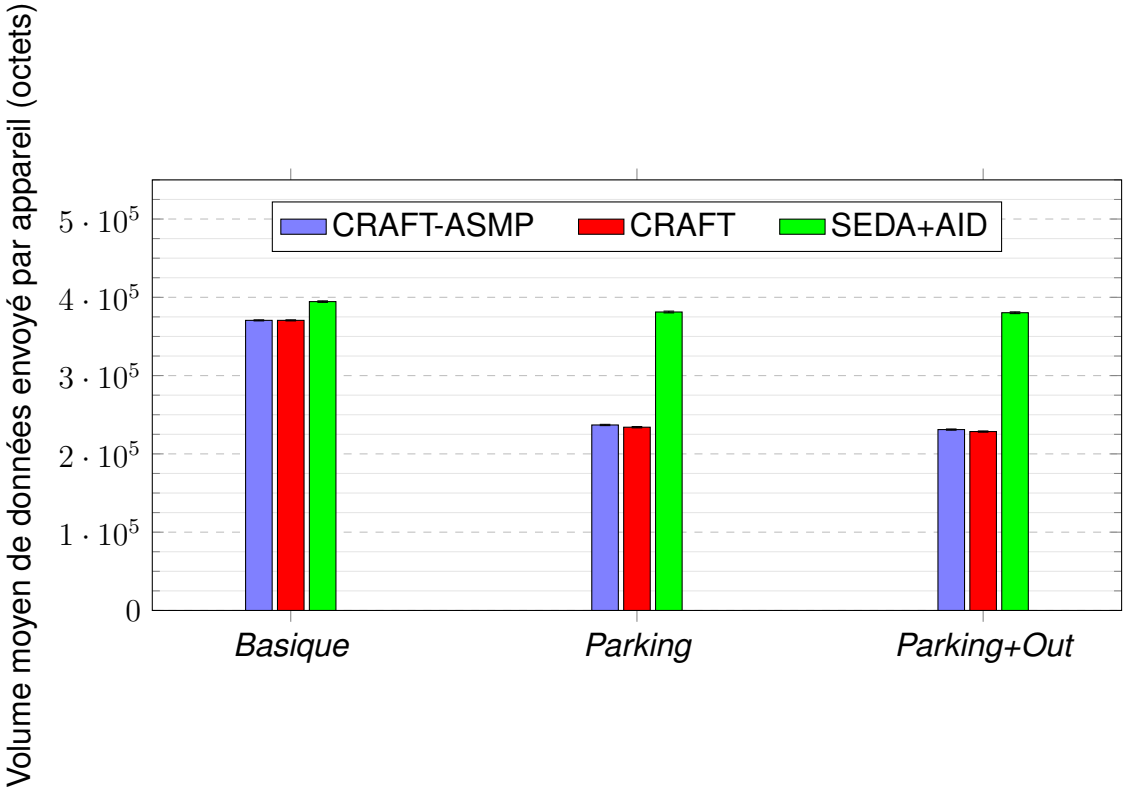


FIGURE 4.17 – Comparaison des frameworks sur trois scénarios concernant le volume de données envoyées

— Dans tous les scénarios, CRAFT-ASMP échange moins de données que SEDA+AID, car les messages utilisés sont plus petits. Dans le scénario *Basique*, la différence entre CRAFT-ASMP et SEDA+AID est de $6,5\%$ (de 394518 à 370561 octets).

— Dans les deux scénarios *Parking* et *Parking+Out*, le framework SEDA+AID a un volume de données supérieur de 60% comparé à CRAFT-ASMP (respectivement 381245 comparé à 236929 octets et 380323 comparé à 231053 octets). Ceci s'explique par le fait que CRAFT-ASMP prend avantage de la mobilité réduite par le parking : cela permet aux nœuds d'échanger moins de messages et de paquets `lost`, et réduit ainsi le volume de données envoyées.

— Dans le scénario *Parking*, CRAFT-ASMP échange $1,2\%$ plus de données que CRAFT (de 234083 à 236929 octets) ce qui s'explique par les nœuds mobiles passant dans l'état de parking et utilisant l'attestation SEDA. L'attestation SEDA est légèrement plus volumineuse que l'attestation US-AID, mais cela ne montre pas un impact significatif sur les performances dans ces simulations. Ainsi, ce résultat démontre que les nœuds CRAFT-ASMP mobiles garés peuvent basculer vers l'attestation SEDA, qui est plus efficace pour les appareils statiques, sans que cela n'impacte les performances.

Scénario	Framework	Ratio de données envoyées par type de nœud		
		Antenne	Statique	Mobile
Basique	CRAFT-ASMP	92,46%	3,66%	3,88%
	CRAFT	92,46%	3,66%	3,88%
	SEDA+AID	96,31%	1,75%	1,94%
Parking	CRAFT-ASMP	87,81%	5,72%	6,47%
	CRAFT	88,40%	5,79%	5,81%
	SEDA+AID	96,04%	1,81%	2,15%
Parking+Out	CRAFT-ASMP	87,94%	5,46%	6,60%
	CRAFT	88,51%	5,52%	5,97%
	SEDA+AID	96,22%	1,72%	2,06%

TABLE 4.11 – Ratio de données envoyées par les différent types de nœuds pour chaque framework sur trois scénarios

La table 4.11 illustre comment les volumes de données sont répartis entre les antennes, les nœuds statiques et les nœuds mobiles dans chaque framework. Les résultats montrent que les antennes sont la source principale de communication, représentant toujours plus de 87% du volume de données.

— La balance entre les antennes et les autres nœuds penchent plus en direction des autres nœuds avec l'utilisation de CRAFT-ASMP et CRAFT que l'utilisation de SEDA+AID (par exemple de $96,04\%$ pour SEDA+AID à $87,81\%$ pour CRAFT-ASMP et $88,40\%$ pour CRAFT dans le scénario *Parking*). Ceci s'explique par l'utilisation des messages `beat` et `lost` pour tous les appareils, ce qui augmente la communication à partir des nœuds (mobiles et statiques) au bénéfice d'une meilleure sécurité.

— Dans les scénarios *Parking* et *Parking+Out* comparés au scénario *Basique*, moins de messages `lost` sont envoyés étant donné que les appareils sont moins mobiles et donc les antennes sont moins sollicitées, ce qui explique le poids réduit des antennes dans le volume global de données envoyées (par exemple, pour CRAFT-ASMP de $92,46\%$ dans le scénario *Basique* à $87,81\%$ dans le scénario *Parking*).

La figure 4.18 illustre le nombre moyen de HMAC calculés par les nœuds, et permet de voir les différences d'impact des différents frameworks en performances à ce niveau. Ce nombre est fortement corrélé avec le nombre de heartbeats envoyés car c'est le type de message le plus fréquemment envoyé, mais prend également en considération l'ensemble des autres messages y compris les attestations des différents protocoles.

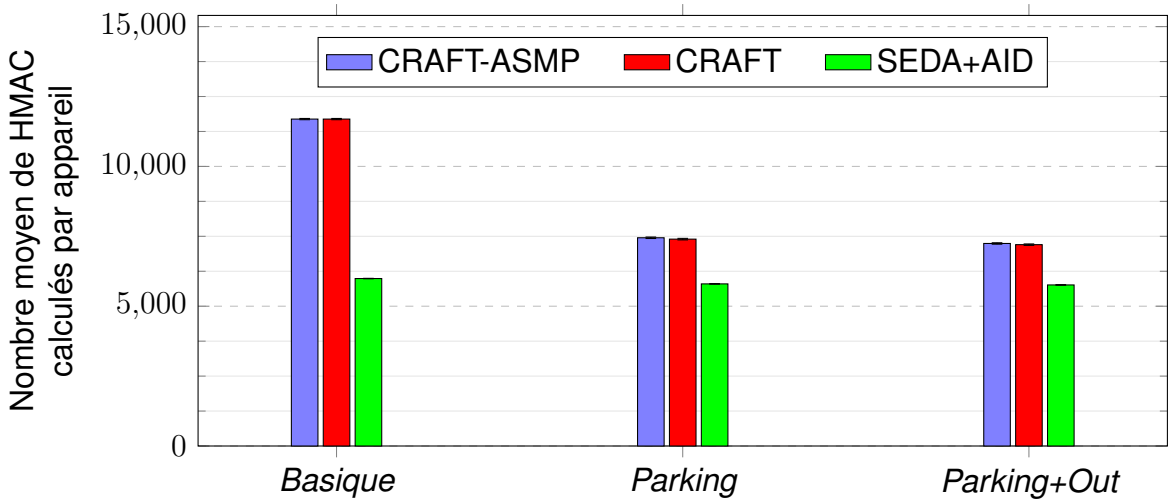


FIGURE 4.18 – Comparaison des frameworks sur trois scénarios concernant le nombre de calculs de HMAC

— CRAFT-ASMP montre des performances équivalentes à CRAFT dans tous les scénarios, même avec ses fonctionnalités supplémentaires. Comme CRAFT, CRAFT-ASMP est également plus efficace dans les scénarios où la mobilité est moindre (et donc dans lesquels moins de messages `lost` sont échangés) : le nombre moyen de HMAC calculés est réduit de plus de 36% (de 11693 à 7448) entre les scénarios *Basique* et *Parking*.

— Les valeurs de SEDA+AID montrent peu de variations dans le nombre moyen de HMAC calculés sur les trois scénarios. La différence avec le scénario *Basique* est de moins de 4% (de 5990 à 5798 pour le scénario *Parking* et à 5761 pour le scénario *Parking+Out*), ce qui s'explique par la mobilité réduite. La variation n'est pas plus grande car le framework ne réagit pas aux changements de contexte, contrairement à CRAFT-ASMP.

— Le framework SEDA+AID calcule toujours moins de HMAC que CRAFT-ASMP. L'excédent est de $48,8\%$ (de 5990 à 11693) pour CRAFT-ASMP comparé à SEDA+AID dans le scénario *Basique*, et de $22,2\%$ (de 5798 à 7448) pour CRAFT-ASMP comparé à SEDA+AID dans le scénario *Parking* (qui est très comparable au scénario *Parking+Out* sur ce point). Cet excédent est expliqué par la sécurité additionnelle que CRAFT-ASMP fournit au travers des messages `beat` pour tous les nœuds, tandis que pour le framework SEDA+AID seuls les nœuds équipés d'US-AID ont cette fonctionnalité (40% des nœuds, soit les nœuds mobiles).

4.5 Conclusion du chapitre

Dans ce chapitre, différents résultats expérimentaux basés sur des simulations dans le simulateur d'événements discrets Omnet++ ont été présentés. Ils illustrent les bonnes performances de CRAFT en terme de sécurité, d'adaptabilité et de performances. CRAFT a notamment été comparé dans un environnement général à des protocoles d'attestation seuls, SEDA et US-AID, qui sont parmi les plus reconnus dans la littérature sur l'attestation à distance. CRAFT et ses fonctionnalités ASMP ont ensuite été comparés au framework composé de la combinaison de SEDA et d'US-AID dans un scénario plus proche d'un cas réel de smart city, illustrant à nouveau les bonnes performances du framework d'attestation continue à distance proposé et de ses extensions.

5

Conclusion et pistes de recherches

Sommaire

La démocratisation des objets connectés et le développement de l'Internet des Objets ont apporté de nouvelles possibilités de visualisation de données et de contrôle à distance dans de nombreux domaines. On peut par exemple citer les capteurs d'hygrométrie et les contrôleurs d'arrosage dans le domaine du smart farming, les système de contrôle de trafic intelligents dans le domaine des smart city, ou encore les robots-chirurgiens contrôlés à distance dans le domaine de la smart health.
Tous ces objets connectés sont des objets physiques, qui viennent se placer en l'interface entre le monde réel et le monde informatique, et sont fréquemment la cible d'attaques informatiques. De part la nature de ces appareils, la surface d'attaque potentielle est large. En effet, les objets connectés reposent sur des composants matériels (microcontrôleurs, mémoires, capteurs, actionneurs, ...), possèdent généralement une interface homme-machine logicielle ou matérielle, et sont connectés à divers types de réseaux filaires ou sans-fil (Ethernet, Wi-Fi, LTE, LoraWan, ...).
Une des principales difficultés de la sécurisation des objets connectés est cette large surface d'attaque qui implique une diversité de solutions de sécurité, tout en respectant le besoin de maintenir un coût de production compétitif et de répondre à des contraintes de fonctionnement comme la consommation en énergie.
Parmi ces solutions de sécurité, l'attestation à distance permet de vérifier que les objets connectés qui composent un réseau sont dans un état de fonctionnement correct en ce qui concerne leur mémoire et l'exécution de leur firmware. Cette solution permet ainsi de maintenir la confiance entre les appareils au sein d'un réseau d'objets connectés.

5.1 Récapitulatif des contributions

Dans cette thèse, l'accent est mis sur l'attestation à distance au travers de plusieurs contributions.

— Un état de l'art présente tout d'abord l'attestation à distance ainsi que les principales sous-catégories qui la compose : attestation matérielle, attestation logicielle et attestation hybride. L'attestation matérielle repose sur l'ajout de composants matériels spécialisés, comme les TPM ou les TEE, et leur utilisation pour réaliser une attestation localement. À l'inverse, l'attestation logicielle n'utilise aucun matériel spécifique et s'adapte donc à un plus grand nombre d'objets, en contrepartie d'une sécurité moindre. L'attestation hybride quant à elle utilise des composants plus classiques comme la ROM, afin de réaliser des attestation locales sécurisées et de les transmettre au sein du réseau.

  Dans cet état de l'art, l'influence des différents travaux de ces catégories est également étudiée de façon innovante au travers d'une chronologie et d'un tableau illustrant leurs liens de citation. Cette état de l'art fait l'objet d'un article qui est en cours de finalisation, et sera proposé à la publication prochainement.

— Cette thèse propose ensuite, à notre connaissance, la première proposition de framework d'attestation continue à distance. Cette proposition se base sur la définition préalable de ce qu'est un framework d'attestation à distance et des exigences fonctionnelles et de sécurité qui le composent, ainsi que sur une définition générale d'un réseau d'objets connectés.

  Sur la base de ces définitions, le framework d'attestation continue à distance CRAFT est détaillé. Il est composé de deux phases, *Hors-ligne* et *En-ligne*, et utilise un ensemble minimal de messages qui assurent le processus d'attestation continue. Ces messages permettent d'intégrer les protocoles d'attestation existants, et l'aspect continu de l'attestation repose sur les messages de type *heartbeat*. Cette contribution a fait l'objet d'une publication dans le journal IEEE Access en 2021.

  Des fonctionnalités additionnelles, dites ASMP (Adaptive Simultaneous Multi-Protocols), sont également introduites. Elles permettent à un appareil de changer le protocole d'attestation utilisé au sein du framework en fonction du contexte, comme par exemple de sa mobilité ou de ses voisins. Ces fonctionnalités avancées font l'objet d'un article en cours de soumission.

  CRAFT fait également l'objet d'une analyse de sécurité basée sur un modèle de menaces adapté à l'attestation à distance. Cette analyse démontre la bonne résistance du framework à une large variété d'attaques.

— Enfin, cette thèse présente des simulations détaillées du framework CRAFT qui le comparent à d'autres protocoles d'attestations existants. Ces simulations sont réalisées à l'aide du simulateur de réseaux à événements discrets Omnet++. Afin que les résultats des simulations soient représentatifs, une méthodologie statistique basée sur la répétition des scénarios de simulations avec des variations aléatoires est utilisée, ce qui permet notamment de calculer un intervalle de confiance des valeurs mesurées.

  Les simulations en elles-mêmes se distinguent en deux sous-partie. CRAFT est tout d'abord comparé à deux protocoles d'attestation seuls, SEDA puis US-AID, dans des scénarios fixes et mobiles avec déplacements aléatoires, pour différents nombres d'appareils dans le réseau. CRAFT est ensuite comparé avec et sans les fonctionnalités ASMP à la combinaison des deux protocoles SEDA et US-AID. Ces simulations se situent dans un environnement représentant une smart city à l'aide d'une grille sur laquelle des appareils mobiles (représentant des véhicules) se déplacent et dans laquelle des antennes et des appareils fixes (représentant des éléments d'infrastructure) sont placés. CRAFT illustre dans ces simulations sa bonne flexibilité, sa capacité d'adaptation aux types de réseaux et au contexte, ainsi que ses bonnes performances au regard de celles de protocoles d'attestation existants. Ces simulations sont inclues dans les différents articles publiés ou en cours de publication précédemment évoqués.

5.2 Futures pistes de recherches

Le framework d'attestation continue à distance CRAFT proposé dans cette thèse permet d'améliorer la confiance dans la sécurité d'un réseau d'objets connectés. CRAFT constitue un framework de base robuste, qui pourrait bénéficier de multiples fonctionnalités et réalisations supplémentaires. Quatre pistes de recherche sont pour le moment à l'étude :

— L'implémentation de CRAFT sur du matériel physique, avec une première phase de démonstration sur du matériel disponible sur l'étagère et une seconde phase sur du matériel de l'entreprise Icohup.

— Le développement d'une bibliothèque permettant d'implémenter CRAFT simplement sur n'importe quel matériel et d'y intégrer d'autres fonctionnalités.

— L'ajout d'une fonctionnalité prenant en compte les cas d'usage dans lesquels les objets connectés ont de multiples propriétaires.

— L'amélioration des fonctionnalités ASMP en y intégrant une détection automatique du contexte et une adaptation du protocole utilisé, basée sur des algorithmes d'intelligence artificielle.

5.2.1 Implémentation de CRAFT sur du matériel réel

Cette implémentation pourra dans un premier temps être réalisée sur un réseau de petite taille afin de démontrer la faisabilité de l'implémentation. Ce réseau pourra se placer dans le contexte d'une pièce, et être constitué d'une à deux antennes et de trois à cinq appareils se connectant à ces antennes. Ces différents appareils peuvent être des Raspberry Pi [71] ou des Arduino [72] par exemple, car ils sont prêts à l'usage et permettraient de rapidement déployer CRAFT. Dans un second temps, une implémentation pourra être faite sur du matériel de l'entreprise Icohup et à l'échelle d'un bâtiment dans le cadre d'un déploiement réel. Le matériel utilisé pourra notamment être le système RiumLights, qui repose sur des antennes réparties sur plusieurs salles, auxquelles sont appairés des boîtiers connectés équipés de voyants lumineux indiquant la mise sous-tension et le processus d'irradiation d'appareils médicaux. Pour un client typique d'Icohup sur ce produit, cela consiste en 5 antennes accompagnées chacune de deux à trois boîtiers connectés. Pour réaliser cette implémentation, il faudra pour cela évaluer l'adéquation du microcontrôleur actuellement utilisé par Icohup, ou proposer de le remplacer dans les produits si besoin, ainsi qu'intégrer le framework en parallèle des tâches métier existantes.

5.2.2 Développement d'une bibliothèque CRAFT prête à l'usage

Le développement de cette bibliothèque aura pour objectif de permettre d'implémenter CRAFT sur n'importe quel appareil capable d'installer cette bibliothèque. La première version sera réalisée en langage C, fréquemment utilisé dans les microcontrôleurs qu'utilisent les objets connectés, mais des portages ou des encapsulations dans d'autres langages peuvent facilement s'envisager. L'intérêt principal d'une telle bibliothèque est d'apporter la sécurité et la confiance d'un framework d'attestation continue à distance d'une façon simple à utiliser. Ainsi, les concepteurs d'objets connectés pourront l'intégrer avec un impact minimal dans leur processus de développement nécessairement axé sur les usages métier. Cette bibliothèque pourra être utilisée en version beta dans le cadre de l'implémentation de CRAFT sur du matériel réel.

5.2.3 Gestion de propriétaires multiples pour un appareil

Dans certains cas d'usages, les objets connectés peuvent être partagés par différents propriétaires [73]. C'est le cas par exemple de flottes de drones de livraison, qui peuvent être sollicités par différentes parties. Dans ce cas, l'objet connecté change régulièrement de réseau global avec lequel il communique, et il est souhaitable que l'attestation à distance soit diffusée à la fois au fournisseur de l'appareil et à son utilisateur actuel,

qui n'ont pas nécessairement une confiance mutuelle. Le fonctionnement actuel de CRAFT ne répond aujourd'hui pas à cette problématique, et il serait intéressant d'y remédier au travers d'une extension.

5.2.4 Algorithmes d'intelligence artificielle pour l'amélioration des fonctionnalités ASMP

L'efficacité des fonctionnalités ASMP repose sur la sélection du protocole d'attestation à distance le plus approprié au contexte courant dans lequel se trouve un appareil. Cela repose donc sur une bonne identification des éléments de contexte en temps réel, ainsi que sur la détermination du protocole le plus approprié vis-à-vis de ce contexte. Pour ce faire, il est possible de mettre en place des algorithmes d'intelligence artificielle, par exemple en utilisant des réseaux de neurones [74] qui peuvent associer les informations connues d'un objet (données des capteurs, topologie du réseau, mobilité,...) et l'associer au protocole adéquat. Cela reposerait sur une phase d'entraînement de l'algorithme, en le faisant évoluer dans différents contextes avec différents protocoles afin d'établir la relation entre les deux, puis sur une phase d'exploitation dans laquelle l'algorithme, alors capable d'analyser le contexte, choisirait automatiquement le protocole le plus approprié. L'enjeu de cette recherche sera d'avoir une diversité de contextes et de protocoles suffisante pour que cette fonctionnalité soit pertinente.

A

Bibliographie

Sommaire

Références

[1] STATISTA, *Number of Internet of Things (IoT) connected devices worldwide from 2019 to 2021, with forecasts from 2022 to 2030*, https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/, 2022.

[2] STATISTA, *Internet of Things (IoT) and non-IoT active device connections worldwide from 2010 to 2025*, https://www.statista.com/statistics/1101442/iot-number-of-connected-devices-worldwide/, 2020.

[3] S. WOLFERT, L. GE, C. VERDOUW et M.-J. BOGAARDT, « Big data in smart farming–a review, » *Agricultural systems*, t. 153, p. 69-80, 2017.

[4] T. ZONTA, C. A. DA COSTA, R. da ROSA RIGHI, M. J. de LIMA, E. S. da TRINDADE et G. P. LI, « Predictive maintenance in the Industry 4.0 : A systematic literature review, » *Computers & Industrial Engineering*, t. 150, p. 106 889, 2020.

[5] *A Standard for the Transmission of IP Datagrams over Ethernet Networks*, RFC 894, avr. 1984. adresse : https://www.rfc-editor.org/info/rfc894.

[6] NFC FORUM, *NFC Forum Specifications*, https://nfc-forum.org/build/specifications, 2022.

[7] ISO/IEC JTC 1/SC 31, *ISO/IEC 18000-63 :2021*, https://www.iso.org/standard/78309.html, 2021.

[8] 802.15 WG - WIRELESS SPECIALTY NETWORKS (WSN) WORKING GROUP, *IEEE Standard for Low-Rate Wireless Networks*, https://standards.ieee.org/ieee/802.15.4/7029/, 2020.

[9] 802.15 WG - WIRELESS SPECIALTY NETWORKS (WSN) WORKING GROUP, *IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 15.1a : Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Wireless Personal Area Networks (WPAN)*, https://standards.ieee.org/ieee/802.15.1/3513/, 2005.

[10] 802.11 WG - WIRELESS LAN WORKING GROUP, *IEEE Standard for Information technology–Telecommunications and information exchange between systems*

Local and metropolitan area networks–Specific requirements - Part 11 : Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, https://standards.ieee.org/ieee/802.11/7028/, 2020.

[11] ETSI, *Long Term Evolution (LTE)*, https://www.etsi.org/technologies/mobile/4G, 2022.

[12] LORA ALLIANCE, *What is LoRaWAN® Specification*, https://lora-alliance.org/about-lorawan/, 2022.

[13] WIRED, *Chinese IoT firm recalls 4.3 million connected cameras after giant botnet attack*, https://www.wired.co.uk/article/internet-down-dyn-october-2016, 2016.

[14] ZDNET.FR, *Les objets connectés peuvent-ils infecter les hôpitaux ?* https://www.zdnet.fr/actualites/les-objets-connectes-peuvent-ils-infecter-les-hopitaux-39923471.htm, 2021.

[15] R. VAN KRANENBURG et A. BASSI, « IoT challenges, » *Communications in Mobile Computing*, t. 1, nº 1, p. 1-5, 2012.

[16] THE OWASP FOUNDATION, *OWASP Top Ten*, https://owasp.org/www-project-top-ten/, 2022.

[17] MICROCHIP, *ATECC608B*, https://www.microchip.com/en-us/product/atecc608b, 2022.

[18] Y. GAO, S. F. AL-SARAWI et D. ABBOTT, « Physical unclonable functions, » *Nature Electronics*, t. 3, nº 2, p. 81-91, 2020.

[19] J. POSTEL, *Internet Protocol*, https://www.ietf.org/rfc/rfc791.txt, 1981.

[20] J. POSTEL, *Transmission Control Protocol*, https://www.ietf.org/rfc/rfc793.txt, 1981.

[21] R. FIELDING, J. GETTYS, J. MOGUL et al., *Hypertext Transfer Protocol – HTTP/1.1*, https://www.ietf.org/rfc/rfc2616.txt, 1999.

[22] A. BANKS, E. BRIGGS, K. BORGENDALE et R. GUPTA, *MQTT Version 5.0*, https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html, 2019.

[23] N. ASOKAN, F. BRASSER, A. IBRAHIM et al., « SEDA : Scalable Embedded Device Attestation, » in *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security*, sér. CCS '15, Denver, Colorado, USA : ACM, 2015, p. 964-975.

[24] F. KOHNHÄUSER, N. BÜSCHER et S. KATZENBEISSER, « SALAD : Secure and Lightweight Attestation of Highly Dynamic and Disruptive Networks, » in *Proceedings of the 2018 on Asia Conference on Computer and Communications*

Security, sér. ASIACCS '18, Incheon, Republic of Korea : ACM, 2018, p. 329-342. adresse : http://doi.acm.org/10.1145/3196494.3196544.

[25] A. IBRAHIM, A.-R. SADEGHI et G. TSUDIK, « US-AID : Unattended Scalable Attestation of IoT Devices, » in *2018 IEEE 37th Symposium on Reliable Distributed Systems (SRDS)*, IEEE, 2018, p. 21-30.

[26] P. KOEBERL, S. SCHULZ, A.-R. SADEGHI et V. VARADHARAJAN, « TrustLite : A Security Architecture for Tiny Embedded Devices, » in *Proceedings of the Ninth European Conference on Computer Systems*, sér. EuroSys '14, Amsterdam, The Netherlands : ACM, 2014, 10 :1-10 :14.

[27] K. ELDEFRAWY, N. RATTANAVIPANON et G. TSUDIK, « HYDRA : hybrid design for remote attestation (using a formally verified microkernel), » in *Proceedings of the 10th ACM Conference on Security and Privacy in wireless and Mobile Networks*, ACM, 2017, p. 99-110.

[28] E. DUSHKU, M. M. RABBANI, M. CONTI, L. V. MANCINI et S. RANISE, « SARA : Secure asynchronous remote attestation for IoT systems, » *IEEE Transactions on Information Forensics and Security*, t. 15, p. 3123-3136, 2020.

[29] T. KOBAYASHI, T. SASAKI, A. JADA, D. E. ASONI et A. PERRIG, « SAFES : Sandboxed Architecture for Frequent Environment Self-measurement, » in *Proceedings of the 3rd Workshop on System Software for Trusted Execution*, sér. SysTEX '18, Toronto, Canada : ACM, 2018, p. 37-41. adresse : http://doi.acm.org/10.1145/3268935.3268939.

[30] TRUSTED COMPUTING GROUP, *Trusted Platform Module (TPM) Summary*, https://trustedcomputinggroup.org/resource/trusted-platform-module-tpm-summary/, 2008.

[31] GLOBAL PLATFORM, *The Trusted Execution Environment : Delivering Enhanced Security at a Lower Cost to the Mobile Market*, https://globalplatform.org/wp-content/uploads/2018/04/GlobalPlatform_TEE_Whitepaper_2015.pdf, 2015.

[32] K. E. DEFRAWY, A. FRANCILLON, D. PERITO et G. TSUDIK, « SMART : Secure and Minimal Architecture for (establishing a dynamic) Root of Trust, » in *Network and Distributed SystemSecurity Symposium (NDSS)*, Internet Society, 2012.

[33] J. NOORMAN, P. AGTEN, W. DANIELS et al., « Sancus : Low-cost trustworthy extensible networked devices with a zero-software trusted computing base, » in *Presented as part of the 22nd {USENIX} Security Symposium ({USENIX} Security 13)*, 2013, p. 479-498.

[34] Y. Li, J. M. McCune et A. Perrig, « VIPER : Verifying the integrity of peripherals' firmware, » in *Proceedings of the 18th ACM conference on Computer and communications security*, 2011, p. 3-16.

[35] C. Kil, E. C. Sezer, A. M. Azab, P. Ning et X. Zhang, « Remote attestation to dynamic system properties : Towards providing complete system integrity evidence, » in *2009 IEEE/IFIP International Conference on Dependable Systems & Networks*, IEEE, 2009, p. 115-124.

[36] A. Seshadri, A. Perrig, L. Van Doorn et P. Khosla, « SWATT : Software-based attestation for embedded devices, » in *IEEE Symposium on Security and Privacy, 2004. Proceedings. 2004*, IEEE, 2004, p. 272-282.

[37] R. V. Steiner et E. Lupu, « Towards more practical software-based attestation, » *Computer Networks*, t. 149, p. 43-55, 2019. adresse : http://www.sciencedirect.com/science/article/pii/S1389128618307631.

[38] M. Conti, E. Dushku et L. V. Mancini, « RADIS : Remote Attestation of Distributed IoT Services, » *arXiv preprint arXiv :1807.10234*, 2018.

[39] F. Toffalini, A. Biondo, E. Losiouk, J. Zhou et M. Conti, « SCARR : A Novel Scalable Runtime Remote Attestation, » *arXiv preprint arXiv :1807.08003*, 2018.

[40] X. Yang, X. He, W. Yu et al., « Towards a low-cost remote memory attestation for the smart grid, » *Sensors*, t. 15, nº 8, p. 20 799-20 824, 2015.

[41] M. Conti, P. Kaliyar, M. M. Rabbani et S. Ranise, « SPLIT : A Secure and Scalable RPL routing protocol for Internet of Things, » in *2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, IEEE, 2018.

[42] X. Carpent, N. Rattanavipanon et G. Tsudik, « Remote attestation of iot devices via smarm : Shuffled measurements against roving malware, » in *2018 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, IEEE, 2018, p. 9-16.

[43] G. Dessouky, T. Abera, A. Ibrahim et A.-R. Sadeghi, « Litehax : Lightweight hardware-assisted attestation of program execution, » in *2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, IEEE, 2018.

[44] M. Ambrosin, M. Conti, A. Ibrahim, G. Neven, A.-R. Sadeghi et M. Schunter, « SANA : Secure and Scalable Aggregate Network Attestation, » in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, sér. CCS '16, Vienna, Austria : ACM, 2016, p. 731-742. adresse : http://doi.acm.org/10.1145/2976749.2978335.

[45] M. AMBROSIN, M. CONTI, R. LAZZERETTI, M. M. RABBANI et S. RANISE, « PADS : Practical Attestation for Highly Dynamic Swarm Topologies, » *CoRR*, t. abs/1806.05766, 2018. arXiv : 1806.05766. adresse : http://arxiv.org/abs/1806.05766.

[46] K. ELDEFRAWY, I. O. NUNES, N. RATTANAVIPANON, M. STEINER et G. TSUDIK, « Formally verified hardware/software co-design for remote attestation, » *Usenix Security 2019*, 2019.

[47] T. ABERA, N. ASOKAN, L. DAVI et al., « C-FLAT : Control-FLow Attestation for Embedded Systems Software, » in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ACM, 2016, p. 743-754.

[48] G. DESSOUKY, S. ZEITOUNI, T. NYMAN et al., « LO-FAT : Low-Overhead Control Flow ATtestation in Hardware, » in *Proceedings of the 54th Annual Design Automation Conference 2017*, ACM, 2017, p. 24.

[49] X. CARPENT, K. ELDEFRAWY, N. RATTANAVIPANON et G. TSUDIK, « Lightweight swarm attestation : a tale of two lisa-s, » in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, ACM, 2017, p. 86-100.

[50] G. BANSAL et B. SIKDAR, « S-MAPS : Scalable mutual authentication protocol for dynamic UAV swarms, » *IEEE Transactions on Vehicular Technology*, t. 70, n° 11, p. 12 088-12 100, 2021.

[51] A. IBRAHIM, A.-R. SADEGHI, G. TSUDIK et S. ZEITOUNI, « DARPA : Device Attestation Resilient to Physical Attacks, » in *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, sér. WiSec '16, Darmstadt, Germany : ACM, 2016, p. 171-182.

[52] L. PETZI, A. E. B. YAHYA, A. DMITRIENKO, G. TSUDIK, T. PRANTL et S. KOUNEV, « SCRAPS : Scalable Collective Remote Attestation for Pub-Sub IoT Networks with Untrusted Proxy Verifier, » in *31st USENIX Security Symposium (USENIX Security 22)*, 2022, p. 3485-3501.

[53] F. BRASSER, B. EL MAHJOUB, A.-R. SADEGHI, C. WACHSMANN et P. KOEBERL, « TyTAN : Tiny trust anchor for tiny devices, » in *Proceedings of the 52nd annual design automation conference*, 2015, p. 1-6.

[54] J. GÖTZFRIED, T. MÜLLER, R. DE CLERCQ, P. MAENE, F. FREILING et I. VERBAUWHEDE, « Soteria : Offline software protection within low-cost embedded devices, » in *Proceedings of the 31st Annual Computer Security Applications Conference*, ACM, 2015, p. 241-250.

[55] S. ZEITOUNI, G. DESSOUKY, O. ARIAS et al., « Atrium : Runtime attestation resilient under memory attacks, » in *Proceedings of the 36th International Conference on Computer-Aided Design*, IEEE Press, 2017, p. 384-391.

[56] J. NOORMAN, J. V. BULCK, J. T. MÜHLBERG et al., « Sancus 2.0 : A low-cost security architecture for IoT devices, » *ACM Transactions on Privacy and Security (TOPS)*, t. 20, nº 3, p. 7, 2017.

[57] W. FENG, Y. QIN, S. ZHAO et D. FENG, « AAoT : Lightweight attestation and authentication of low-resource things in IoT and CPS, » *Computer Networks*, t. 134, p. 167-182, 2018.

[58] N. ASOKAN, T. NYMAN, N. RATTANAVIPANON, A.-R. SADEGHI et G. TSUDIK, « ASSURED : Architecture for secure software update of realistic embedded devices, » *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, t. 37, nº 11, p. 2290-2300, 2018.

[59] M. AMMAR, B. CRISPO et G. TSUDIK, « SIMPLE : A Remote Attestation Approach for Resource-constrained IoT devices, » in *2020 ACM/IEEE 11th International Conference on Cyber-Physical Systems (ICCPS)*, IEEE, 2020, p. 247-258.

[60] P.-H. YANG et S.-M. YEN, « SARA : Sandwiched attestation through remote agents for cluster-based wireless sensor networks, » *International Journal of Distributed Sensor Networks*, 2017. eprint : https://doi.org/10.1177/1550147717719192. adresse : https://doi.org/10.1177/1550147717719192.

[61] A. IBRAHIM, A.-R. SADEGHI et S. ZEITOUNI, « SeED : Secure Non-interactive Attestation for Embedded Devices, » in *Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, sér. WiSec '17, Boston, Massachusetts : ACM, 2017, p. 64-74. adresse : http://doi.acm.org/10.1145/3098243.3098260.

[62] F. KOHNHÄUSER, N. BÜSCHER, S. GABMEYER et S. KATZENBEISSER, « SCAPI : A Scalable Attestation Protocol to Detect Software and Physical Attacks, » in *Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, sér. WiSec '17, Boston, Massachusetts : ACM, 2017, p. 75-86. adresse : http://doi.acm.org/10.1145/3098243.3098255.

[63] X. CARPENT, G. TSUDIK et N. RATTANAVIPANON, « ERASMUS : Efficient remote attestation via self-measurement for unattended settings, » in *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, IEEE, 2018, p. 1191-1194.

[64] H. TAN, G. TSUDIK et S. JHA, « MTRA : Multi-Tier randomized remote attestation in IoT networks, » *Computers & Security*, t. 81, p. 78-93, 2018.

[65] S. C. HELBLE, I. D. KRETZ, P. A. LOSCOCCO, J. D. RAMSDELL, P. D. ROWE et P. ALEXANDER, « Flexible mechanisms for remote attestation, » *ACM Transactions on Privacy and Security (TOPS)*, t. 24, nº 4, p. 1-23, 2021.

[66] J. D. RAMSDELL, P. D. ROWE, P. ALEXANDER et al., « Orchestrating layered attestations, » in *International Conference on Principles of Security and Trust*, Springer, Cham, 2019, p. 197-221.

[67] R. M. HALLDÓRSSON, E. DUSHKU et N. DRAGONI, « ARCADIS : Asynchronous Remote Control-Flow Attestation of Distributed IoT Services, » *IEEE Access*, t. 9, p. 144 880-144 894, 2021.

[68] D. DOLEV et A. YAO, « On the security of public key protocols, » *IEEE Transactions on Information Theory*, t. 29, nº 2, p. 198-208, mars 1983.

[69] INET COMMUNITY, *INET Framework*, https://inet.omnetpp.org/, 2022.

[70] wolfSSL INC., *wolfCrypt Embedded Crypto Engine*, https://www.wolfssl.com/docs/benchmarks/, 2020.

[71] RASPBERRY PI FOUNDATION, *Raspberry Pi*, https://www.raspberrypi.org/, 2022.

[72] THE ARDUINO TEAM, *Arduino*, https://www.arduino.cc/, 2022.

[73] M.-O. PAHL, « Multi-Tenant IoT Service Management towards an IoT App Economy, » in *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, IEEE, 2019, p. 1-4.

[74] W. ZHANG, Y. ZHANG, L. XU et al., « Modeling IoT equipment with graph neural networks, » *IEEE Access*, t. 7, p. 32 754-32 764, 2019.

Liste des Travaux

Journaux internationaux à comité de lecture

- L. MOREAU, E. CONCHON et D. SAUVERON, « CRAFT : A Continuous Remote Attestation Framework for IoT, » *IEEE Access*, t. 9, p. 46 430-46 447, 2021. adresse : https://ieeexplore.ieee.org/document/9382291.

En cours de soumission

- L. MOREAU, E. CONCHON et D. SAUVERON, *An Adaptive Simultaneous Multi-Protocol extension of CRAFT*.
- L. MOREAU, E. CONCHON et D. SAUVERON, *A comprehensive survey of remote attestation and its evolution*.

Solutions de confiance pour la sécurité dans l'Internet des Objets

Résumé : Dans cette thèse, nous proposons un framework original d'attestation continue à distance qui permet d'améliorer la confiance et la sécurité au sein de réseaux d'objets connectés. Ce framework permet à des objets connectés d'utiliser n'importe quelle solution d'attestation à distance existante et d'adapter le protocole à utiliser en fonction de l'environnement dans lequel ils se trouvent. Cette thèse présente tout d'abord un état de l'art détaillé de ce qu'est l'attestation à distance, que ce soit les approches centrées sur les composants matériels, celles reposant sur des solutions uniquement logicielles, et enfin les solutions dites hybrides qui incluent des aspects à la fois matériels et logiciels. Cette thèse détaille ensuite CRAFT, le premier framework d'attestation continue à distance. CRAFT repose sur un ensemble de messages permettant de répondre à différentes exigences fonctionnelles et de sécurité. Il s'appuie sur une définition générique d'un réseau d'objets connectés, et fait également l'objet d'une analyse de sécurité pour montrer sa résilience aux attaques sur les réseaux d'objets connectés. Enfin, nous évaluons notre proposition à l'aide du simulateur d'événements discrets Omnet++. Ces simulations s'appuient sur une méthodologie statistique robuste et mettent en évidence les bons résultats du framework.

Mots clés : Attestation continue à distance, sécurité IoT, framework de confiance

Trust solutions for security in the Internet of Things

Abstract : In this thesis, we propose an original continuous remote attestation framework to improve trust and security in networks of connected objects. This framework allows connected objects to use any existing remote attestation protocol and adapt the one to be used according to the environment in which they are located. This thesis first presents a detailed state of the art of remote attestation, whether it is the approaches centered on hardware components, those based on software-only solutions, and finally hybrid solutions that include both hardware and software aspects. This thesis then details CRAFT, the first continuous remote attestation framework. CRAFT is based on a set of messages allowing to meet different functional and security requirements. It is based on a generic definition of a network and is also subject to a security analysis to show its resilience to attacks on connected object networks. Finally, we evaluate our proposal using the discrete event simulator Omnet++. These simulations are based on a robust statistical methodology and highlight the good results of the framework.

Keywords : Continuous remote attestation, IoT security, trust framework

Univ. de Limoges, CNRS, XLIM, UMR 7252, F-87000 Limoges, France