De la vulnérabilité des nœuds capteurs à la certification des transactions sur le réseau, une approche de la sécurisation de l'Internet des Objets

Loïc Dalmasso

THESE POUR OBTENIR LE GRADE DE DOCTEUR DE L'UNIVERSITE DE MONTPELLIER

En SyAM - Systèmes Automatiques et Microélectroniques

École doctorale I2S - Information, Structures et Systèmes

Unité de recherche LIRMM

De la vulnérabilité des nœuds capteurs à la certification des transactions sur le réseau, une approche de la sécurisation de l'Internet des Objets

Présentée par Loïc Dalmasso

Le 15/12/2020

Sous la direction de Pascal Benoit

Devant le jury composé de

Pascal Benoit	Maître de Conférences, Université de Montpellier	Directeur de thèse
Florent Bruguier	Maître de Conférences, Université de Montpellier	Examinateur
Loïc Lagadec	Professeur, ENSTA-Bretagne	Rapporteur
Lionel Torres	Professeur, Université de Montpellier	Examinateur, Président
Assia Tria	Directrice Scientifique, CEA-Leti	Examinatrice
François Verdier	Professeur, Université de Nice	Rapporteur

UNIVERSITÉ DE MONTPELLIER

De la vulnérabilité des nœuds capteurs à la certification des transactions sur le réseau, une approche de la sécurisation de l'Internet des Objets

Table des matières

Liste des figures

Liste des tableaux

1. Introduction

1.1. Une inéluctable évolution

Depuis son origine, l'Homme n'a cessé de faire preuve d'inventivité et d'ingéniosité : des premiers outils d'os et de silex à la préhistoire, en passant par l'imprimerie, la révolution industrielle du XIX^{e} siècle, et jusqu'à plus récemment, celle du numérique. Cette dernière est principalement portée à la fois par la conception du tout premier microprocesseur (*Intel 4004*), et par la mise en réseau d'une vingtaine d'ordinateurs éloignés géographiquement, prélude d'Internet. De là, une multitude de systèmes ont vu le jour, avec évidemment les premiers ordinateurs personnels, mais aussi des baladeurs numériques, des consoles de jeux et bien d'autres. Ce phénomène est d'autant plus amplifié par la miniaturisation des composants, dont les célèbres lois de Moore en illustrent la dimension. Avec une prédiction qui s'est avérée exacte, stipulant que le nombre de transistors sur une puce silicium se verrait multiplier par deux tous les deux ans, les systèmes sont devenus de plus en plus petits et moins couteux, tout en étant plus rapides et puissants. À tel point que le champ d'action des chercheurs est passé de l'électronique, à la microélectronique, et aujourd'hui à la nanoélectronique en l'espace de quelques décennies.

Cette miniaturisation exponentielle n'a fait qu'accroitre les possibilités d'automatisation des machines, pouvant réaliser des tâches de plus en plus complexes. Mais plus encore, elle a participé à la démocratisation de la technologie, et à son expansion à tous les secteurs d'activités tels que la santé, l'industrie, l'automobile, l'énergie, etc. C'est le début de l'ère informatique ubiquitaire, définie selon Mark Weiser [1], qui place l'ordinateur (ou les machines intelligentes au sens large) comme le prolongement de l'inconscience de l'Homme, prévu pour l'assister. Sans l'avoir explicitement stipulé, le concept de ce qui deviendra l'Internet des Objets était né.

1.2. L'Internet des Objets (IdO)

1.2.1. Définition

Tel que défini par l'Union Internationale des Télécommunications (UIT), l'Internet des Objets (IdO) est « une infrastructure mondiale pour la société de l'information, qui permet de disposer de services évolués en interconnectant des objets (physiques et virtuels) grâce aux technologies de l'information et de la communication interopérables existantes ou en évolution ». [2]. En d'autres termes, l'IdO exploite les capacités de captation, de traitement de l'information, d'action et de communication d'entités autonomes, dans le but de créer de nouveaux services et applications. Par exemple, l'interconnexion de plusieurs capteurs de température, d'humidité et de pression répartit

sur une zone géographique (*e.g.* région, pays ...) permet de créer un service météo. De la même manière, la mise en commun de données telles que la position GPS et la vitesse, améliore grandement l'aide à la navigation, en proposant en temps réel le trajet le plus adapté. Grâce à sa faculté d'interagir à la fois son environnement et ses homologues, mais également avec l'Homme et le réseau mondial, le modèle de l'IdO prend toute son ampleur. C'est la numérisation du monde de bout en bout. Son champ d'application touche tous les secteurs d'activités, comme l'agriculture [3] (*e.g.* suivi météorologique, optimisation de ressources, détection de parasites), l'industrie [4] (*e.g.* traçabilité des produits), les villes [5] (*e.g.* gestion du trafic, de l'énergie et des déchets). Plus récemment encore, le secteur de la santé a suscité un énorme intérêt, avec la mise en place de l'application « StopCovid » [6], déployée sur les smartphones de tout un chacun, dans le but de tracer et endiguer la propagation de la pandémie COVID-19.

Comme son nom le souligne, l'élément phare de l'IdO concerne les objets. Ce sont des systèmes physiques intégrant un ou plusieurs capteurs et/ou actionneurs, avec des capacités de traitement de données et un moyen de communication. Cela peut être un simple tag RFID (*Radio Frequency Identification*), un capteur/actionneur ou encore des systèmes plus complexes tels que les smartphones. Plus spécifiquement, ils sont souvent associés au terme « nœuds capteurs », comme indiqué par l'Union International des Télécommunications (UIT) [2], lorsqu'il est question de réseaux sans fil (*Wireless Sensor Networks*, WSN). N'échappant pas à la règle, les objets sont soumis à différentes contraintes, comme par exemple la puissance de calcul ou la mémoire, suivant le contexte d'application. Mais il y en a une qui impose sa forte influence sur les autres, la taille. En effet, l'objectif de l'IdO est de répandre une multitude de systèmes dans l'environnement, ce qui s'avère être plus simple d'un point de vue mise en œuvre, lorsque ces derniers sont petits. Il en est de même pour l'aspect budgétaire, permettant leur production en grande quantité. La réduction de la taille d'un objet conduit inévitablement à limiter les ressources qu'il peut physiquement embarquer tels que la mémoire, le microprocesseur/microcontrôleur. De plus, en raison de leur immersion dans l'environnement, les objets connectés bénéficient rarement d'un accès direct au réseau électrique. Par conséquent, ils doivent embarquer leur propre source d'énergie, elle aussi contrainte par la taille. Cela peut être une batterie, voire une pile pour les plus petits d'entre eux. Bien que certains puissent être équipés de récupérateurs d'énergie telles que des cellules photovoltaïques, l'énergie constitue une ressource rare et cruciale pour les objets connectés. Malgré toutes leurs contraintes, il est demandé aux objets d'embarquer de plus en plus de fonctionnalités, les rendant plus intelligents et aptes à traiter les informations. Il va sans dire que cela impacte fortement le besoin en énergie.

Une autre propriété phare d'un objet connecté est sa faculté à transmettre et recevoir des informations. Là encore, les besoins en débit, en portée, et en énergie, vont pousser la recherche à développer de nouveaux protocoles sans fil, plus propices au contexte de l'IdO. En effet, les réseaux

cellulaires (*e.g.* 3G, 4G) bénéficient d'un fort débit avec une grande portée (plusieurs dizaines de kilomètres), mais sont trop énergivores pour être employés sur des objets, destinés à être déployés sur de longue période dans leur environnement et sans accès au réseau électrique. De plus, la plupart des cas d'usages de l'IdO ne nécessitent, à ce jour, qu'une faible quantité de données à transmettre, ce qui ouvre l'opportunité de concevoir des réseaux bas débits, et peu énergivores. Avec une portée jusqu'à 100m, le Wi-Fi et le Bluetooth Low Energy (BLE) font eux aussi partie des protocoles réseaux bien connus du grand public. Alors que le premier induit une consommation énergétique non négligeable, le BLE se montre plus clément, au détriment d'un plus faible débit. Néanmoins, leur principale contrainte réside dans la nécessité d'une passerelle (*gateway*), entre l'objet et Internet. Le protocole ZigBee fait lui aussi partie de ces réseaux faible consommation. Avec une portée de seulement 10m, il est spécialement conçu pour des applications domotiques. Peu énergivore et n'atteignant qu'un faible débit, il possède la particularité de fonctionner en réseau maillé : chaque entité transmet, reçoit et relaie les informations à ses homologues. Enfin, et certainement ceux pouvant favoriser le déploiement massif de l'IdO, les réseaux basse consommation à longue portée (plusieurs kilomètres), appelés *Low-Power Wide-Area Networks* (LPWAN). Spécialement développés pour les objets connectés, ils exploitent les basses fréquences (800-900 MHz) offrant une faible consommation énergétique, toujours avec un bas débit. Parmi eux, SigFox, technologie propriétaire, et son opposé *open source* LoRa, développé par le consortium LoRa Alliance. Les deux protocoles permettent aux objets de communiquer directement entre eux (pair à pair). Toutefois, ces derniers ne peuvent directement accéder au réseau global Internet, en raison des trop faibles ressources dont ils disposent. Ayant entrevu le potentiel qu'offrirait cette interconnexion, les concepteurs des LPWAN intègrent dans leur protocole, la possibilité aux objets de communiquer avec des passerelles (*gateways*). Elles jouent le rôle d'intermédiaire entre les réseaux sans fil et Internet, décuplant ainsi les opportunités. La Figure 1-1 illustre cette architecture, devenue la définition même d'un réseau IdO. Enfin, l'année 2020 entame un premier pas vers le déploiement de la 5G, 5ème génération des standards de la téléphonie mobile. Alliant à la fois haut débit, faible latence, forte scalabilité et efficacité énergétique [7], la 5G promet d'accélérer l'expansion de l'IdO. D'une part, elle se positionne comme une meilleure alternative aux solutions actuelles comme le LoRa ou le SigFox offrant par exemple une plus grande longévité des batteries. Mais la 5G ouvre également de nouveaux usages, avec l'IdO haut débit (*e.g.* surveillance par image/vidéo) et les applications critiques utilisant des équipements médicaux ou encore les voitures connectées, où les échanges de données doivent être garantis dans un temps imparti (*i.e.* temps réel).

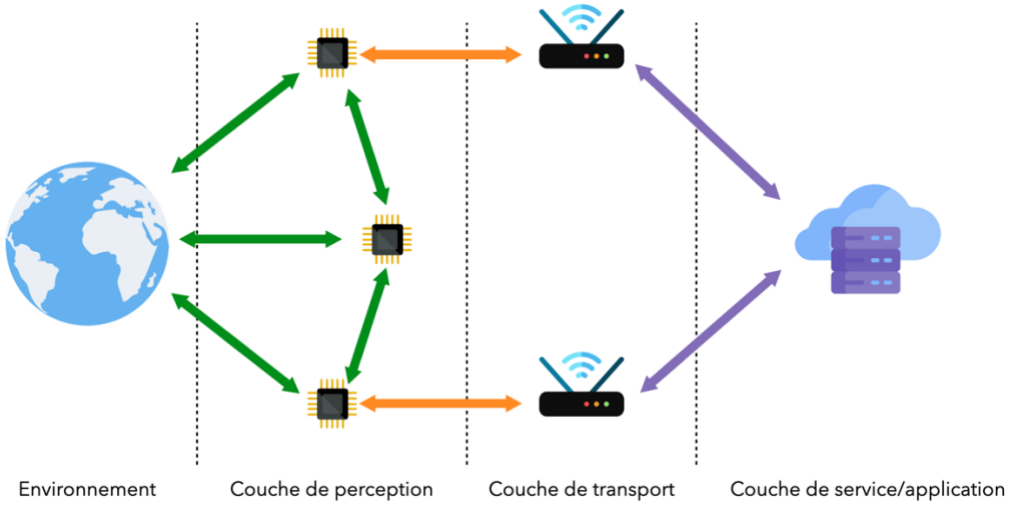


Figure 1-1 - Architecture classique d'un réseau IdO

L'architecture peut être découpée en trois couches :

Couche de perception : composée des objets, ces derniers numérisent et agissent sur leur environnement. Ils peuvent également collecter et s'échanger des informations soit directement entre eux soit par l'intermédiaire des passerelles.

Couche de transport : composée des passerelles dont leur rôle est de convertir le signal radio des entités de la couche de perception (*e.g.* Wi-Fi, BLE, LoRa), vers le protocole TCP/IP, utilisé sur le réseau Internet.

Couche de service/application : regroupant la majorité des ressources du réseau (*i.e.* puissance de calcul, espace de stockage), cette couche est composée de serveurs, en chargent de stocker les données des objets, provenant des passerelles, mais aussi d'effectuer les traitements et la prise de décision.

1.2.2. Un écosystème hétérogène, dynamique et en expansion

De par la multitude de plateformes, de réseaux, et d'applications, l'Internet des Objets fait preuve d'une incroyable hétérogénéité, dont l'UIT la désigne comme l'une des caractéristiques fondamentales de l'IdO [8]. Principalement pour des raisons d'économie d'énergie, les objets sont en perpétuel changement d'état (*i.e.* veille/réveil, connecté/déconnecté), rendant l'écosystème hautement dynamique [8]. Les capacités d'interconnectivité des entités leur permettent de s'échanger des données et de coopérer, créant ainsi de nouveaux services. Face à ce fort potentiel, et ce, dans quasiment tous les secteurs d'activité, la quantité d'objets connectés n'a cessé de croitre cette dernière décennie. Une étude de *Strategy Analyics* estime leur nombre à 38,6 milliards en 2025 [9].

L'entreprise CISCO prévoit même une extension de ce concept avec « l'Internet de tout » (*Internet of Everything*) [10], faisant référence à un réseau de réseaux d'IdO, regroupant des milliards d'appareils connectés.

Malgré cet engouement, de nombreuses contraintes technologiques restent à résoudre. Parmi elles, l'interopérabilité entre les différents objets nécessite l'adhésion à un standard. La quantité considérable de données produites implique des moyens de stockage adaptés ainsi qu'une capacité à les transmettre et les traiter (*i.e.* scalabilité). Étant donnée leur nature, les objets sont déployés dans un environnement où bien souvent leur source d'énergie est limitée (*e.g.* batterie, pile), avec parfois un récupérateur d'énergie (*e.g.* panneaux solaires). Au-delà des aspects technologiques, la qualité de service, par exemple le temps de réponse et la fiabilité des informations doivent être assurés. Enfin, un des axes majeurs concerne la sécurité, que ce soit au niveau des données, du réseau, ou plus généralement, d'un point de vue systémique.

1.2.3. Les enjeux sécuritaires

Après la ruée vers l'or, et celle du pétrole, la donnée numérique devient la nouvelle ressource précieuse, qu'il convient d'apprivoiser, de contrôler et surtout de sécuriser. En effet, de par la numérisation intensive de l'environnement, associée à l'hyperconnectivité de tous les appareils, les données constituent le carburant des applications numériques. Comme indiqué par l'UIT [8], la sécurité doit assurer la confidentialité, l'intégrité et l'authenticité des informations circulant sur le réseau, mais également des services qui en découlent. L'autre axe majeur concerne la vie privée. Effectivement, une grande partie des objets connectés appartient à des propriétaires, traite des données provenant d'utilisateurs. Ces dernières peuvent contenir des informations sensibles qu'il convient de protéger (*e.g.* coordonnées bancaires, données médicales, etc.). Alors que ces éléments semblent évidents, une première difficulté réside dans l'enchevêtrement des champs d'action de la sécurité et de la vie privée. La protection de cette dernière ne doit pas empêcher l'authentification de la première. En lien avec l'actualité, l'application « StopCovid » [6] est un cas d'école : elle doit permettre la traçabilité des personnes testées positives au COVID-19, afin d'alerter toutes celles ayant été en contact récemment. Dans un scénario aussi sensible qu'est le secteur de la santé, plusieurs problématiques entrent en compte [11]. La première concerne l'identification. Doit-on utiliser les véritables identités des personnes ou au contraire des pseudonymes ? Dans ce dernier cas, est-il possible d'assurer l'impossibilité de remonter jusqu'à la véritable identité ? Ces éléments mènent directement au second point clé, la gestion des malades. Faut-il utiliser une base de données centralisée dont l'entité en charge pourrait abuser de son pouvoir ? Ou au contraire, adopter une approche dite décentralisée où chaque téléphone transmet à tout le monde la liste des (pseudo-) contacts récents ? Là encore, il convient de déterminer la méthode de transmission et d'envoi d'alerte.

Faut-il utiliser un réseau pair à pair, ou passer par un intermédiaire ? Alors que la première approche ne peut être efficace que si l'application est utilisée par un nombre suffisant de personnes (ce qui implique une adhésion de la population), la seconde possède un caractère centralisé, où là encore, l'intermédiaire pourrait intercepter et récupérer des informations confidentielles.

L'exemple ci-dessus n'est pas un cas isolé et s'adresse plus généralement à tout le contexte de l'IdO. Ce-dernier, porté par l'hyperconnectivité des appareils dans tous les domaines, se doit d'intégrer des mesures de sécurité et de protection de vie privée, de l'origine des données, à leur transmission, en passant par leur agrégation, leur stockage ainsi que de leur traitement [8]. Mais au-delà de la fiabilité et de la sécurité des données, l'IdO constitue une cible de choix pour les cyberattaques. En effet, la plus grande attaque par déni de service distribué (DDoS), a été réalisée à l'aide du logiciel malveillant « Mirai » [12], dédié aux objets connectés. Telle une pandémie numérique, l'objectif est dans un premier temps de répandre « Mirai » sur une multitude de systèmes. Grâce à leur connectivité, ces derniers tentent d'infecter leurs homologues, en y installant le logiciel malveillant. Cette opération est rendue possible par la négligence des constructeurs, laissant les accès et les mots de passe par défaut (*e.g.* 123456). La seconde étape consiste en l'attaque DoS en elle-même. L'attaquant transmet alors une commande aux objets malveillants, les forçant à inonder le réseau de requêtes, faisant saturer les services ciblés.

Avec l'expansion de l'IdO les objets connectés envahissent notre environnement et avec eux, le risque de cyberattaques se trouve démultiplié. De par leur omniprésence dans tous les domaines d'activités et les informations qu'ils échangent, les objets sont des cibles de choix pour les cybercriminels. De ce fait, la sécurité à tous les niveaux, c'est-à-dire de la confidentialité, de l'intégrité et de l'authenticité, jusqu'au respect de la vie privée, en passant par la résilience aux attaques, constitue un axe majeur dans la viabilité du déploiement de l'IdO. Toutefois, les ressources disponibles dans les objets, en termes de puissance de calcul et de mémoire sont souvent limitées au strict minimum requis par l'application, afin de réduire la dépense énergétique et par conséquent, améliorer l'autonomie. Cette restriction des ressources rend inadapté, voire impossible, l'emploi d'outils traditionnels comme la cryptographie classique pour assurer la sécurité. C'est pourquoi il est primordial d'étudier de nouvelles alternatives alliant robustesse, efficacité et faibles utilisations de ressources. En effet, garantir la sécurité sans compromettre les performances des objets connectés est un défi de taille, en raison de leur capacité de calcul, de mémoire et d'énergie limitées. Afin de surmonter ces contraintes, les systèmes de Cloud ont intensivement été déployés durant ces dernières années. Toutefois, l'expansion de l'IdO est telle, que les architectures centralisées ne sont pas adaptées aux besoins de réactivité et de consommation énergétique des objets. De plus, d'un point de vue sécuritaire, les systèmes de Cloud représentent un point de défaillance unique (*single point of failure*). En réponse à cela, l'émergence de ce qui est communément appelé le *Fog Computing*, apporte de

nombreux avantages en termes d'efficacité et de scalabilité. Dans ce nouvel écosystème, chaque entité coopère grâce à un protocole de communication décentralisé *Machine-to-Machine* (M2M), sans s'appuyer sur un organisme central. Cependant, la suppression d'une entité unique de confiance exige de nouveaux mécanismes de sécurité pour assurer la confiance sur le réseau, à tous les niveaux, depuis les objets jusqu'au Cloud, en passant par les passerelles (*gateways*).

1.3. Objectif de la thèse

Les contributions de cette thèse se découpent selon deux axes principaux. Afin d'apporter une clarté visuelle, la Figure 1-2 met en image le positionnement de ces derniers dans l'écosystème IdO.

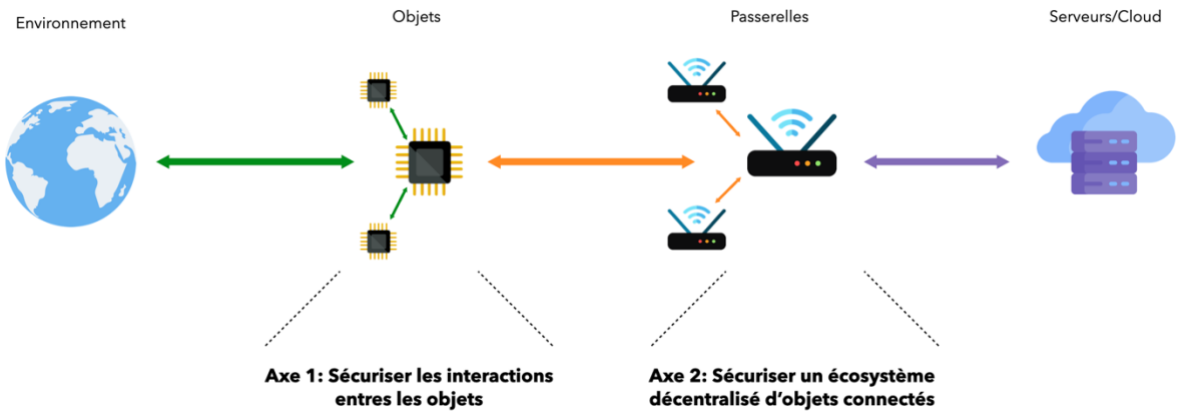


Figure 1-2 - Axes d'étude abordés dans cette thèse

Le premier axe se focalise sur la sécurisation des interactions entre les objets, et plus particulièrement sur la confidentialité des données transmises sur le réseau. L'objectif de cette étude est de comparer les gains de la cryptographie légère par rapport à la classique, dans leurs implémentations matérielles. Plus précisément, les travaux se sont concentrés sur l'évaluation du standard AES-128 [13], et de ses dérivés proches PRESENT-80/128 [14] et GIFT-64-128/GIFT-128-128 [15]. Bien que plusieurs contributions scientifiques soient disponibles dans l'état de l'art, l'intérêt de l'étude réalisée repose sur l'utilisation de métriques communes, permettant une meilleure comparaison des algorithmes en termes de taille d'implémentations, de performances, d'énergie, mais aussi de résistance aux attaques par canaux auxiliaires, reconnues comme un vecteur de vulnérabilité majeur. De plus, les travaux proposés sont parmi les premiers à avoir étudié et comparé l'algorithme GIFT à ses homologues.

Le second axe adopte une approche systémique. Assurer la confidentialité des interactions entre les objets est certes primordial, mais ne suffit pas à garantir la sécurité de tout l'écosystème. Par exemple, elle n'est d'aucune utilité pour contrecarrer une attaque DDoS initiée par « Mirai » [12].

De plus, avec une architecture réseau devenant propice à la décentralisation, c'est-à-dire sans organisme tiers, l'IdO doit se doter de nouveaux mécanismes, lui permettant d'être entièrement autonome et d'assurer sa propre sécurité. Par ailleurs, l'idée d'incitation au partage de données, de ressources et des services qui en découlent doit faire l'objet d'une attention toute particulière. En effet, étant le fondement même de l'IdO, favoriser les échanges, au sens qualitatif, constitue un point majeur dans son développement et son déploiement. En d'autres termes, l'objectif de ce second axe est d'instaurer la confiance dans un écosystème IdO décentralisé. Pour cela, les travaux de cette partie se sont orientés sur le concept de la blockchain, qui a intensément fait parler de lui ces dernières années, et qui se montre intéressant pour répondre à la problématique de cet axe. Une analyse approfondie a permis d'extraire les points critiques afin de concevoir un protocole de confiance spécialement dédié à l'Internet des Objets.

2. Sécuriser les interactions entre les objets

2.1. Contexte

Avec l'expansion de l'IdO, le nombre de systèmes embarqués ne cesse d'augmenter, et ce, dans tous les secteurs d'activité (*e.g.* le médical, l'industrie, l'environnement, etc.). Cette omniprésence implique un nombre considérable d'interactions entre les objets, mais aussi une forte manipulation des données. Face à ce phénomène, la recherche et l'étude de solutions de sécurité adaptées à de tels systèmes, en termes de taille, de performance et de robustesse, sont au centre de nombreux travaux. La cryptographie constitue un moyen efficace d'assurer la confidentialité, l'intégrité et l'authenticité des objets et de leurs communications. Malgré leur forte popularité, il est communément admis que les standards comme l'AES exigent trop de ressources, les rendant inefficaces dans un contexte d'objets fortement contraints. Pour cette raison, l'intérêt de la cryptographie légère prend une tout autre ampleur. Parmi elles, PRESENT ainsi que le récemment publié GIFT, adoptent la même structure que l'AES, rendant possible les comparaisons directes. Ce premier chapitre de cette thèse propose plusieurs évaluations de performances et de robustesse de ces trois algorithmes, afin d'identifier la solution la mieux adaptée pour sécuriser les objets fortement contraints.

2.2. État de l'art

2.2.1. Le standard AES

Suite à de nombreuses critiques, mais aussi à la publication de plusieurs attaques à son encontre dès la fin des années 1990, l'algorithme de chiffrement Data Encryption Standard (DES) doit se trouver un remplaçant plus sécurisé [16]. C'est en 2001, lors d'un appel à candidatures lancé quatre ans plus tôt par l'agence américaine National Institute Standard of Technology (NIST), qu'un nouvel algorithme est sélectionné pour devenir l'Advanced Encryption Standard (AES) [13]. Aujourd'hui encore, l'AES reste mathématiquement sécurisé, ce qui lui vaut d'être l'algorithme de chiffrement symétrique de référence et le plus utilisé. Afin de parfaire la compréhension de ce chapitre, son fonctionnement est décrit ci-après.

L'AES est un chiffrement symétrique par bloc : n bits du message clair en entrée produisent n bits du message chiffré en sortie. Dans le cas de l'AES, les messages clairs et chiffrés sont de 128 bits. La clé de chiffrement, l'unique secret qui assure la sécurité de l'algorithme, peut-être de 128, 192 ou 256 bits, définissant ainsi la variante de l'AES utilisée, nommée respectivement AES-128, AES-192 et AES-256. De plus, il peut être utilisé suivant plusieurs modes opératoires, définissant la manière dont sont traités les blocs de messages clairs et chiffrés. Parmi eux, on note le mode *ECB* (*Electronic Code*

Book), *CTR* (*Counter*) ou encore *CBC* (*Cipher Block Chaining*). Dans le cadre de cette étude, le mode *ECB* sera privilégié en raison de sa mise en œuvre plus simple et donc potentiellement plus adapté au contexte de l'IdO.

Le principe de fonctionnement de l'AES repose sur un réseau de substitution et permutation (Substitution-Permutation Network, SPN en anglais). La Figure 2-1 détaille l'algorithme l'AES-128.

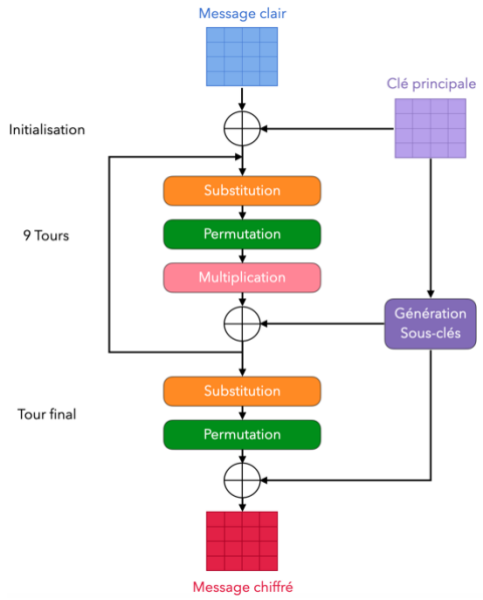


Figure 2-1 - Algorithme de l'AES-128

L'AES ordonne ses données sous forme d'une matrice 4x4. Chaque case de la matrice représente un octet de données, soit 8 bits. L'algorithme est constitué de quatre opérations, regroupées en plusieurs tours, ainsi que d'un générateur de sous-clé. En fonction de la variante de l'AES choisie, 10, 12 ou 14 tours sont nécessaires pour effectuer le chiffrement complet. Par exemple, pour l'AES-128 on distingue une initialisation, suivie de 9 tours et un tour final, comme suit :

Initialisation : cette étape consiste à mélanger le message clair avec la clé principale, à l'aide de l'opérateur binaire *XOR* (⊕).

Les 9 tours : En premier lieu vient l'opération de substitution, permettant notamment de répondre à l'un des principes de base de la cryptographie moderne énoncés par Claude Elwood Shannon [17], la confusion. Cette notion stipule que chaque bit du texte chiffré doit dépendre de manière hautement non linéaire aux bits du texte clair et de la clé. Pour cela, l'opération de substitution convertit un octet de données en un autre octet, suivant une table de conversion prédéfinie, comme indiqué sur la Figure 2-2. Les couches de permutation (Figure 2-3) et de multiplication (Figure 2-4)

quant à elles, correspondent à l'idée de diffusion, second principe de Shannon. Une modification même mineure du message clair doit impacter un grand nombre de bits du message chiffré. Autrement dit, chaque symbole chiffré doit dépendre de nombreux symboles clairs. Plus précisément, l'opération de permutation correspond à un décalage des lignes de la matrice du message clair, alors que l'opération de multiplication applique une transformation linéaire dans le corps de Galois GF(2^8), entre les colonnes de la matrice du message clair et une matrice prédéfinie. Enfin, la sous-clé du tour correspondant, générée à partir de la clé principale, est mélangée avec le résultat des opérations précédentes (Figure 2-5).

Tour final : le résultat de ce dixième et dernier tour correspond au message chiffré. Ce tour est identique aux précédents à l'exception de l'opération de multiplication qui n'y figure pas. Cette décision a été prise par les concepteurs de l'AES, afin d'avoir une structure similaire pour les opérations de chiffrement et de déchiffrement. De plus, ils affirment que cette omission n'impacte pas la sécurité globale de l'algorithme. Toutefois, les travaux de Dunkelman et Keller [18] soulignent une diminution de la robustesse sur des variantes de l'AES utilisant moins de tours, comme l'AES-192 sur 7 tours et par conséquent fragilise l'algorithme dans son ensemble.

Générateur de sous-clés : chaque tour de l'AES a sa propre clé, dérivée de la clé principale. Cette extension de clés suit les mêmes propriétés que les tours, à savoir la confusion et la diffusion. Son principe de fonctionnement est détaillé sur la Figure 2-6.

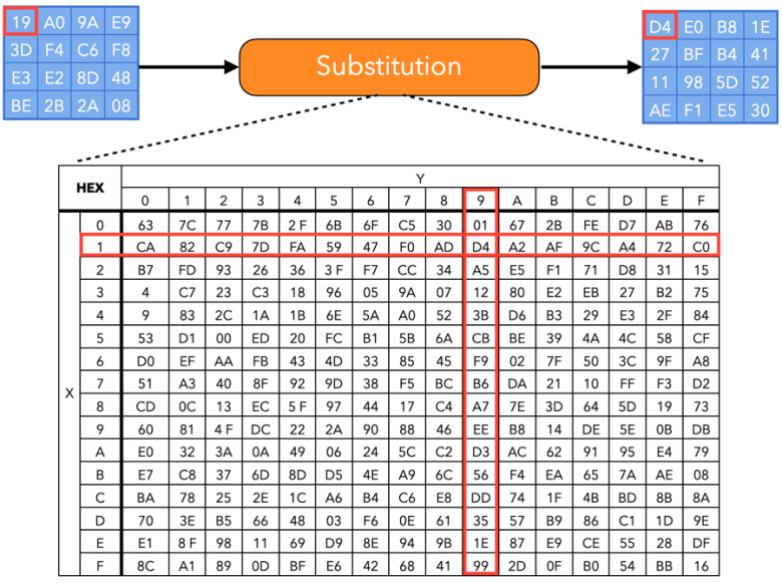


Figure 2-2 - Substitution de l'AES-128

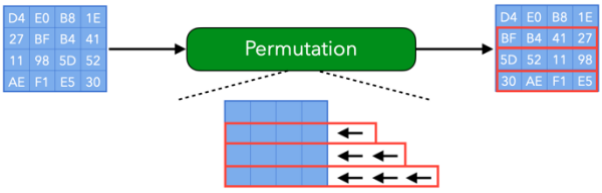


Figure 2-3 - Permutation de l'AES-128

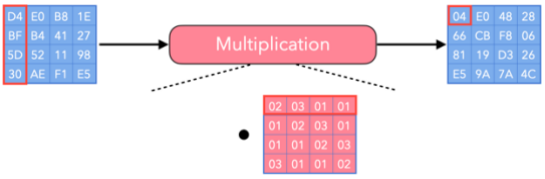


Figure 2-4 - Multiplication de l'AES-128

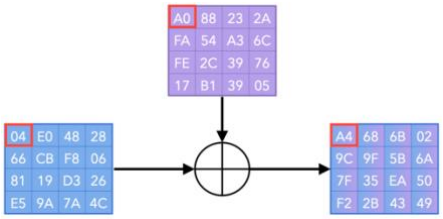


Figure 2-5 - Mélange de la clé de l'AES-128

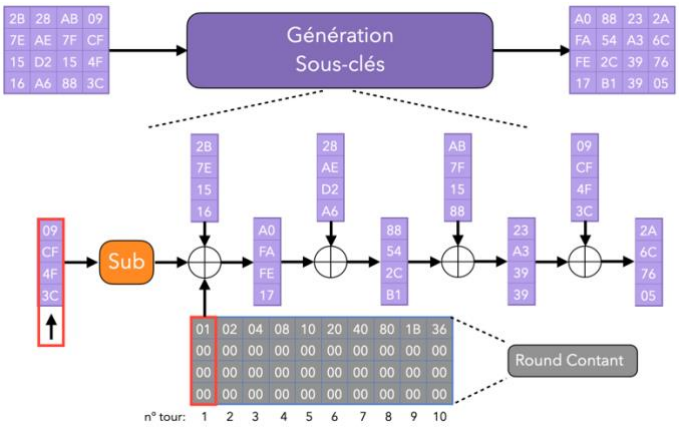


Figure 2-6 - Générateur de sous-clé de l'AES-128

Bien que l'AES soit le standard de la cryptographie symétrique par bloc, il n'est pas conçu pour les architectures fortement contraintes tels que les simples capteurs connectés. Une des principales raisons concerne sa taille. Que ce soit dans sa version logicielle ou matérielle, l'implémentation de l'AES a un cout en termes de taille de code (version logicielle) ou de surface silicium (version

matérielle). Plusieurs études [19]–[23] ont été menées pour tenter d'optimiser l'AES. Les résultats ont cependant démontré l'incompatibilité de ce dernier à être à la fois compact et efficace. Afin d'assurer la sécurité tout en préservant l'utilisation des ressources, plusieurs algorithmes légers ont vu le jour cette dernière décennie. Parmi eux, la famille de chiffrement à flot ChaCha ainsi que les chiffrements authentifiés ASCON et ACORN sont largement étudiés dans les travaux [24]. LED [25], MIDORI [26] et SKINNY [27] sont eux des algorithmes de chiffrements légers par bloc, basés sur la même structure que l'AES. PRESENT [14] et GIFT [15] fonctionnent également sur le principe d'un SPN comme les algorithmes précédents, toutefois ils diffèrent dans leur approche orientée sur la manipulation des bits au lieu de mot (comme les octets). Cette structure ouvre de nouvelles opportunités sur les optimisations matérielles puisque ce type de manipulation est plus efficace sur les implémentations matérielles que logicielles.

2.2.2. PRESENT, ISO-IEC 29192-2 de la cryptographie symétrique légère par bloc

Aujourd'hui, PRESENT est une référence de la cryptographie symétrique légère par bloc et est inclus dans le standard ISO-IEC 29192-2 [28]. Comme pour l'AES, PRESENT possède deux variantes, dépendantes de la longueur de la clé principale. On note PRESENT-80 et PRESENT-128 les versions utilisant 80 bits et 128 bits de clé respectivement. Les concepteurs recommandent l'utilisation de la version 80 bits afin d'avoir un algorithme plus léger pour les architectures fortement contraintes. Toutefois, le nombre total de tours reste inchangé dans les deux cas. Seul le générateur de sous clés est différent. La Figure 2-7 reprend la structure de PRESENT.

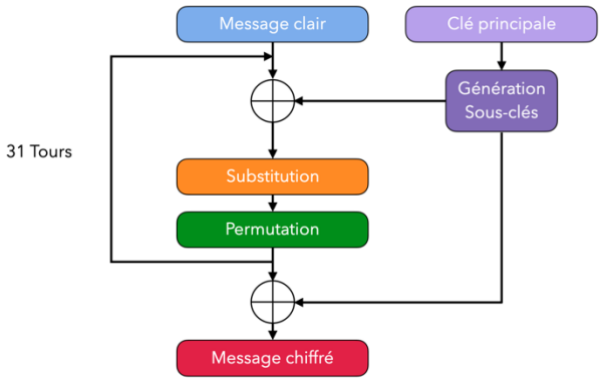


Figure 2-7 - Algorithme de PRESENT

De manière générale, on constate que la structure de PRESENT est très similaire à celle de l'AES. L'algorithme est constitué de plusieurs tours, composés de trois opérations : mélange du message clair avec la clé à l'aide de l'opérateur binaire *XOR* (⊕), une couche de substitution suivie d'une

permutation. Le principe de l'opération de substitution est identique à celle de l'AES, cependant dans le cas de PRESENT elle opère sur des mots de 4 bits contrairement aux 8 bits de l'AES. La table de substitution est détaillée sur la Figure 2-8. Comme évoqué précédemment, PRESENT est orienté sur une manipulation au niveau binaire. Cela provient principalement de l'opération de permutation, dont l'objectif est de répondre à la propriété de diffusion, en permutant la position de différents bits, comme indiqué sur la Figure 2-9. Comparée à l'AES, cette opération est l'équivalent léger des opérations de permutation et de multiplication.

Entrée	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Sortie	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

Figure 2-8 - Substitution de PRESENT

Position bits d'entrée	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Position bits de sortie	0	16	32	48	1	17	33	49	2	18	34	50	3	19	35	51	4	20	36	52	5	21	37	53	6	22	38	54	7	23	39	55

Position bits d'entrée	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
Position bits de sortie	8	24	40	56	9	25	41	57	10	26	42	58	11	27	43	59	12	28	44	60	13	29	45	61	14	30	46	62	15	31	47	63

Figure 2-9 - Permutation de PRESENT

Tout comme l'AES, PRESENT utilise une sous-clé à chaque tour. En raison de la différence de taille des données à chiffrer (64 bits) et de la clé principale (80 ou 128 bits), chaque sous-clé correspond aux 64 bits de poids fort de la clé principale. Ensuite, cette dernière est mise à jour suivant une couche de permutation, de substitution et un mélange avec un compteur 5 bits. Le processus de mise à jour de la clé principale est légèrement différent suivant la version PRESENT-80 et PRESENT-128, comme le montrent les Figure 2-10 et Figure 2-11.

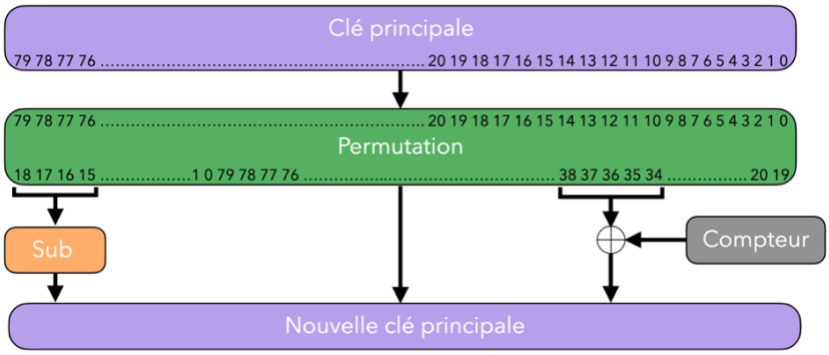


Figure 2-10 - Mise à jour de la clé principale de PRESENT-80

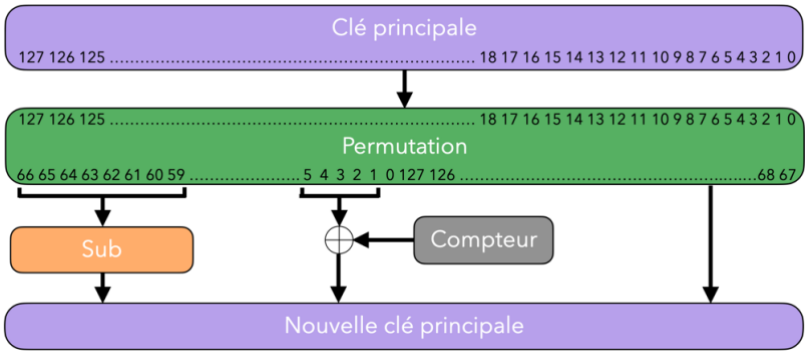


Figure 2-11 - Mise à jour de la clé principale de PRESENT-128

Pour la version PRESENT-80, la sous-clé d'un tour correspond aux bits 79 à 16 de la clé principale. Puis cette dernière subit une rotation vers la gauche de 61 bits, une opération de substitution est ensuite effectuée sur les quatre bits de poids fort et la valeur du compteur (*cf.* le numéro du tour) est mélangée avec les bits 19 à 15.

2.2.3. GIFT, amélioration de PRESENT

Depuis plus d'une dizaine d'années, la communauté scientifique a fourni d'importants efforts dans la mise en œuvre de nouveaux algorithmes de chiffrements légers. L'un des derniers chiffrements légers par bloc en date se nomme GIFT [15]. Présenté lors de sa sortie en 2017 comme étant l'algorithme le plus léger et le moins énergivore, GIFT est une amélioration de PRESENT. En effet, il partage la même structure que ce dernier, ce qui lui vaut également des similitudes avec l'AES. De plus, il permet de corriger certaines faiblesses de PRESENT telle que l'approximation linéaire [29]. Enfin, ses concepteurs soulignent que les caractéristiques de GIFT dépassent, dans une certaine implémentation, celles de SKINNY [27] apparu un an auparavant.

GIFT est lui aussi un SPN, se déclinant en deux variantes : l'une permet de chiffrer un message de 64 bits en 28 tours, et l'autre un message de 128 bits en 40 tours. En revanche la taille de la clé est de 128 bits pour les deux variantes. Ces dernières sont nommées GIFT-64-128 et GIFT-128-128 respectivement. La Figure 2-12 présente l'algorithme de GIFT.

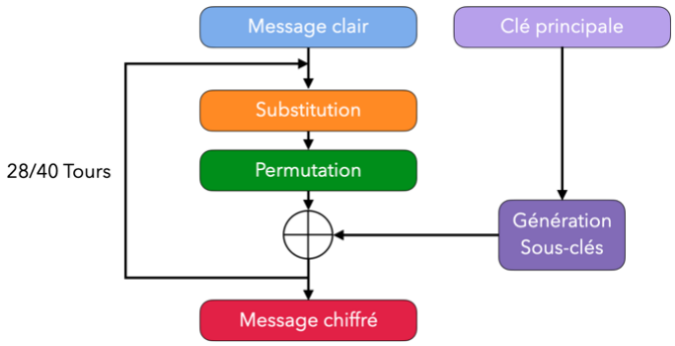


Figure 2-12 - Algorithme de GIFT-64-128 et GIFT-128-128

À première vue, les structures de PRESENT et GIFT sont identiques : les tours sont constitués d'une opération de substitution, de permutation et du mélange de la clé par l'opérateur binaire *XOR* (⊕). Seule la position de ce dernier diffère, en se trouvant à la fin du tour dans le cas de GIFT. Toutefois, les opérations de GIFT sont différentes de PRESENT. En effet, la table de substitution de GIFT a évolué comparé à celle de PRESENT comme le montre la Figure 2-13. Selon ses concepteurs, cette mise à jour permet une réduction de la taille d'implémentation ainsi qu'une amélioration contre les attaques par cryptanalyse [15]. Dans cette même optique, une autre modification intervient dans la couche de permutation. De plus, puisque GIFT opère soit sur un message de 64 bits soit de 128 bits, l'opération de permutation est différente suivant la variante choisie comme indiqué sur les Figure 2-14 et Figure 2-15 respectivement.

Entrée	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Sortie	1	A	4	C	6	F	3	9	2	D	B	7	5	0	8	E

Figure 2-13 - Substitution de GIFT

Position bits d'entrée	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Position bits de sortie	0	17	34	51	48	1	18	35	32	49	2	19	16	33	50	3	4	21	38	55	52	5	22	39	36	53	6	23	20	37	54	7
Position bits d'entrée	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
Position bits de sortie	8	25	42	59	56	9	26	43	40	57	10	27	24	41	58	11	12	29	46	63	60	13	30	47	44	61	14	31	28	45	62	15

Figure 2-14 - Permutation de GIFT-64-128

Position bits d'entrée	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Position bits de sortie	0	33	66	99	96	1	34	67	64	97	2	35	32	65	98	3	4	37	70	103	100	5	38	71	68	101	6	39	36	69	102	7
Position bits d'entrée	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
Position bits de sortie	8	41	74	107	104	9	42	75	72	105	10	43	40	73	106	11	12	45	78	111	108	13	46	79	76	109	14	47	44	77	110	15
Position bits d'entrée	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
Position bits de sortie	16	49	82	115	112	17	50	83	80	113	18	51	48	81	114	19	20	53	86	119	116	21	54	87	84	117	22	55	52	85	118	23
Position bits d'entrée	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
Position bits de sortie	24	57	90	123	120	25	58	91	88	121	26	59	56	89	122	27	28	61	94	127	124	29	62	95	92	125	30	63	60	93	126	31

Figure 2-15 - Permutation de GIFT-128-128

Une autre différence intervient dans le mélange de la clé avec le message à chiffrer. En effet, la clé principale de GIFT est divisée en 8 mots de 16 bits. Inspiré du même principe que PRESENT, la sous-clé de chaque tour correspond directement à une partie de la clé principale et sa taille correspond à la moitié de celle du message à chiffrer. Dans le cas de GIFT-64-128 il s'agit des deux mots de poids faible M2 et M1, soit 32 bits au total, alors que pour GIFT-128-128, la sous-clé est composée des mots M6, M5, M2 et M1, soit 64 bits au total. Ces extractions sont représentées sur les Figure 2-16 et Figure 2-17.

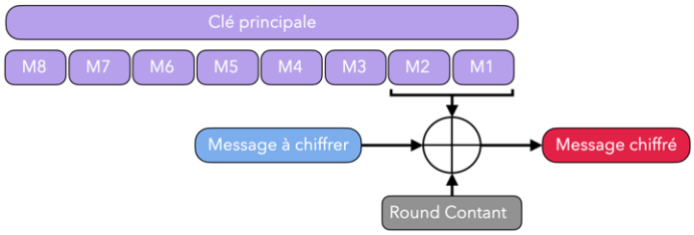


Figure 2-16 - Extraction de la sous-clé de GIFT-64-128

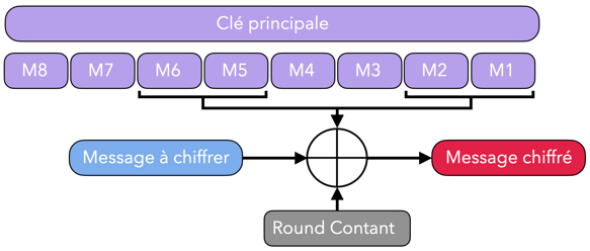
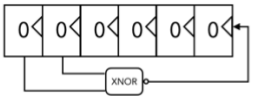


Figure 2-17 - Extraction de la sous-clé de GIFT-128-128

Une fois extraite, la sous-clé est ensuite mélangée avec le message à chiffrer ainsi qu'avec un nombre constant (*Round Constant*) sur 6 bits, généré par un LFSR (*Linear Feedback Shift Register*),

représenté sur la Figure 2-18. Cette méthode permet d'être plus légère que le compteur de PRESENT d'un point de vue implémentation matérielle.



N° Tour	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Round Constant	01	03	07	0F	1F	3E	3D	3B	37	2F	1E	3C	39	33	27	0E	1D	3A	35	2B	16	2C	18	30

N° Tour	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
Round Constant	21	02	05	0B	17	2E	1C	38	31	23	06	0D	1B	36	2D	1A	34	29	12	24	08	11	22	04

Figure 2-18 - Round Constant de GIFT

La répartition du mélange entre le message à chiffrer, la sous-clé et le Round Constant est présentée sur la Figure 2-19 et Figure 2-20 pour les versions GIFT-64-128 et GIFT-128-128 respectivement. À noter, une particularité de cette opération est le mélange du message avec un '1' sur le bit de poids fort, présent sur les deux variantes.

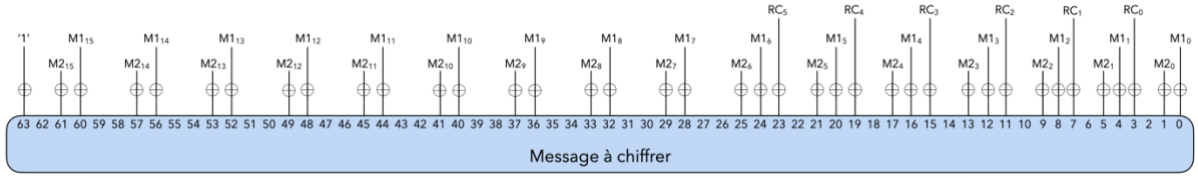


Figure 2-19 - Mélange de la clé de GIFT-64-128

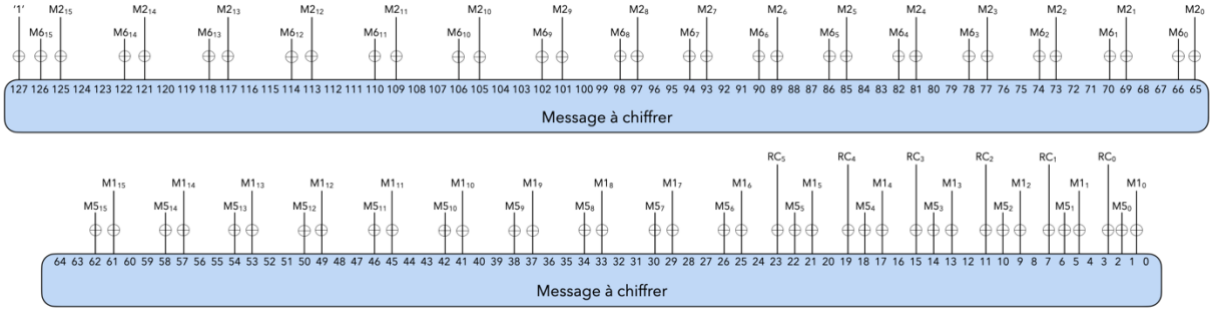


Figure 2-20 - Mélange de la clé de GIFT-128-128

Une fois le mélange effectué, le *Round Constant* ainsi que la clé principale peuvent être mis à jour. À l'inverse de PRESENT, cette mise à jour est réduite à de simples décalages de bits, ce qui permet d'alléger la structure de GIFT. En effet, dans un premier temps, les deux mots de 16 bits les plus à droite (M2 et M1) subissent un décalage interne de 2 et 12 bits respectivement. Puis, tous les mots sont décalés vers la droite les uns par rapport aux autres comme présenté sur la Figure 2-21.

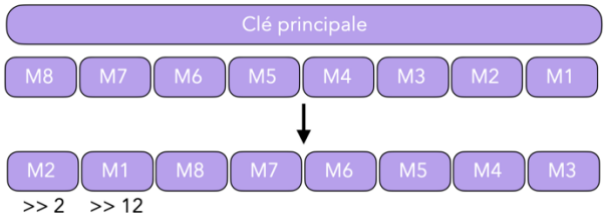


Figure 2-21 - Mise à jour de la clé principale de GIFT

2.2.4. Différents types d'implémentation

Dès lors qu'il y a un besoin de sécurité, les algorithmes cryptographiques font l'objet d'une attention particulière. En effet, le choix de ces derniers dépend de plusieurs facteurs, comme le niveau de sécurité désiré, le cout de déploiement, mais également des performances recherchées. La Figure 2-22, inspirée de [30], illustre l'équilibre à trouver entre ces éléments.

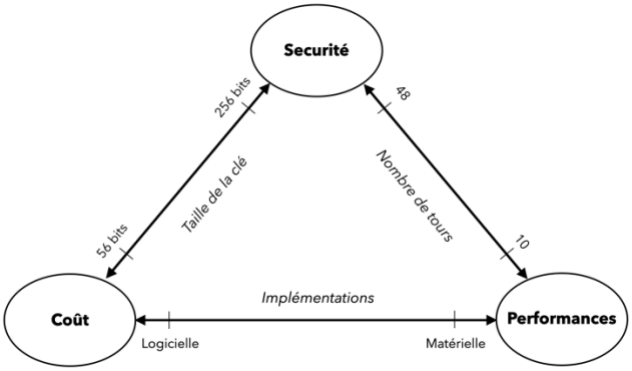


Figure 2-22 - Équilibre entre sécurité, cout et performances, inspiré de [30]

Théoriquement, le niveau de sécurité est intrinsèquement lié à l'algorithme. Plus la taille de la clé est grande, plus le niveau de sécurité est élevé. Il en est de même pour le nombre de tours à effectuer. En revanche, la taille de la clé alourdit les opérations de l'algorithme, le rendant plus complexe et donc augmentant son cout. D'autre part, le nombre de tours impacte directement les performances, rendant l'opération plus longue à effectuer. Enfin, l'implémentation de la solution cryptographique a des conséquences non négligeables sur le cout mais aussi sur les performances. De plus, quand bien même un algorithme est considéré comme sécurisé théoriquement, l'implémentation de celui-ci peut introduire des failles sécuritaires importantes, telles que les attaques par canaux auxiliaires [31], exploitant les temps d'exécution [32], [33] la consommation de courant [34] ou encore le rayonnement électromagnétique [35]. Il convient donc d'étudier précisément les besoins et les caractéristiques de l'application à mettre en œuvre. Par exemple, un tag passif RFID doit être peu couteux, avec une taille d'implémentation minimale et une consommation énergétique

très faible, tandis que les performances (*e.g.* débit) ne sont pas ici un critère décisif. En revanche, le lecteur RFID qui lui doit scanner de nombreux appareils en même temps doit avoir un débit très élevé alors que sa consommation énergétique et sa taille sont de moindre importance. Afin de mettre en lumière leurs caractéristiques, cette section traite des différentes implémentations employées dans l'état de l'art.

a) Implémentation logicielle

La première, la plus intuitive, consiste à inclure directement dans le programme de l'application, les fonctions cryptographiques. L'avantage d'une telle approche réside dans sa simplicité de mise en œuvre ainsi que dans sa forte maintenabilité et portabilité. Toutefois, la taille du code de l'application se voit augmenter, ce qui peut être problématique. En effet, dans le domaine de l'embarqué tels que les objets connectés, la mémoire programme du microcontrôleur est souvent très limitée (de quelques dizaines de Kilooctets à quelques Mégaoctets). De plus, une implémentation logicielle monopolise le microcontrôleur, l'empêchant d'effectuer toute autre tâche pendant l'opération cryptographique, ce qui peut être problématique dans un système temps réel où la réactivité de ce dernier par rapport à son environnement est primordiale. Enfin, certains algorithmes tels que PRESENT et GIFT sont basés sur des manipulations de bits, à l'inverse de l'AES qui lui travaille sur des mots (octets). Bien qu'une implémentation particulière de GIFT [15] permette d'augmenter son efficacité logicielle, les algorithmes utilisant ce type d'opération bas niveau sont efficaces d'un point de vue matériel, mais les rendant moins performants dans leur version logicielle. Le projet FELICS [36] (*Fair Evaluation of Lightweight Cryptographic Systems*) s'efforce de fournir une évaluation de différents algorithmes de chiffrement implémentés sur trois microcontrôleurs susceptibles d'être utilisés dans le domaine des objets connectés : AVR ATmega128 8 bits [37], TI MSP430 16 bits [38], et ARM Cortex M3 32 bits [39]. À titre d'exemple, une implémentation de l'AES-128 (chiffrement et génération des sous-clés inclus) sur une architecture ARM, nécessite 3 282 octets de mémoire programme et 20 265 cycles pour terminer le chiffrement [36]. PRESENT-80, requiert 1 264 octets de mémoire, soit une réduction de plus d'un facteur 2 sur l'espace de stockage, mais pour un temps de chiffrement multiplié par 7 (142 041 cycles). GIFT n'est à ce jour pas encore pris en compte par le projet FELICS. Toutefois, de par les similitudes qu'ils partagent, on peut supposer que leurs efficacités sont du même ordre de grandeur. Il convient de rappeler que ces résultats dépendent fortement de diverses optimisations et configurations des outils de compilations.

b) Implémentations matérielles

Une seconde approche consiste, à déporter la partie cryptographique sur un système secondaire, dans le but de soulager le système principal, laissant ce dernier entièrement dédié à la partie

applicative. L'algorithme de chiffrement est alors implémenté matériellement, ce qui lui vaut l'appellation d'accélérateur matériel. Malgré leur faible maintenabilité (*i.e.* mise à jour ou changement d'algorithmes de chiffrement) par rapport à leurs homologues logiciels, les implémentations matérielles s'avèrent être une excellente alternative pour des objets fortement contraints. Cette section recense les différentes architectures matérielles employées dans l'état de l'art, en soulignant pour chacune d'entre elles leurs avantages et inconvénients.

Implémentation parallèle : ce type d'implémentation consiste dérouler la totalité de l'algorithme à implémenter, et connecter en cascade chacun de ses tours, comme indiqué sur la Figure 2-23. À noter que la génération des sous-clés est également incluse dans les tours. Ainsi, chaque tour s'exécute en parallèle, et un seul cycle d'horloge est nécessaire pour effectuer un chiffrement complet. Cette implémentation permet donc de maximiser le débit, au détriment de la taille d'implémentation et de la consommation énergétique [40]. Cependant, l'implémentation parallèle ne permet pas d'atteindre des fréquences élevées. En effet, chaque opérateur électronique constituant un tour possède un temps de réponse non nul. En raison de la mise en cascade des tours, le temps de réponse total du premier au dernier tour correspond à la somme des temps de réponse de ces éléments. Ainsi, il existe un temps minimal et donc une fréquence maximale jusqu'à laquelle l'implémentation parallèle peut fonctionner. Bien que le nombre de cycles soit minimal, cette approche ne constitue pas la meilleure alternative pour les fonctionnements à de hautes fréquences.

Figure 2-23 - Implémentation parallèle

Implémentation à pipelines : la principale solution pour remédier au problème de fréquence de l'implémentation parallèle consiste à réduire ce chemin critique. Pour cela, des registres sont insérés entre les tours, créant ainsi une implémentation à pipelines, illustrée par la Figure 2-24. Plusieurs configurations sont possibles : les registres peuvent être insérés entre chaque tour (*Fully pipelined*), uniquement entre certains d'entre eux (*Partial pipelined*), ou encore à la fois entre et à l'intérieur des tours (*Subpipelined*) comme étudiés dans les travaux [40] et [41]. De manière imagée, les registres permettent de « fractionner » le chemin critique en plusieurs, plus courts, de part et d'autre du registre. Par conséquent, plus le nombre de registres est important, plus la réduction des chemins est importante, augmentant ainsi la fréquence maximale du système. Toutefois, l'ajout de ces registres impose un cout supplémentaire sur la taille d'implémentation. De plus, ces derniers imposent un ajout supplémentaire de cycles au démarrage, proportionnel au nombre de registres ajoutés. Cela signifie qu'une implémentation à pipelines fournit un chiffrement tous les cycles d'horloge (comme

une implémentation parallèle classique), mais au bout de *n* cycles après l'application de la première donnée à chiffrer, le temps de propager l'information sur les différentes couches de registres.

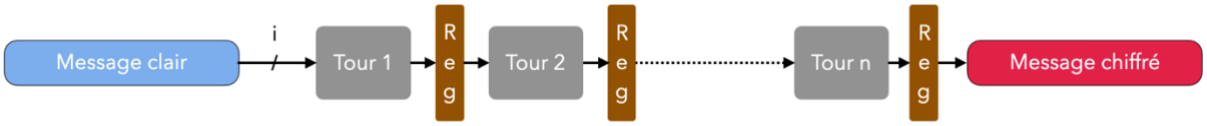


Figure 2-24 - Implémentation à pipelines

Implémentation à tours : cette implémentation consiste à ne construire qu'un tour de l'algorithme de chiffrement, sur lequel un rebouclage est effectué, pour exécuter les différents tours requis (par exemple, 10 pour l'AES-128), comme le montre la Figure 2-25. Un cycle d'horloge est nécessaire pour effectuer un tour complet. La latence correspond donc ici au nombre de tours défini par l'algorithme, ce qui réduit le débit comparé aux implémentations parallèles ou à pipelines. Toutefois, cette approche a l'avantage de réduire très fortement la taille de l'algorithme (théoriquement d'un facteur proportionnel au nombre de tours de l'algorithme comparé aux implémentations parallèles/pipelines), et peut atteindre de hautes fréquences de fonctionnement. Bien que négligeable, il convient de noter que de la logique additionnelle est nécessaire pour contrôler le flot du chiffrement (par exemple, à l'aide d'une machine d'état). Ainsi, l'implémentation à tours constitue un compromis idéal entre la taille et le débit.

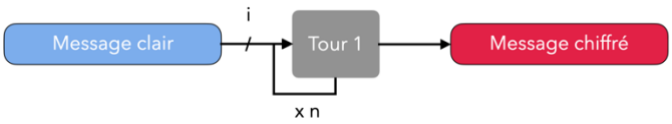


Figure 2-25 - Implémentation à tours

Implémentation sérielle : à l'opposé des implémentations parallèles et à pipelines, l'implémentation sérielle consiste à réduire drastiquement la taille de l'algorithme de chiffrement au détriment du débit. Cette approche peut être appréhendée comme une variante de l'implémentation à tours. En effet, elle est elle aussi basée sur un rebouclage sur un seul tour. Cependant, sa particularité réside dans la réduction du bus de données, permettant de diminuer le nombre d'opérateurs, comme sur la Figure 2-26. La taille du nouveau bus de données peut être configurée en fonction du niveau de sérialisation souhaité. Un cas d'école est celui de l'AES-128. Celui-ci utilise un bus de données sur 128 bits et son opération de substitution est par défaut définie sur 8 bits. Cela signifie que dans une implémentation à tours, cette opération est instanciée 16 fois (16 * 8 bits pour les 128 bits au total). Pour une implémentation sérielle efficace, la taille des données peut être réduite à 8 bits, permettant de n'avoir qu'une seule instanciation de l'opération de substitution. Ainsi, plus le niveau de

sérialisation est élevé, plus la taille des données est réduite. Toutefois, cela implique une augmentation du nombre de cycles. En effet, puisqu'un seul tour ne traite qu'une partie des données totales, plus de tours sont nécessaires pour traiter l'ensemble des données, impliquant une longue période d'activité du circuit. Tout comme pour l'implémentation à tours, cette approche permet un haut niveau de fréquence, et de la logique additionnelle est nécessaire. Cependant, suivant le niveau de réduction de la taille de l'algorithme, cet ajout peut ici s'avérer être démesuré, réduisant à néant l'intérêt d'une telle approche.

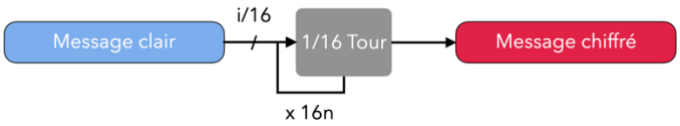


Figure 2-26 - Implémentation sérielle

2.2.5. Implémentations matérielles sur FPGA

Les sections précédentes soulignent l'intérêt qu'apporte une implémentation matérielle des fonctions cryptographiques. De plus, les algorithmes cryptographiques légers comme PRESENT et GIFT, sont destinés aux objets fortement contraints, ce qui exclut toute implémentation parallèle ou à pipelines. Cela se confirme dans l'état de l'art où la plupart des travaux de recherche dans le contexte de l'IdO se focalisent sur l'implémentation à tours, mais également sur l'approche série en raison de sa faible consommation énergétique et de sa taille réduite.

Une manière de réaliser les implémentations matérielles est le prototypage sur FPGA (*Field Programmable Gate Array*). Ce dernier a l'avantage d'être moins onéreux et plus rapide à développer que son homologue ASIC (*Application-Specifique Integrated Circuit*). De plus, grâce à sa propriété reprogrammable (contrairement aux ASICs) il représente un outil de prédilection pour le prototypage. Cette partie dresse l'état de l'art des différentes implémentations de l'AES, PRESENT et GIFT sur FPGA.

Tout d'abord, lorsqu'il est question de comparaisons matérielles, il n'est pas toujours possible d'opposer les implémentations de manière totalement équitable. Tout d'abord, en raison de l'évolution technologique, les FPGA ne sont pas constitués des mêmes éléments logiques, ce qui impacte leurs performances (taille d'implémentation, fréquence, consommation, etc.). Afin d'illustrer ces différences, le Tableau 2-1 reprend la composition des *slices* en termes de *LUT* et de *Flip-Flop* (*FF*), des différentes séries du constructeur Xilinx. Grâce à l'évolution de l'architecture des *slices*, une implémentation sur un FPGA de série 7 (*e.g. Spartan-7)* aura une taille inférieure à celle sur un *Spartan-III,* en termes de nombre de *slices* utilisés. De même, les fréquences maximales

atteignables des implémentations varient suivant le type de FPGA. Les familles *Spartan* et *Artix* correspondent aux entrées de gamme, conçues pour minimiser la consommation énergétique, réduisant inévitablement leurs performances. À l'opposé, les *Kintex* et enfin les *Virtex* sont dédiés aux hautes performances, au détriment d'une consommation plus importante. Ainsi, une même implémentation atteindra une fréquence maximale plus importante sur une gamme *Virtex* que sur un *Spartan* de même génération. Mais la difficulté de comparaison ne s'arrête pas uniquement au type de série choisi. Suivant le modèle de FPGA mais aussi le *Speed Grade* et le *package* d'une même série (*e.g. Artix-7 XC7A100T-1CSG324C* et *Kintex-7 XC7K70T-3FBG484*), le placement des éléments et les fréquences atteignables sont différents, ce qui là encore impacte l'implémentation. Enfin, de nombreux éléments de configuration comme la fréquence d'utilisation, mais aussi les différentes options de l'outil de développement (*e.g. AreaOptimized_High*, *Flatten_Hierarchy* sur *Vivado Design Suite* [42] de Xilinx) ainsi que certains attributs propres au langage de description matériel (*e.g.* « *don't touch* » [42] du VHDL) ont un impact significatif sur les performances et la taille des implémentations. De plus, de nombreux travaux ne détaillent pas entièrement les configurations utilisées, ce qui complique davantage les comparaisons.

Tableau 2-1 - Composition des *slices* des différentes séries de FPGA Xilinx

Année	Séries	Composition d'un *slice*	
		LUT	*Flip-Flop* (*FF*)
2000	Spartan-II [43]	2 *LUT*-4 entrées	2
2004	Virtex-4 [44]		
2006	Spartan-III [45]		
2006	Virtex-5 [46]	4 *LUT*-6 entrées	4
2009	Spartan 6 [47]	4 *LUT*-6 entrées	8
2011	Séries 7 [48]		

Outre l'aspect technologique, les différentes approches choisies par les concepteurs lors de l'évaluation de leur implémentation entrainent elles aussi des difficultés de comparaison. Cela s'avère particulièrement vrai lorsque ces derniers ne fournissent pas suffisamment de détails. L'élément critique concerne le nombre de cycles requis pour effectuer l'opération cryptographique complète. En effet, ce facteur est à l'origine des performances de l'implémentation en termes de débit, d'efficacité et d'énergie, comme le souligne les équations (2-1), (2-2) et (2-3). Suivant les travaux, le nombre de cycles considéré correspond soit strictement au temps de calcul de l'opération cryptographique (*cf.* nombre de tours ou portion de tour), soit en ajoutant à ces derniers le temps de chargement et de récupération de la donnée à chiffrer et du résultat. Bien que cette dernière approche

offre une vision plus réaliste (souvent définie comme la latence), elle est dépendante de l'interfaçage entre le cœur cryptographique et le reste du système, comme illustrée sur la Figure 2-27.

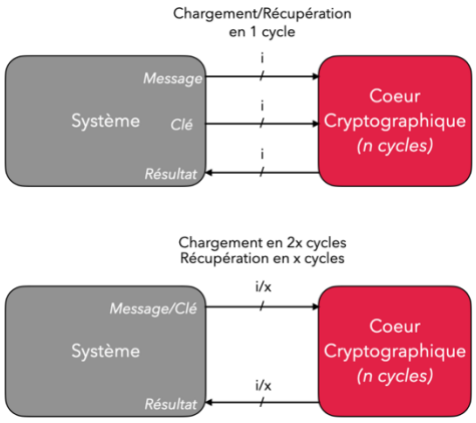


Figure 2-27 - Interfaçage avec le cœur cryptographique

Suivant la méthode employée, le nombre de cycles total varie, indépendamment de la partie cryptographique. Ce travail étant focalisé sur cette dernière, seul le nombre de cycles nécessaires pour effectuer l'opération est pris en considération. Chaque implémentation de l'état de l'art est analysée afin d'uniformiser autant que faire se peut les résultats. Par ailleurs, on constate d'après les équations (2-1), (2-2) et (2-3) que la fréquence a un impact important sur les performances du système. Les résultats présents dans l'état de l'art sont réalisés aux fréquences maximales des implémentations. Toutefois, bien qu'il puisse être intéressant d'évaluer leurs performances dans de telles conditions, en pratique les systèmes de l'IdO sont souvent cadencés à des fréquences prédéfinies par des normes. C'est notamment le cas des cartes sans contact, fonctionnant à 13,56 MHz, définies par l'ISO/IEC 14443-2 [49]. Dans le but d'avoir des comparaisons les plus fiables possibles, les tableaux ci-après fournissent une normalisation du débit et de l'efficacité des implémentations de l'état de l'art à 13,56 MHz.

$$Débit = \frac{Taille_{Données}}{NB_{Cycles}} \times Fréquence \tag{2-1}$$

$$Efficacité = \frac{Débit}{Surface} \tag{2-2}$$

$$Energie = \frac{NB_{Cycles} \times Puissance_{Consommée}}{Fréquence} \tag{2-3}$$

Après avoir éclairé le lecteur des difficultés de comparaison et spécifié les hypothèses choisies, la suite de cette section dresse l'état de l'art des implémentations sur FPGA. En raison de sa maturité et

de sa popularité, l'AES a été intensivement étudié. De ce fait, le Tableau 2-2 dresse une liste non exhaustive des travaux sur l'AES-128. Les Tableau 2-3 et Tableau 2-4 reprennent les implémentations de PRESENT-80 et PRESENT-128 respectivement. Enfin, les Tableau 2-5 et Tableau 2-6 celles de GIFT-64-128 et GIFT-128-128.

Tableau 2-2 - État de l'art des implémentations de l'AES-128 sur FPGA

N°	Papiers	Implémentation	Cycles	Surface (*Slices*)	Fréquence Max (MHz)	Normalisation @13,56 MHz	
						Débit (Mbps)	Efficacité (Mbps/*Slices*)
Spartan-II							
1	[40]	Sérielle[1] 8 bits	-	124 + 2 BRAM	67	-	-
2	[50]	Sérielle[2] 8 bits	160	130 + 4 BRAM	34	10,85	-
3	[51]	Sérielle[2] 8 bits	160	258	66	10,85	0,042
4	[52]	Sérielle[1] 32 bits	40	222 + 3 BRAM	60	43,39	-
Virtex-II							
5	[53]	Sérielle 8 bits	144	333	56	12,05	0,036
6	[53]	Sérielle 8 bits[3]	144	267	43,79	12,05	0,045
7	[54]	Sérielle[1] 32 bits	44	146 + 3 BRAM	123	39,44	-
8	[55]	Tours	-	1780	77,91	-	-
9	[53]	Tours	10	1672	74,42	173,57	0,104
10	[53]	Tours[3]	10	976	60,94	173,57	0,178
Spartan-III							
11	[56]	Sérielle 8 bits	160	184	45,642	10,85	0,059
12	[57]	Sérielle 8 bits	-	177	-	-	-
13	[54]	Sérielle[1] 32 bits	44	163 + 3 BRAM	71,5	39,44	-
14	[58]	Sérielle 32 bits	-	234 + 3 BRAM	-	-	-
15	[59]	Tours	10	1877	-	173,57	0,092
16	[55]	Tours	-	1800	150	-	-
17	[60]	Tours	-	392 + 10 BRAM	-	-	-
Virtex-5							
18	[53]	Sérielle 8 bits	144	113	113,25	12,05	0,107
19	[53]	Sérielle 8 bits[3]	144	137	77,29	12,05	0,088
20	[57]	Sérielle 8 bits	-	65	-	-	-
21	[58]	Sérielle 32 bits	-	155	-	-	-
22	[55]	Tours	-	349	350	-	-
23	[53]	Tours	10	359	96,04	173,57	0,483

24	[53]	Tours[3]	10	456	136,84	173,57	0,380
25	[60]	Tours	-	342	-	-	-
Spartan-6							
26	[56]	Sérielle 8 bits	160	80	72,66	10,85	0,135
27	[57]	Sérielle 8 bits	-	65	-	-	-
28	[58]	Sérielle 32 bits	-	133	-	-	-
29	[60]	Tours	-	332	-	-	-
Artix-7 (Série 7)							
30	[57]	Sérielle 8 bits	-	87	-	-	-
31	[58]	Sérielle 32 bits	-	133	-	-	-
32	[60]	Tours	-	332	-	-	-
33	[61]	Tours	10	530[4]	141,418	173,57	0,327
34	[62]	Tours	10	522	149,502	173,57	0,332
Kintex-7 (Série 7)							
35	[63]	Sérielle 8 bits	308	95	287	5,63	0,060
36	[57]	Sérielle 8 bits	-	94	-	-	-
37	[58]	Sérielle 32 bits	-	132	-	-	-
38	[60]	Tours	-	330	-	-	-
Virtex-7 (Série 7)							
39	[57]	Sérielle 8 bits	-	91	-	-	-
40	[58]	Sérielle 32 bits	-	132	-	-	-
41	[60]	Tours	-	330	-	-	-

1 : Chiffrement + Déchiffrement + Clés pré-chargées

2 : Chiffrement + Déchiffrement

3 : Compactes SBOX par corps commutatifs (*Subfield Arithmetic*)

4 : Utilisation de 320 MUXF7 et 160 MUXF8 des slices

Tableau 2-3 - État de l'art des implémentations de PRESENT-80 sur FPGA

N°	Papiers	Implémentation	Cycles	Surface (*Slices*)	Fréquence Max (MHz)	Normalisation @13,56 MHz	
						Débit (Mbps)	Efficacité (Mbps/*Slices*)
Spartan-III							
42	[64]	Sérielle 16 bits	124	124	213,81	7,00	0,057
43	[65]	Sérielle[1] 16 bits	124	256	132,19	7,00	0,027
44	[59]	Tours	31	271	-	28,00	0,103
45	[66]	Tours	31	202	240	28,00	0,139
46	[66]	Tours[2]	31	176	258	28,00	0,159
47	[65]	Tours[1]	31	265	192,23	28,00	0,106

48	[67]	Tours	32	181	196	27,12	0,150
49	[68]	Tours	32	175	251	27,12	0,155
Virtex-5							
50	[64]	Sérielle 16 bits	124	67	542,30	7,00	0,104
51	[68]	Tours	32	55	240	27,12	0,493
52	[69]	Tours	32	57	306,84	27,12	0,476
Spartan-6							
53	[64]	Sérielle 16 bits	124	48	257,40	7,00	0,146
54	[65]	Sérielle[1] 16 bits	124	69	172,92	7,00	0,101
55	[65]	Tours[1]	31	74	221,63	28,00	0,378
56	[69]	Tours	32	59	278	27,12	0,460
Artix *-7 (Série 7)*							
57	[70]	Tours	32	83	303,361	27,12	0,327
Kintex-7 (Série 7)							
58	[63]	Tours	32	103	542	27,12	0,263

1 : Clé pré-chargées (compatible avec PRESENT-80 et PRESENT-128)

2 : SBOX booléennes avec optimisation via Espresso

Tableau 2-4 - État de l'art des implémentations de PRESENT-128 sur FPGA

N°	Papiers	Implémentation	Cycles	Surface (*Slices*)	Fréquence Max (MHz)	Normalisation @13,56 MHz	
						Débit (Mbps)	Efficacité (Mbps/*Slices*)
Virtex-II							
59	[53]	Sérielle 8 bits	279	131	114,17	3,11	0,024
60	[53]	Tours	31	155	155,76	28,00	0,181
Spartan-III							
61	[65]	Sérielle[1] 16 bits	124	256	132,19	7,00	0,027
62	[71]	Sérielle 16 bits	248	117	113,84	3,50	0,030
63	[72]	Sérielle 16 bits	1062	83 + 1 BRAM	129,77	0,82	-
64	[72]	Sérielle[2] 16 bits	1248	85 + 1 BRAM	135,15	0,70	-
65	[64]	Sérielle 16 bits	124	151	194,63	7,00	0,046
66	[65]	Tours[1]	31	265	192,23	28,00	0,105
67	[66]	Tours	31	202	254	28,00	0,134
68	[66]	Tours[3]	31	202	250	28,00	0,134
69	[67]	Tours	32	158	196	27,12	0,172
70	[68]	Tours	32	201	207	27,12	0,135
Virtex-5							
71	[53]	Sérielle 8 bits	279	70	245,76	3,11	0,044
72	[73]	Sérielle[4] 8 bits	279	62	236,574	3,11	0,050

73	[64]	Sérielle 16 bits	124	73	431,78	7,00	0,096
74	[53]	Tours	31	87	250,89	28,00	0,322
75	[69]	Tours	32	69	306,84	27,12	0,393
76	[68]	Tours	32	66	573	27,12	0,411
Spartan-6							
77	[74]	Sérielle 8 bits	279	45	131,87	3,11	0,070
78	[65]	Sérielle[1] 16 bits	124	69	172,92	7,00	0,101
79	[64]	Sérielle 16 bits	124	61	210,66	7,00	0,115
80	[74]	Sérielle 16 bits	124	61	159,21	7,00	0,115
81	[65]	Tours[1]	31	74	221,63	28,00	0,380
82	[74]	Tours	31	56	145,35	28,00	0,500
Artix-7 (Série 7)							
83	[74]	Sérielle 8 bits	279	45	131,87	3,11	0,069
84	[74]	Sérielle 16 bits	124	48	159,21	7,00	0,146
85	[61]	Tours	31	183	324,828	28,00	0,153
86	[70]	Tours	32	93	316,304	27,12	0,292
87	[74]	Tours	31	66	145,35	28,00	0,424

1 : Clé pré-chargées (compatible avec PRESENT-80 et PRESENT-128)

2 : SBOX dans BRAM

3 : SBOX booléennes avec optimisation via Espresso

4 : SBOX booléennes avec optimisation via la table de Karnaugh

Tableau 2-5 - État de l'art des implémentations de GIFT-64-128 sur FPGA

N°	Papiers	Implémentation	Cycles	Surface (*Slices*)	Fréquence Max (MHz)	Normalisation @13,56 MHz	
						Débit (Mbps)	Efficacité (Mbps/*Slices*)
Spartan-6							
88	[74]	Sérielle 8 bits	252	44	225,53	3,44	0,078
89	[74]	Sérielle 16 bits	140	67	219,44	6,20	0,092
90	[74]	Tours	28	58	218,10	31,00	0,534
Artix-7 (Série 7)							
91	[74]	Sérielle 8 bits	252	46	225,53	3,44	0,075
92	[74]	Sérielle 16 bits	140	52	219,44	6,20	0,119
93	[74]	Tours	28	52	218,10	31,00	0,596
94	[61]	Tours	28	112	334,259	31,00	0,277

Tableau 2-6 - État de l'art des implémentations de GIFT-128-128 sur FPGA

N°	Papiers	Implémentation	Cycles	Surface (*Slices*)	Fréquence Max (MHz)	Normalisation @13,56 MHz	
						Débit (Mbps)	Efficacité (Mbps/*Slices*)
Spartan-6							
95	[74]	Sérielle 8 bits	680	81	144,15	2,55	0,031
96	[74]	Sérielle 32 bits	200	66	217,63	8,68	0,131
97	[74]	Tours	40	93	189,93	43,39	0,466
Artix-7 (Série 7)							
98	[74]	Sérielle 8 bits	680	66	144,15	2,55	0,039
99	[74]	Sérielle 32 bits	200	82	217,63	8,68	0,106
100	[74]	Tours	40	76	189,93	43,39	0,571

Comme le souligne cet état de l'art, de nombreux travaux ont été réalisés sur l'AES et PRESENT dans le but d'optimiser leur taille d'implémentation en utilisant différentes architectures comme celle à tours ou sérielle. Les études [40], [52], [54] et [65] proposent de ne pas implémenter le générateur de sous-clé, en supposant que ces dernières soient préchargées dans un bloc mémoire (*BRAM*). Cette solution est souvent privilégiée lorsque le système doit embarquer à la fois la partie chiffrement et déchiffrement, permettant de générer qu'une seule fois les différentes sous-clés. Commune à de nombreux algorithmes de chiffrement, la couche de substitution (*SBOX*) fait également l'objet d'une intensive attention. Les auteurs de [53] se basent sur le théorème des corps commutatifs (*Subfield Arithmetic*), permettant de réduire le nombre d'éléments logiques nécessaires. De même [66] et [73] s'orientent sur des implémentations booléennes des *SBOX* avec une optimisation suivant l'outil Espresso [75] pour le premier, et l'utilisation de table de Karnaugh pour le second. Enfin l'étude [72] propose d'implémenter ces dernières directement dans les blocs mémoires (*BRAM*), atteignant la plus petite taille en termes de *slices* utilisés (sur même cible FPGA), mais au détriment d'un grand nombre de cycles additionnels.

L'état de l'art ci-dessus confirme que l'AES-128 a une taille d'implémentation plus importante que les versions de PRESENT et GIFT. À titre d'exemple, si l'on compare les plus importantes tailles d'implémentations tous types confondus des différents algorithmes sur une même cible FPGA *Artix-7*, l'utilisation de la cryptographie légère permet de réduire la taille jusqu'à un facteur 6 (AES-128 n°33, PRESENT-80 n°57, PRESENT-128 n°85, GIFT-64-128 n°94 et GIFT-128-128 n°99). De même, en comparant cette fois les meilleurs débits de chaque algorithme, normalisé à 13,56 MHz, toutes plateformes confondues, celui de l'AES (173,57 Mbps) est environ 6 fois plus important que ceux des deux versions de PRESENT (28 Mbps) ainsi que de GIFT-64-128 (31 Mbps). Cet écart est

intrinsèquement lié à la structure des algorithmes, à savoir la taille des données à chiffrer et le nombre de tours, comme l'illustre l'équation (2-1). En effet, l'AES-128 n'effectue que 10 tours, pour une donnée de 128 bits alors que PRESENT et GIFT-64-128 en effectuent 31 et 28 respectivement pour chiffrer 64 bits. En revanche, tout comme l'AES-128, GIFT-128-128 utilise la même taille de donnée, mais son nombre de tours est quadruplé, divisant son débit par 4 par rapport à l'AES-128, soit 43,39 Mbps). Enfin, en se focalisant sur les efficacités maximales (rapport débit/surface) de chaque algorithme, GIFT-64-128 (n°93) constitue la meilleure version, atteignant une efficacité supérieure à celle de l'AES-128 (n°23) d'un facteur 1,23. Vient ensuite GIFT-128-128 (n°100) qui malgré un nombre de tours important, se montre 1,18 fois plus efficace que l'AES-128 (n°23), notamment grâce aux 128 bits de données qu'il peut traiter. Les versions de PRESENT (n°51 et n°82) sont pénalisées par le nombre de tours constant et relativement élevé (31 tours). Mais la principale raison de son efficacité réduite provient des 64 bits de données qu'il peut chiffrer. Toutefois, les faibles tailles d'implémentations de PRESENT-80 et PRESENT-128 leur permettent une augmentation de l'efficacité de 2% et 3% respectivement, par rapport à l'AES-128.

L'analyse précédente de l'état de l'art souligne de manière générale, l'intérêt de la cryptographie légère en termes de taille et d'efficacité au détriment du débit. Cependant, il convient aussi de comparer les performances des différentes implémentations (à tours et sérielles). À noter que l'utilisation de blocs mémoires (*BRAM*) rend une fois encore une comparaison directe impossible puisqu'ils constituent des ressources différentes aux *slices*. Concernant l'AES-128, on constate une nette différence entre l'implémentation à tour et sérielle, pouvant atteindre un facteur 10 en termes de *slices* utilisés sur les anciennes séries de FPGA telles que les *Spartan-III*. Sur les plus récents, les *séries 7*, cette différence est réduite entre un facteur 3 et 6. Ces écarts entre les types d'implémentations, mais aussi entre les familles de FPGA s'expliquent par la complexité des *LUT* et la composition des *slices* (*cf.* Tableau 2-1). Ainsi, l'implémentation d'une opération sur une famille *Spartan-III* (2 *LUT*-4 entrées et 2 FF par *slice*) nécessitera plus de *LUT* et par conséquent de *slice*, que sur une *série 7* (4 *LUT*-6 entrées et 8 FF par *slice*).

L'écart de taille entre PRESENT et GIFT oscille en moyenne autour d'un coefficient 2, avec un maximum de 4 (PRESENT-128 sur *Artix-7*). De plus, certaines implémentations sérielles requièrent plus de surface que l'approche à tours. Comme expliqué dans la partie 2.2.4.b), cela provient de l'ajout d'éléments logiques nécessaires à la gestion des données pendant le chiffrement, ce qui réduit considérablement l'intérêt d'une telle approche. D'un point de vue du débit, normalisé à 13,56 MHz [49], l'impact de l'approche sérielle est proportionnel à la réduction du bus de données. La diminution de ce dernier augmente le nombre de tours requis pour effectuer le chiffrement, dégradant ainsi le débit, comme le souligne l'équation (2-1). Pour quantifier cette dégradation, il convient d'extraire des tableaux précédents les meilleurs débits normalisés à 13,56 MHz atteints par

l'implémentation à tours et sérielle de chaque algorithme (toutes familles de FPGA confondues). Après analyse, l'implémentation sérielle de PRESENT-80 affiche un débit diminué d'un facteur 4 par rapport à celui obtenu dans son approche à tours. Il en est de même pour les autres algorithmes, dont la dégradation atteint un facteur 9 pour GIFT-64-128, 17 pour GIFT-128-128, et jusqu'à 30 et 40 pour l'AES-128 et PRESENT-128 respectivement. La conséquence directe à cela porte sur l'efficacité des implémentations. PRESENT-128 montre l'écart le plus important en étant près de 21 fois plus efficace avec une approche à tours. Il en est de même pour PRESENT-80 et GIFT-128-128 avec une efficacité environ 18 fois supérieure par rapport à leurs implémentations sérielles. On note également un gain de 13 pour l'AES-128 et 8 pour GIFT-64-128.

2.2.6. Synthèse

La cryptographie permet d'assurer la sécurité d'un système en termes de confidentialité, d'intégrité et d'authenticité. Défini comme le standard de la cryptographie symétrique par blocs, l'AES est encore aujourd'hui l'algorithme de chiffrement le plus utilisé. Toutefois, il n'est pas conçu pour les architectures fortement contraintes tels que les objets connectés. Cela a conduit la communauté scientifique à mettre au point de nouveaux algorithmes, plus légers, capables d'assurer la sécurité tout en limitant l'utilisation de ressources (puissance de calcul, énergie, surface silicium ...). Parmi eux, PRESENT [14] et GIFT [15]. Ces derniers ont la même structure que l'AES, mais ont une approche orientée sur la manipulation des bits. Le Tableau 2-7 reprend les différentes caractéristiques de ces trois algorithmes.

Tableau 2-7 - Caractéristiques de l'AES, PRESENT et GIFT

Algorithme	Clé (bits)	Donnée (bits)	Nombre de tours
AES-128	128	128	10
PRESENT-80	80	64	31
PRESENT-128	128	64	31
GIFT-64-128	128	64	28
GIFT-128-128	128	128	40

Le choix d'une fonction cryptographique repose d'une part sur ses propriétés intrinsèques, mais également sur son implémentation. En effet, suivant les contraintes prioritaires du système (taille, latence, fréquence, consommation, débit) un algorithme de chiffrement, bien que performant sur le plan sécurité, peut s'avérer être totalement inefficace d'un point de vue fonctionnel et énergétique. Pour cela, il convient d'analyser les possibilités d'implémentations. La première, logicielle, est souvent la plus simple à mettre en œuvre. Cependant, elle augmente l'utilisation de la mémoire programme, souvent très limitée dans les objets connectés et monopolise le microprocesseur. Pour pallier cela, une

solution consiste à déporter le calcul cryptographique sur un système à part, appelé accélérateur matériel. Comme son nom l'indique, l'implémentation des fonctions cryptographique est entièrement matérielle, et peut être déclinée suivant plusieurs approches : une architecture parallèle, à pipelines, à tours ou sérielle. Chacune d'elles permet une optimisation soit de la taille, de la latence, de la fréquence ou encore du débit, comme résumé par le Tableau 2-8.

Tableau 2-8 - Caractéristiques des différentes implémentations

Implémentations	Taille	Latence (cycles)	Fréquence	Débit	Consommation
Parallèle	Élevée	Minimale	Minimale	Maximal	Élevée
Pipelines	Maximale	Minimale	Élevée	Maximal	Maximale
Tours	Faible	*cf.* algorithme	Maximale	Élevé	Faible
Sérielle	Minimale	Maximale	Maximale	Minimal	Minimale

Ainsi, dans le contexte des objets connectés, où de nombreux appareils sont fortement contraints en termes de taille, de puissance de calcul et d'énergie, l'implémentation matérielle constitue la solution idéale. L'état de l'art de la section précédente met en avant l'efficacité bien supérieure de l'approche à tours comparée à l'implémentation sérielle et qui plus est, requiert parfois plus de surface, notamment sur les algorithmes légers. Parmi ces derniers, PRESENT (standard ISO-IEC 29192-2 [28]) et GIFT suivent tous deux le même modèle que le standard AES. De plus, ils sont de par leur conception, optimisés pour les implémentations matérielles, ce qui fait d'eux d'excellents candidats pour sécuriser les objets connectés. En raison de sa récente publication (2017), GIFT n'a pas encore fait l'objet de nombreuses recherches. Lors des premiers résultats de cette thèse, aucune étude n'avait encore mis en avant les performances de GIFT sur cible FPGA. Depuis, seulement deux travaux [61] et [74] sont à ce jour disponibles. [61] propose des implémentations robustes face aux injections de fautes. Cependant grâce à la mise en ligne de leurs codes sources, il est possible de reproduire leur implémentation sans protection afin qu'elle puisse être comparée avec les autres travaux. Bien que remarquable, [74] oppose GIFT et PRESENT sur une même cible matérielle, mais ne réalise aucune comparaison sur les gains obtenus par rapport à la cryptographie classique telle que l'AES. De plus, l'aspect sécuritaire n'est pas traité. Enfin, pour être complet, il convient de noter que l'étude [76] propose plusieurs variantes d'implémentations à seuil afin de protéger GIFT-128-128 contre les attaques par canaux auxiliaires. Bien que l'évaluation de leurs contremesures soit réalisée sur FPGA, aucune information n'est donnée quant aux performances et à la taille d'implémentation. Par conséquent, les travaux de cette thèse s'inscrivent parmi les premiers à comparer de manière fiable les implémentations matérielles de GIFT, PRESENT et l'AES sur FPGA, mais aussi d'évaluer leur sécurité face aux attaques par canaux auxiliaires.

2.3. Évaluations d'implémentations matérielles de l'AES, PRESENT et GIFT sur FPGA

Le premier objectif de ce chapitre est de comparer de manière fiable, grâce à des métriques communes, les implémentations matérielles d'algorithmes de cryptographies légers afin d'en souligner les gains obtenus par rapport à la cryptographie classique. De nombreux travaux portent sur l'implémentation de ces algorithmes, mais comme vu précédemment les comparaisons sont souvent difficiles à réaliser (différences des technologies employées, multiples configurations, etc.). L'avantage principal de ce travail réside dans l'utilisation de métriques communes, tout au long de l'évaluation. Cette étude peut également guider tout concepteur de systèmes embarqués dans le choix d'un algorithme de chiffrement adapté aux contraintes d'implémentation abordées dans cette section, mais également du point de vue sécuritaire, développée dans la suivante (*cf.* 2.4).

2.3.1. Architectures et implémentations

De par leur nature, les objets connectés sont souvent utilisés pour capter l'information et la transmettre sur le réseau, de manière sécurisée (*cf.* confidentialité, intégrité, authenticité). Cette approche justifie l'intérêt d'embarquer uniquement la fonction de chiffrement d'un algorithme cryptographique, permettant de réduire la taille d'implémentation. Le déchiffrement n'est en effet utile que lorsqu'une communication à double sens doit être sécurisée. Comme détaillé dans la partie 2.2.4.b), chaque type d'implémentation matérielle a son lot d'avantages et d'inconvénients. Dans le contexte de l'IdO, constitué d'objets fortement contraints en termes de taille et de consommation énergétique, les solutions compactes sont donc privilégiées. Parmi elles, l'implémentation à tours s'avère être la plus efficace comme le montre l'état de l'art. Par conséquent, les travaux de cette thèse se concentrent sur cette dernière, avec pour objets d'étude l'AES-128 (mode *ECB*), PRESENT-80, PRESENT-128, GIFT-64-128 et GIFT-128-128, et plus particulièrement sur leur fonction de chiffrement.

Afin d'obtenir une évaluation des plus fiables, l'implémentation des algorithmes se doit d'être similaire pour chacun d'eux, dans la mesure du possible. La Figure 2-28 illustre l'architecture proposée. Celle-ci peut être découpée en deux parties distinctes : le cœur cryptographique (en rouge) qui, comme son nom l'indique, correspond à l'algorithme de chiffrement, et le contrôle de ce dernier (en gris). Cette séparation entre les éléments de contrôle et la partie cryptographique offre une évaluation précise des ressources demandées par cette dernière. Son fonctionnement est le suivant : l'utilisateur transmet le message clair et la clé par le bloc *UART* (*Universal Asynchronous Receiver Transmitter*), contrôlé par la machine d'états (bloc *FSM*). Une fois la réception complète, le bloc *FSM* transmet les informations au cœur cryptographique et déclenche le chiffrement (via le signal

Initialisation). Une fois ce dernier terminé, le cœur cryptographique informe la machine d'états (*Fin chiffrement*), qui récupère alors le message chiffré. Celui-ci est finalement réceptionné par l'utilisateur via le bloc *UART*. Le chiffrement peut être à nouveau effectué après réactivation du signal *Initialisation*. De ce fait, le cœur cryptographique proposé est conçu pour une intégration des plus simples (« *Plug and Play* »), qui se réduit au simple contrôle du signal *Initialisation*. Le cœur s'arrête automatiquement lorsque le chiffrement est terminé et conserve le résultat jusqu'au prochain démarrage. À noter que le cadencement du cœur cryptographique (*Horloge cryptographique*) est contrôlé grâce au bloc *MMCM* (*Mixed-Mode Clock Manager*) de Xilinx. Cela permet d'une part une meilleure gestion de l'arbre d'horloge, d'activer/désactiver le cœur uniquement, mais surtout de fixer indépendamment du reste du circuit, la fréquence de fonctionnement de la partie chiffrement. De ce fait, il est possible d'obtenir une évaluation fiable des performances du cœur cryptographique à différentes fréquences, sans être impacté par le reste du circuit. Cependant, en raison de son incapacité à générer de faibles fréquences (<4,688MHz), un diviseur (bloc *Div*), basé sur un compteur programmable, contrôle l'activation de l'horloge de sortie (*Clock Enable*) du MMCM, produisant une division de sa fréquence. Toutefois, lorsque le cadencement de la partie chiffrement et le reste du circuit est différent, de la logique de contrôle est nécessaire pour assurer la synchronisation entre les deux domaines d'horloges, notamment pour le pilotage de l'initialisation du cœur. Une méthode simple consiste à inclure un registre à décalage à deux étages, chacun d'eux fonctionnant sur une des horloges. Enfin, le bloc *UART* est configuré pour une vitesse de transmission de 9 600 bauds, avec 8 bits de données, sans parité, et avec 1 bit de stop (9600-8-N-1). Ainsi, à l'exception du message clair/chiffré et de la clé, aucune modification de l'architecture n'est nécessaire. De plus, elle constitue une possible mise en œuvre d'un système cryptographique complet, permettant de la tester en conditions réelles.

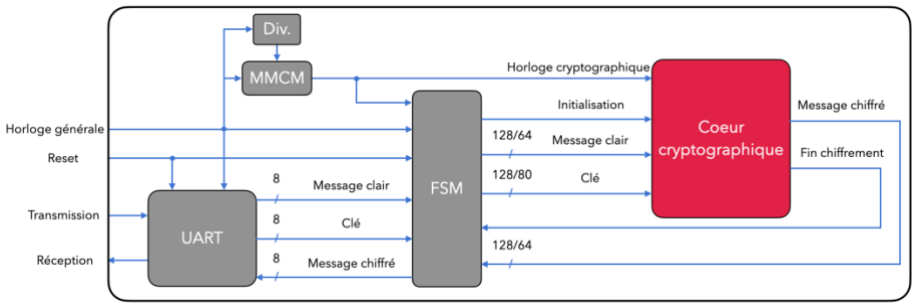


Figure 2-28 - Architecture complète proposée

Le reste de cette section se focalise sur la partie cryptographique. Toujours dans un but de maximiser la fiabilité des comparaisons, les architectures d'implémentation de chaque algorithme suivent la même structure, composée de trois sous-ensembles : *Tour*, *Génération Sous-Clés* et *Génération Compteur/Round Constant*. À noter que contrairement au message clair, à la clé et au

message chiffré, le *Compteur/Round Constant* ne figure pas parmi les entrées/sorties du bloc cryptographique. Étant défini par l'algorithme, il est fixé à l'intérieur du bloc. Les Figure 2-29 à Figure 2-33 illustrent les implémentations proposées pour les cinq algorithmes.

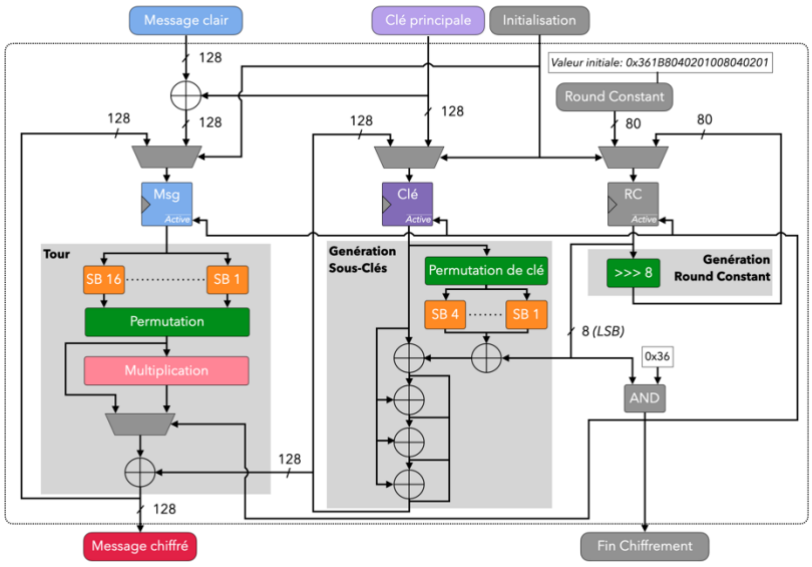


Figure 2-29 - Architecture de l'AES-128

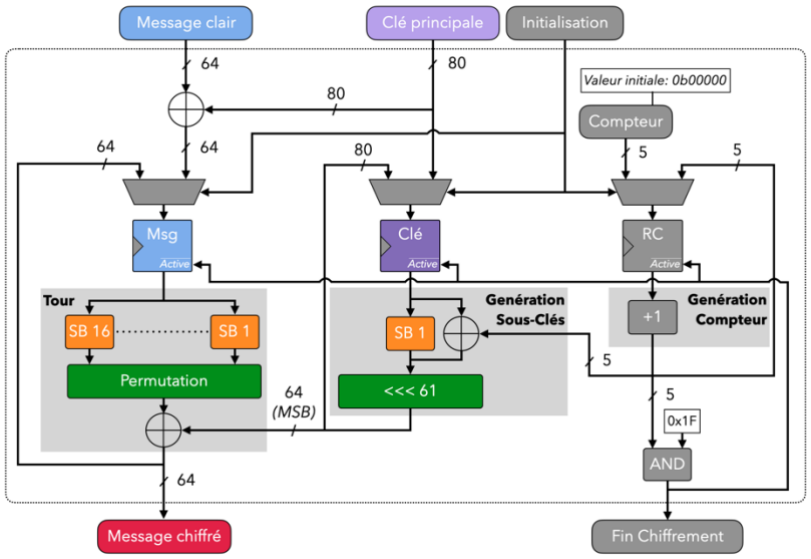


Figure 2-30 - Architecture de PRESENT-80

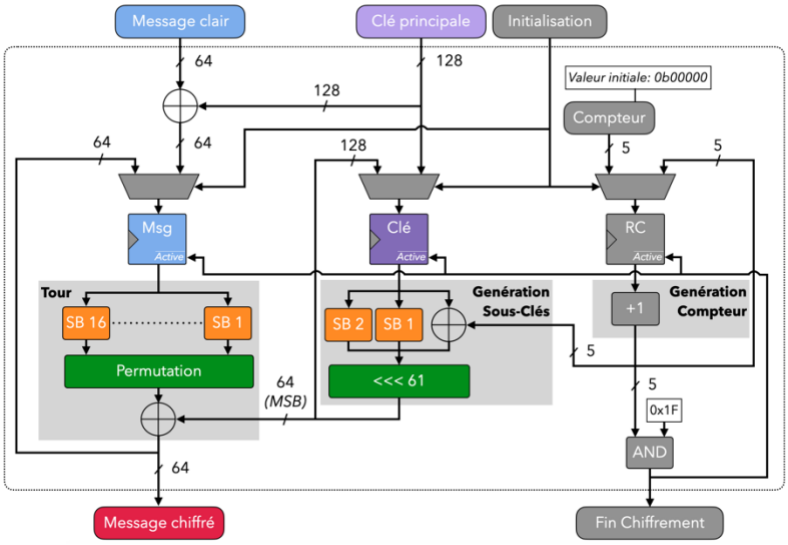


Figure 2-31 - Architecture de PRESENT-128

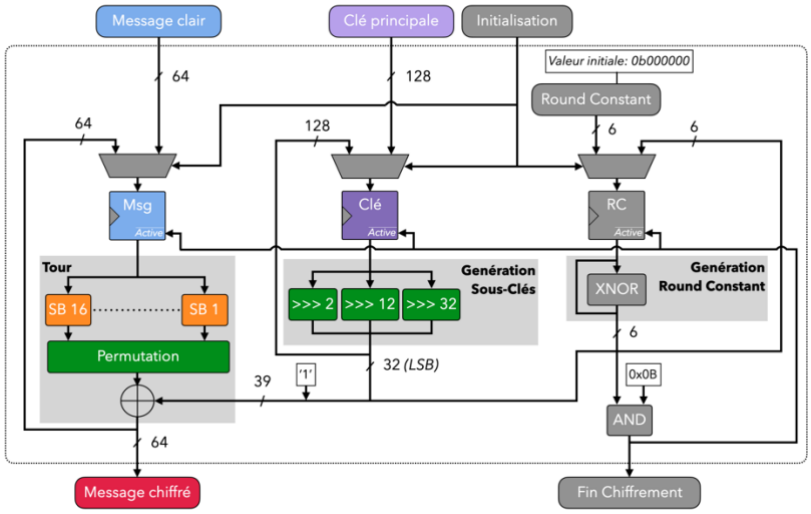


Figure 2-32 - Architecture de GIFT-64-128

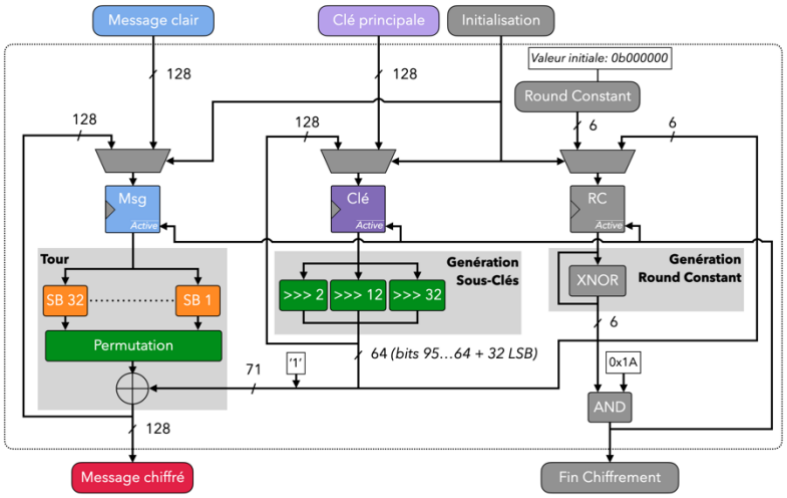


Figure 2-33 - Architecture de GIFT-128-128

Toute l'architecture est implémentée de manière combinatoire, à l'exception des registres *Msg*, *Clé* et *RC* ou *Cpt*. On retrouve les éléments détaillés dans les sections 2.2.1, 2.2.2 et 2.2.3, tels que l'opérateur *XOR* (⊕), la couche de substitution (*SBOX*, en orange) et de permutation (en vert). Comme le souligne l'état de l'art, plusieurs méthodes d'implémentations sont possibles pour les *SBOX*. L'approche sélectionnée ici est la plus classique, en utilisant des *LUT*, en tant que mémoires à lecture seule (*ROM*). À l'inverse, la couche de permutation repose uniquement sur des croisements de câblage, ne nécessitant aucune ressource matérielle, hormis les éléments de routage.

Au démarrage, certaines implémentations existantes attendent la fin du chargement du message clair, de la clé et de la valeur initiale du *Compteur/Round Constant*, avant de commencer le chiffrement, ce qui requiert un cycle d'horloge supplémentaire dans le processus global. De plus, cela entraine un ajout d'éléments logiques afin de distinguer les différentes phases de fonctionnement du cœur cryptographique, à savoir le chargement du message clair/clé, le premier tour de chiffrement et les tours intermédiaires jusqu'au résultat final. Dans l'architecture proposée, le signal d'initialisation permet pour chaque sous-ensemble (*Tour*, *Génération Sous-Clés* et *Génération Compteur/Round Constant*) de sélectionner le flux de données à appliquer aux opérateurs. Pendant le chargement de celles-ci, le signal d'initialisation, associé aux multiplexeurs, sélectionne les flux provenant du message clair, de la clé et du *Compteur/Round Constant*. Ces derniers sont alors stockés dans leurs registres respectifs, nommés *Msg*, *Clé* et *RC* ou *Cpt* et sont propagés dans leur sous-ensemble. À noter que ces registres sont cadencés par un signal d'horloge en entrée du cœur cryptographique (*Horloge cryptographique*), non représentée ici afin d'alléger les schémas. Pendant toute la durée du chargement, la machine d'états (bloc *FSM* de l'architecture complète) maintient le signal d'initialisation, ce qui force le cœur cryptographique à conserver la lecture des flux d'entrées. Cela

signifie que seul le premier tour de l'algorithme de chiffrement est actif (aucun rebouclage n'est encore pris en compte). Une fois le chargement terminé, le signal d'initialisation ordonne aux multiplexeurs de sélectionner les flux provenant des boucles de chaque sous-ensemble. Les données de ces dernières sont en réalité le résultat du premier tour de chiffrement, effectué pendant le chargement. Par conséquent, ce système permet de s'affranchir du délai supplémentaire, présent en début de tout nouveau chiffrement, mais aussi de réduire le nombre d'éléments logiques nécessaires. En revanche, lorsque le transfert du message clair et de la clé est fragmenté en plusieurs sous-parties, chacune d'elles provoque l'exécution du premier tour de l'algorithme de chiffrement. C'est le cas de l'*UART*, où chaque transmission contient uniquement 8 bits de données. Cela signifie que pour transmettre 128 bits au total, seize envois sont nécessaires et autant d'exécution du premier tour du chiffrement. Dans un tel scénario, les bénéfices de l'approche proposée sur le cœur cryptographique, à savoir la réduction de la latence globale et de la taille d'implémentation, ne semblent pas appropriés d'un point de vue énergétique. Toutefois, il n'est pas sans rappeler que l'objectif de cette thèse porte sur l'évaluation des algorithmes de chiffrement et non sur un système complet. Lors d'une utilisation réelle, d'autres architectures/protocoles que l'*UART* peuvent être mises en œuvre, n'étant pas impactées par le problème énergétique soulevé. Quand bien même, et ce au détriment du précédent gain sur la latence, il est possible d'annuler ce comportement transitoire du premier tour, en désactivant l'horloge du cœur cryptographique jusqu'à la fin du chargement du message clair et de la clé. De plus, ce procédé de « *Clock Enable* » constitue une solution souvent employée afin de réduire la consommation des systèmes contraints [77]. Ainsi, les exécutions supplémentaires du premier tour de l'algorithme de chiffrement sont désactivées tout en conservant sa taille d'implémentation la plus minimaliste possible.

Comme le souligne la description précédente, le message clair et la clé sont chargés dans le cœur cryptographique, puis les résultats de chaque tour intermédiaire proviennent du rebouclage des sous-ensembles et sont mis à jour dans les différents registres. Il convient alors d'avoir un système permettant d'évaluer l'avancement du chiffrement afin de déterminer le nombre de tours restant. Une première méthode consisterait à ajouter un simple compteur, en guise de machine d'états. Toutefois, en analysant les algorithmes, il s'avère que ce système est déjà présent. En effet, le *Compteur/Round Constant* possède une valeur unique pour chaque tour. Par conséquent, il peut être utilisé afin de déterminer la fin du chiffrement (*i.e.* le dernier tour). L'architecture proposée utilise cette particularité. L'avantage de celle-ci est qu'elle permet l'implémentation de ce contrôle de tour avec un seul opérateur logique *AND* (implémenté dans une *LUT*), ce qui représente un cout matériel supplémentaire infime comparé à la première méthode. Dans le cas de PRESENT, son *compteur* est défini sur 5 bits, représentant les tours 1 à 31 (« 00001 » à « 11111 » au format binaire). Le déclenchement du signal de fin de chiffrement est déclenché grâce à un opérateur logique *AND*

sur 5 bits, en comparant la valeur du *compteur* avec celle du dernier tour, soit 31. GIFT suit cette même approche, mais avec un *AND* 6 bits en raison de son *Round Constant*. Les valeurs de détections de fin de chiffrement sont 0x0B et 0x1A pour GIFT-64-128 et GIFT-128-128 respectivement. Enfin, il en est de même pour l'AES-128, avec un AND 8 bits vérifiant la valeur du dernier tour 0x36. Toutefois, il convient de noter que cette vérification s'effectue directement à la sortie du registre *RC*, contrairement à PRESENT et GIFT. En effet, l'implémentation du *Round Constant* et du sous-ensemble *Génération Round Constant* suit une stratégie particulière. L'approche adoptée ici est de concaténer chacune des valeurs (sur 8 bits) du *Round Constant* au cours des 10 tours dans un seul registre de 80 bits (10 x 8 bits). De cette manière, les 8 bits de poids faibles correspondent au *Round Constant* du tour en cours. Un simple décalage vers la droite suffit à mettre à jour le registre pour le prochain tour, nécessitant aucun opérateur logique. Cette stratégie réduit ainsi le cout matériel en privilégiant l'usage des *FF*, plus abondantes dans les *slices* que les *LUT*.

D'après les explications précédentes, le temps de chiffrement requis par les architectures proposées correspond au nombre de tours défini par l'algorithme. Toutefois, grâce à l'exécution du premier tour pendant la phase de chargement du message clair et de la clé, la latence globale (*i.e.* temps entre le chargement et la récupération du message chiffré) est réduite d'un cycle. Au-delà de ces caractéristiques architecturales, la suite de ce paragraphe se concentre sur l'implémentation en elle-même. Comme vu précédemment, les comparaisons matérielles dépendent de nombreux facteurs. Afin de fournir des résultats des plus fiables possibles, principalement sur les implémentations de PRESENT et GIFT, la cible choisie fait partie de la même série que celle utilisée dans l'état de l'art, à savoir le FPGA *Artix-7 XC7A100T-1CSG324C*. Notamment présent sur la carte d'évaluation *Nexys-4* de Digilent [78], les *Artix* correspondent à l'entrée de gamme de la série 7 de Xilinx, ce qui fait d'eux d'excellents candidats pour le contexte de l'IdO. Comme défini précédemment dans le Tableau 2-1, leurs *slices* sont principalement constitués de quatre *LUT* de six entrées et de huit FF. De plus, certains *slices* peuvent être configurés en tant que mémoires dites « distribuées » (*LUTRAM*) ou en registres à décalages de 16 ou 32 bits (*SRL16/SRL32, Shift Register Look-Up Table*) [79]. Puisque ces différentes configurations ont un impact sur la taille et la performance, plusieurs implémentations des algorithmes de chiffrement sont proposées, afin de sélectionner la plus efficace. Le Tableau 2-9 décrit les ressources utilisées pour chacune d'elles.

Tableau 2-9 - Configurations des slices

Nom de l'implémentation	Multiplexeurs d'entrées	Stockage des données	Utilisation de primitives Xilinx
« Portable »	*LUT*	*FF*	Non
« FF »	*MUX* des *slices*	*FF*	Oui
« SRL »	*MUX* des *slices*	*LUT*	Oui

En premier lieu, l'implémentation « Portable » correspond à une utilisation classique des *LUT* et les *FF*, sans configuration spécifique. La version « FF » tire profit des multiplexeurs (*F7*) présents au nombre de deux, dans les *slices* pour gérer le flux de données (entrées/rebouclage). Enfin la configuration « SRL » et une extension de « FF » à la différence qu'ici le stockage des données est effectué dans les *LUT*, grâce à la primitive *SRL*. La Figure 2-34 illustre les éléments des *slices* utilisés pour les différentes configurations. Contrairement à ses homologues, la version « Portable » n'utilise aucune primitive propriétaire, la rendant compatible avec n'importe quel type/fabricant de FPGA. Il convient de préciser que toutes ces implémentations utilisent les mêmes sous-ensembles *Tour*, *Génération Sous-Clés* et *Génération Compteur/Round Constant*. Seule la sélection des flux d'entrées (multiplexeurs) et le stockage des données diffèrent.

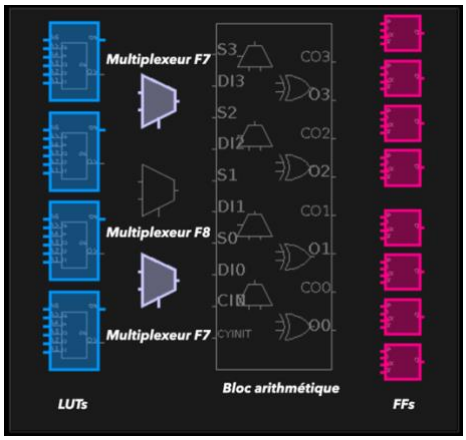


Figure 2-34 - Composition d'un slice de série 7

Toutes les implémentations sont codées en VHDL, synthétisées et implémentées sous *Vivado Design Suite 2016.2* [42] pour le FPGA *Artix-7 XC7A100T-1CSG324C* de Xilinx. L'option « *AreaOptimized_High* » est sélectionnée afin de réduire autant que possible la taille d'implémentation. Étant donnée la nature sensible d'un algorithme de chiffrement, aucune modification/optimisation logique ne doit être faite durant l'étape de synthèse. Afin de conserver l'architecture telle qu'elle a été décrite, sous peine d'inclure de potentielles failles de sécurités, le paramètre « *Flatten_Hierarchy* » [42] est désactivé, et l'attribut VHDL « *don't touch* » [42] est

inclus dans les fichiers sources sur chaque bloc et signal de l'architecture. Deux blocs de placement/routage sont créés dans le FPGA : un dédié uniquement au cœur cryptographique (nommé *pblock_CRYPTO)*, l'autre pour le reste du système à savoir l'*UART* et la *FSM* (nommé *pblock_CTRL)*. Ces blocs permettent d'évaluer précisément les ressources engagées par chaque partie du système. Leurs tailles sont fixes et dimensionnées de sorte qu'ils puissent contenir toutes les versions des implémentations, de la plus petite à la plus grande. Quant à leurs positions, comme illustrées sur la Figure 2-35, ils sont positionnés au plus proche afin d'avoir une implémentation générale la plus compacte possible, mais aussi pour évaluer les fréquences maximales atteignables. Tous les résultats d'implémentations sont donnés après placement-routage. L'ensemble des codes sources et des fichiers de contraintes est disponible en ligne (*https://gite.lirmm.fr/ldalmasso/LightweightCryptoCiphers.git*).

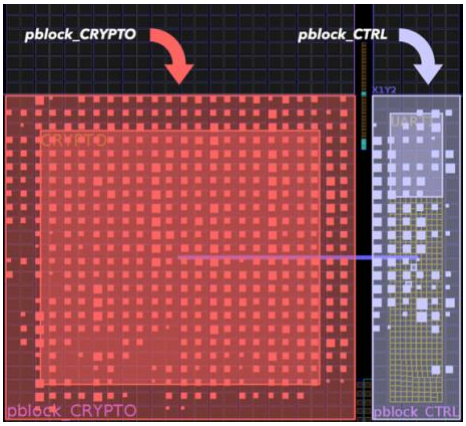


Figure 2-35 - Placement des *pblocks*

Par ailleurs, les quelques travaux de l'état de l'art ayant mis à disposition leurs codes sources ([61], [62], [70] et [74]), s'avèrent être finalement les plus pertinents à étudier. Toujours dans le but d'optimiser la fiabilité des comparaisons, ces derniers sont resynthétisés et re-implémentés dans les mêmes conditions de ce travail : implémentation à tours, incorporation dans l'architecture complète (*cf.* Figure 2-28), cible *Artix-7 XC7A100T-1CSG324C*, paramètres « *AreaOptimized_High* » et « *Flatten_Hierarchy* » de *Vivado*, attribut « *don't touch* » et placement/routage dans les *pblocks* correspondants. L'adaptation des codes sources a uniquement pour but la compatibilité avec l'architecture complète proposée dans cette thèse, en minimisant les modifications des originaux autant que possible.

2.3.2. Performances à fréquence maximale

Cette section a pour but de comparer les implémentations proposées avec celles de l'état de l'art, majoritairement évaluées à leur fréquence maximale. Cette dernière est déterminée empiriquement

en l'augmentant jusqu'à ce que le *WNS* (*Worst Negative Slack*) du cœur cryptographique ne soit plus satisfait (<0). Afin de s'assurer de leur fonctionnement, toutes les implémentations sont testées sur la carte FPGA *Nexys-4* [78]. Les résultats en termes de taille, débit et efficacité sont détaillés dans le Tableau 2-10.

Tableau 2-10 - Résultats des implémentations à tours à fréquence maximale sur FPGA Artix-7

Papiers	Cycles	Surface			Fréquence Max (MHz)	Débit (Mbps)	Efficacité (Mbps/*Slices*)
		LUT	*FF*	*Slices*			
AES-128							
[61]	10	2049	264	530[1]	141,418	1810,15	3,415
[62]	10	2010	264	522	149,502	1913,63	3,666
« Portable »	10	1485	336	**412**	165,781	2122	**5,150**
« FF »	10	1560	336	473[2]	**174,194**	**2229,68**	4,714
« SRL »	10	1716	0	506[2]	130,435	1669,56	3,3
PRESENT-80							
[70]	32	275	214	83	**303,361**	606,72	7,310
« Portable »	31	185	149	**53**	302,000	**623,48**	**11,764**
« FF »	31	177	149	88[3]	250,781	517,74	5,883
« SRL »	31	275	0	116[3]	196,032	404,71	3,489
PRESENT-128							
[61]	31	598	198	183	**324,828**	**670,61**	3,664
[70]	32	300	262	93	316,304	632,61	6,802
[74]	31	429	202	115	235,676	486,56	4,231
« Portable »	31	228	197	**73**	285,507	589,43	**8,074**
« FF »	31	181	197	117[4]	262,000	540,90	4,623
« SRL »	31	294	0	139[4]	191,919	396,22	2,850
GIFT-64-128							
[61]	28	389	198	112	**334,259**	**764,02**	6,822
[74]	28	1334	205	388	173,519	396,615	1,022
« Portable »	28	193	198	**59**	319,318	729,87	**12,371**
« FF »	28	94	198	119[5]	295,000	674,29	5,666
« SRL »	28	203	0	136[5]	209,412	478,66	3,519
GIFT-128-128							
[74]	40	1795	270	477	142,151	454,88	0,954
« Portable »	40	290	262	**84**	**314,394**	**1006,06**	**11,977**
« FF »	40	172	262	153[6]	278,906	892,50	5,833
« SRL »	40	315	0	180[6]	206,286	660,11	3,667

1 : Utilisation de 320 MUXF7 et 160 MUXF8 des slices

2 : Utilisation de 336 *MUXF7* des *slices*

3 : Utilisation de 149 MUXF7 des slices

4 : Utilisation de 197 MUXF7 des slices

5 : Utilisation de 198 MUXF7 des slices

6 : Utilisation de 262 MUXF7 des slices

Tout d'abord, il convient de valider la cohérence des implémentations proposées dans ce travail avec celles de l'état de l'art. D'après le Tableau 2-10, il apparait clairement que les performances en termes de taille, de débit et d'efficacité sont du même ordre de grandeur que celle des autres études. Toutefois, il est important de mettre en avant l'écart des résultats entre ceux présentés dans l'étude [74] (*cf.* Tableau 2-4, Tableau 2-5 et Tableau 2-6) et ceux resynthétisés dans ce travail à partir de leurs codes sources (*cf.* Tableau 2-10). Un écart de presque un facteur deux pour PRESENT-128, jusqu'à plus de sept pour GIFT-64-128 est noté concernant la taille d'implémentation. Comme évoqué précédemment, de nombreux facteurs sont à prendre en considération. En effet, les optimisations du synthétiseur n'ont visiblement pas été désactivées dans les travaux d'origines (*i.e.* attribut « *don't touch* » non utilisé), ce qui permet de réduire les ressources matérielles utilisées, au risque d'introduire des failles de sécurité. De plus, le fichier de contraintes de placement/routage n'étant pas fourni par les auteurs, l'analyse de leurs travaux n'a donc pas pu être complète, ce qui rend difficile la reproductibilité des résultats, notamment pour les performances à fréquences maximales. Concernant ces dernières, la version resynthétisée permet de presque doubler le cadencement de PRESENT-128 par rapport à l'étude d'origine. En revanche, les deux implémentations de GIFT atteignent moins de 80% de la fréquence maximale indiquée par les auteurs. Malgré cela, il est possible d'affirmer la cohérence des résultats obtenus, validés lors des tests réels effectués sur carte FPGA.

Suite aux explications précédentes, ce paragraphe s'efforce de comparer les performances des différents algorithmes, suivant les trois implémentations proposées dans cette thèse, à savoir « Portable », « FF » et « SRL ». Comme attendu, les résultats du Tableau 2-10 montrent que l'AES atteint le meilleur débit jusqu'à près d'un facteur 4 en moyenne. Cela s'explique notamment par la taille de son bus de données de 128 bits, deux fois supérieure à celle de PRESENT-80/128 et GIFT-64-128, mais aussi grâce à son faible nombre de tours, limité à 10. Bien que GIFT-128-128 utilise les mêmes tailles de données et clé, il est fortement pénalisé par ses 40 tours. Toutefois, la taille de l'AES est en moyenne quatre fois plus importante que ses homologues, ce qui induit indéniablement une plus forte consommation énergétique. Cela confirme que l'AES n'est pas une solution adaptée pour les objets fortement contraints, soulignant l'intérêt de la cryptographie légère. Parmi elles, le standard PRESENT-80 offre la plus petite taille d'implémentation de cette étude. Cependant, cette réduction n'est possible que par un compromis sur le niveau de sécurité, avec une clé de seulement 80 bits. Le passage à la version plus sécurisée, PRESENT-128, implique une augmentation de taille entre 20% et 38% selon l'implémentation et par conséquent une diminution de l'efficacité entre 18% et 31%. Enfin GIFT se positionne comme étant le plus efficace, dans ses deux versions, en raison de sa faible empreinte matérielle et de sa capacité à atteindre les fréquences les plus élevées. Avec seulement 11% supplémentaires de surface, GIFT-64-128 offre un meilleur niveau de sécurité, en termes de longueur de clé, et un gain de 5% sur l'efficacité par rapport à PRESENT-80. De même, il permet une

réduction de taille jusqu'à 19% et un gain maximal sur l'efficacité de 53% par rapport à PRESENT-128, qui utilise tout comme GIFT, une clé de 128 bits. Cette diminution des ressources matérielles provient principalement du nombre de *LUT* utilisées pour l'opérateur *XOR* dans le sous-ensemble *Tour*. En effet, dans PRESENT-128 la totalité des 64 bits de données du message à chiffre sont mélangés avec la sous-clé alors que GIFT effectue cette opération uniquement sur 39 bits ('1' sur le bit de poids fort + 32 bits de sous clé + 6 bits du *Round Constant*). Le sous-ensemble *Génération Sous-Clés* présente également une particularité, permettant de réduire la taille d'implémentation de GIFT. Alors que ce dernier n'utilise que des permutations binaires qui ne requièrent aucun élément logique, PRESENT génère ses sous-clés à l'aide d'un opérateur *XOR*, ce qui lui vaut une taille plus importante. Enfin, GIFT-128-128 et l'AES-128 utilisent tous deux des données et une clé de 128 bits, ce qui permet de les comparer. Bien que sa fréquence maximale soit presque deux fois supérieure à celle de l'AES-128, le débit de GIFT-128-128 est fortement impacté par ses 40 tours requis pour effectuer le chiffrement. Toutefois, en raison de sa faible empreinte matérielle, il montre une efficacité de plus d'un facteur 2 par rapport à l'AES-128. Ainsi, au travers de ces différents résultats, GIFT s'avère être un candidat idéal pour sécuriser les objets fortement contraints tout en optimisant l'utilisation des ressources.

Comme détaillé dans la section 2.3.1, plusieurs configurations d'implémentations sont proposées, nommées « Portable », « FF » et « SRL ». Outre les comparaisons précédentes entre les différents algorithmes, les résultats du Tableau 2-10 permettent également d'identifier la meilleure version. Pour l'ensemble des algorithmes, la configuration « Portable » offre la plus petite taille d'implémentation en termes de *slices*. De plus, elle permet également d'atteindre les plus hautes fréquences maximales, à l'exception de l'AES (expliqué ci-après). Comme défini par l'équation (2-2), l'efficacité dépend de la fréquence et de la surface d'implémentation. De ce fait, la version « Portable » constitue de loin la version la plus efficace. Cela est d'autant plus vrai pour la cryptographie légère, où l'écart s'élève à plus d'un facteur 3. Il est toutefois important de noter que la meilleure efficacité n'est pas forcément atteinte pour une fréquence maximale, en raison du placement/routage, qui utilise plus ou moins de *slices* pour garantir le *WNS*. C'est notamment le cas de l'AES, qui atteint un maximum d'environ 174 MHz pour sa version « FF », mais dont l'efficacité n'est pas maximale. Pour comprendre ces différences de performances, il convient de faire plusieurs rappels. Tout d'abord, la seule différence entre les différentes configurations réside dans la méthode de stockage des données et l'implémentation des multiplexeurs d'entrées. Comme détaillé dans la section 2.3.1, un *slice* est constitué de quatre *LUT*, huit *Flip-Flops* et deux multiplexeurs *F7*. Les configurations « Portable » et « FF » utilisent les *Flip-Flops* pour stocker les données, contrairement à la version « SRL » qui utilise une configuration particulière des *LUT*, moins abondantes dans les *slices*. De plus, la configuration « Portable » implémente les multiplexeurs d'entrées via les *LUT*,

alors que ses homologues « FF » et « SRL » utilisent directement ceux présents dans les *slices*, limités au nombre de deux. De ce fait, suivant l'abondance des éléments logiques nécessaires au stockage des données et à la sélection des flux d'entrées, plus ou moins de *slices* sont utilisés, résultant sur une augmentation progressive des tailles des versions « Portable », « FF » et « SRL ». Enfin, la capacité de fonctionnement à haute fréquence dépend principalement des contraintes de placement/routage. Ces dernières sont les plus strictes dans la version « SRL » puisque les *LUT* de seulement certains *slices* ne peuvent être configurées comme élément de mémoire (*Shift Register Look-Up Table*). Cela réduit alors les possibilités d'optimisations de placement, limitant ainsi la fréquence maximale atteignable. C'est pourquoi toutes les versions « SRL » ont un faible niveau de cadencement par rapport à leurs homologues. À l'opposé, « Portable » permet d'atteindre les plus hauts niveaux de fréquence, à l'exception de l'AES. Cet effet est la conséquence directe de la densité de l'implémentation. En effet, plus cette dernière est petite et compacte, meilleures sont ses capacités à monter en fréquence. Les versions « FF » et « SRL » utilisent les deux multiplexeurs *F7* des *slices* pour la sélection des données d'entrée. Or, de par leur nature plus légère, les algorithmes PRESENT et GIFT requièrent moins d'éléments logiques. Cela signifie que certains *slices* sont uniquement utilisés pour leurs multiplexeurs, ce qui augmente la surface de l'implémentation tout en diminuant sa densité. À l'inverse, l'AES nécessite d'importantes ressources matérielles, ce qui lui offre un meilleur taux de remplissage des *slices* (*i.e.* densité) dans sa version « FF ». Ce phénomène est mis en avant par les résultats du Tableau 2-10, avec un écart maximal de fréquence de seulement 43 MHz pour l'AES, alors qu'il oscille autour de 100 MHz pour PRESENT et GIFT. De même, la capacité de la cryptographie légère à fonctionner à de plus hautes fréquences que l'AES réside cette fois non pas dans la densité, mais dans la taille d'implémentation.

2.3.3. Performances en conditions réelles pour l'IdO

a) Performances suivant l'ISO/IEC 14443-2 à 13,56 MHz

Dans la pratique de nombreux systèmes de l'IdO sont cadencés à des fréquences prédéfinies, comme les cartes sans contact à 13,56 MHz, définies par l'ISO/IEC 14443-2 [49]. De ce fait, il est intéressant d'étudier en premier lieu, les performances des différents cœurs cryptographiques suivant cette norme. D'un point de vue implémentation, cette section se concentre sur la meilleure configuration parmi celles proposées dans ce travail, à savoir la version « Portable ». La sortie du bloc *MMCM*, qui délivre l'horloge du cœur cryptographique, est configurée à 13,56 MHz. Les Figure 2-36, Figure 2-37 et Figure 2-38 illustrent respectivement pour chaque algorithme leur taille d'implémentation, leur efficacité et l'énergie/bit consommée. Le Tableau 2-11 détaille les éléments logiques utilisés.

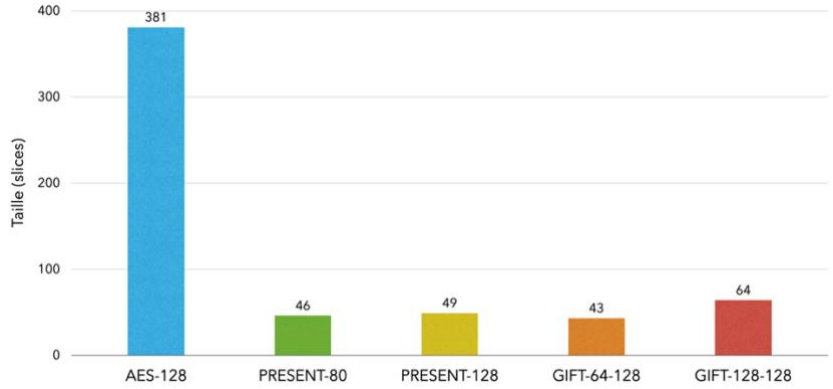


Figure 2-36 - Comparaison des tailles d'implémentations à 13,56 MHz

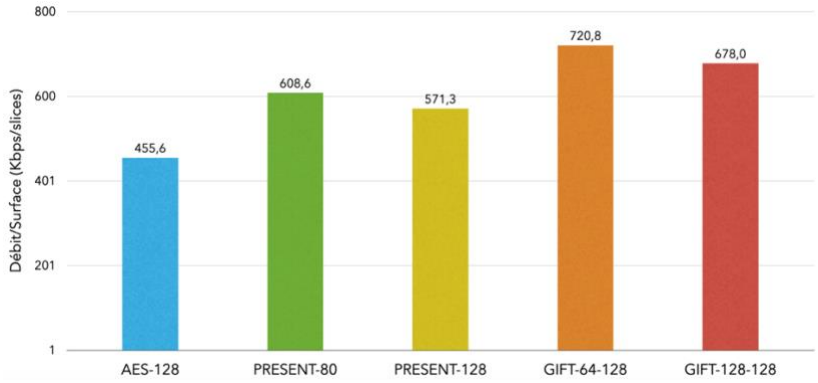


Figure 2-37 - Comparaison de l'efficacité des implémentations à 13,56 MHz

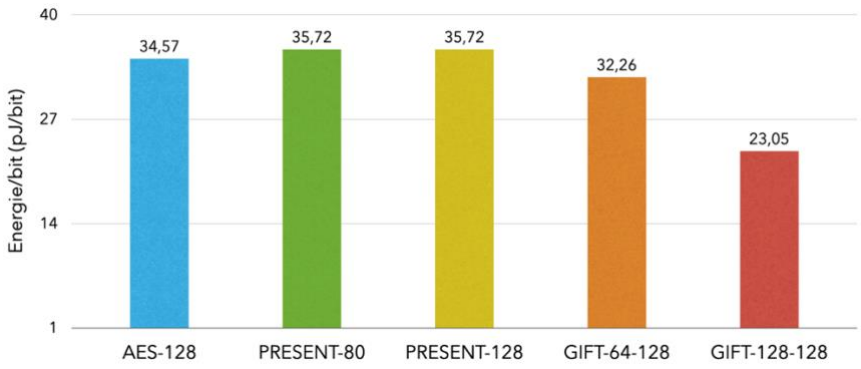


Figure 2-38 - Comparaison de l'énergie/bit des implémentations à 13,56 MHz

Tableau 2-11 - Détails des éléments logiques utilisés à 13,56 MHz

Algorithmes	Surface		
	LUT	*FF*	*Slices*
AES-128	1377	336	381
PRESENT-80	160	149	46
PRESENT-128	181	197	49
GIFT-64-128	157	198	43
GIFT-128-128	240	262	64

Les résultats ci-dessus montrent clairement les bénéfices de la cryptographie légère par rapport à l'AES-128 en termes de taille, d'un facteur 6 à 9 environ. Il en est de même pour l'efficacité, avec un écart pouvant presque atteindre un coefficient deux, au profit des algorithmes légers. En revanche, le gain en termes d'énergie/bit, calculé par l'équation (2-4) n'est pas significatif. Pour rappel, la fréquence est ici fixée à 13,56 MHz pour tous les algorithmes. La puissance consommée, estimée par l'outil *Vivado Design Suite* [42], s'élève à 6 mW pour l'AES contre 1 mW pour la cryptographie légère. Cependant, l'AES tire profit de son faible nombre de tours (10) pour une taille de donnée maximale (128 bits), ce qui lui confère une énergie/bit compétitive.

$$Energie_{bit} = \frac{NB_{Cyles} \times Puissance_{Consommée}}{Fréquence \times Taille_{Données}} \quad \textbf{(2-4)}$$

En se focalisant sur la cryptographie légère, GIFT se positionne comme étant à la fois le plus petit, le plus efficace et le moins énergivore. GIFT-64-128 atteint la plus petite taille, avec une réduction de 7% et 12% par rapport à PRESENT-80 (moins sécurisé, en termes de taille de clé) et PRESENT-128 respectivement, ainsi qu'une efficacité de 18% à 26% supérieure à celles de ces derniers. Il convient de souligner que dans le scénario précédent, PRESENT-80 possède la plus petite taille, ce qui n'est pas le cas ici. Cela s'explique par sa fréquence maximale (302 MHz) légèrement inférieure à celle de GIFT-64-128 (319 MHz), résultant sur une augmentation du nombre d'éléments logiques requis pour ce dernier.

D'un point de vue énergétique, GIFT-64-128 se place en seconde position, derrière sa version plus sécurisée GIFT-128-128. Malgré son nombre de tours conséquent (40), ce dernier permet de réduire la consommation énergétique par bit de plus de 28% par rapport à GIFT-64-128 et jusqu'à 35% de moins que PRESENT. La raison principale provient des 128 bits de données qu'il est capable de traiter, soit le double de ses homologues. En revanche, cela lui confère une augmentation de taille dont l'écart maximal s'élève à 49% par rapport à GIFT-64-128, mais ne réduit son efficacité que de seulement 6% par rapport à ce dernier, le plaçant devant les deux versions de PRESENT. Enfin, il

convient de souligner que pour un même niveau de sécurité (en termes de tailles de données et de clé), GIFT-128-128 offre un gain d'environ 50% sur l'efficacité, pour une baisse de 33% en énergie par bit et une taille d'implémentation six fois plus compacte que l'AES-128.

b) Performances suivant différents protocoles de communication de l'IdO

L'une des plus importantes motivations de la cryptographie légère est de résoudre les problèmes de sécurité dans les systèmes embarqués fortement contraints, tels que les nœuds capteurs de l'IdO. Généralement, ces derniers utilisent des protocoles de communication sans fils comme le Wi-Fi, le BLE (Bluetooth Low Energy), le ZigBee ou encore le LoRa, chacun d'eux offrant un débit radio spécifique : 11 Mbps, 1 Mbps, 250 Kbps et 50 Kbps respectivement. Puisque le but premier d'un algorithme de chiffrement est de chiffrer la donnée avant sa transmission sur le réseau, il est intéressant d'évaluer les différents cœurs cryptographiques aux débits fixés par les protocoles. Comme précédemment, seule l'implémentation « Portable » est étudiée. Pour chaque protocole, la fréquence est fixée de telle sorte qu'elle permette d'atteindre le débit maximal de ce premier. Pour rappel, le bloc *MMCM* ne peut générer des fréquences inférieures à 4.688 MHz. Par conséquent, sa sortie est fixée à 100 MHz (horloge native de la *Nexys-4*) et le bloc *Div* (compteur programmable contrôlant l'activation du *MMCM*) est configuré de manière à ce que la fréquence souhaitée soit atteinte. Les Figure 2-39 et Figure 2-40 illustrent respectivement pour chaque algorithme leur taille d'implémentation et leur efficacité, suivant les différents protocoles.

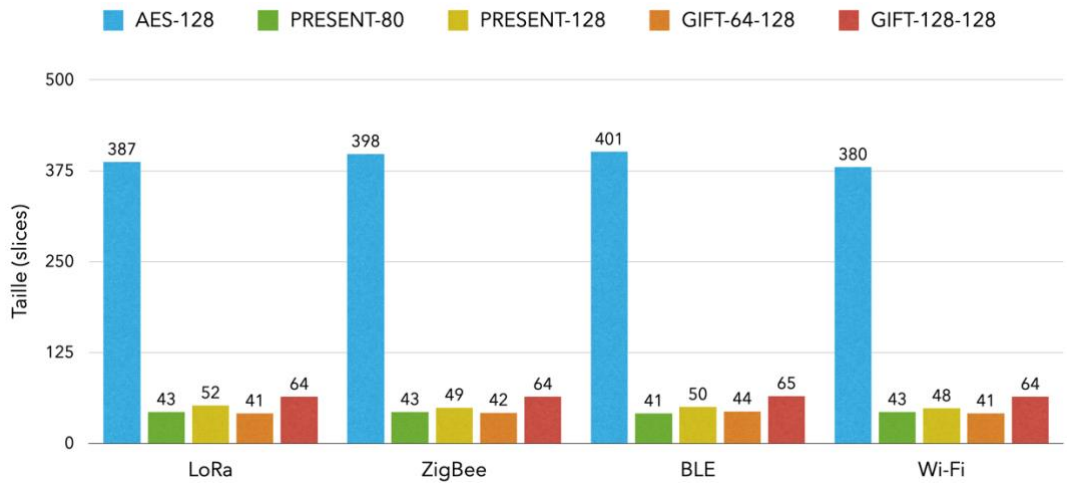


Figure 2-39 - Comparaison des tailles d'implémentations à débits fixés

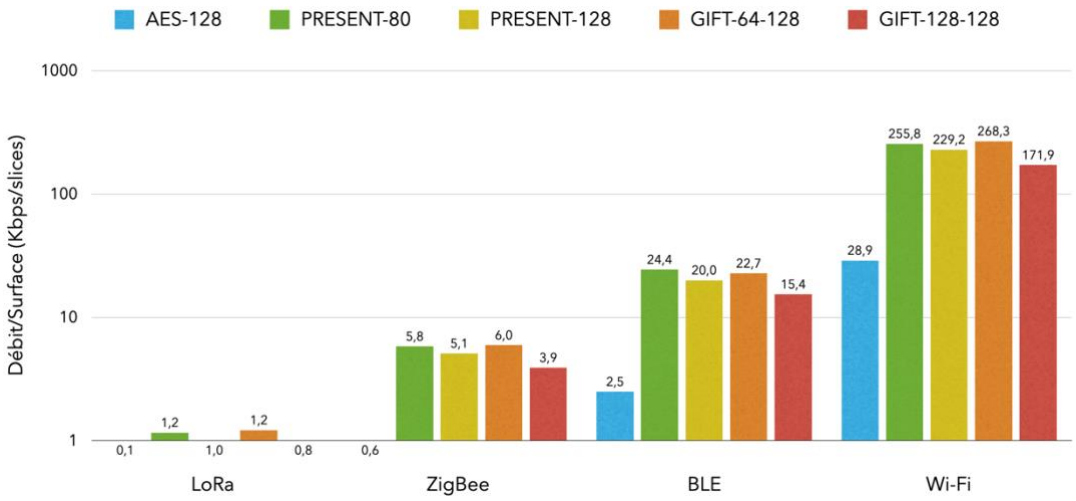


Figure 2-40 - Comparaison de l'efficacité des implémentations à débits fixés

Tout d'abord, il convient de rappeler que la taille d'implémentation et par conséquent l'efficacité dépendent fortement de la fréquence et du placement/routage, comme en témoignent les différences de performances d'un même algorithme suivant les protocoles. Comme précédemment, les histogrammes ci-dessus montrent clairement les bénéfices de la cryptographie légère par rapport à l'AES-128 en termes de taille, mais aussi d'efficacité (ratio débit/surface) d'environ un facteur 10. GIFT-64-128 est toujours le plus petit et le plus efficace. Il n'est cependant plus compact que PRESENT-80 de seulement 1% en moyenne, contre 15% face à PRESENT-128. Son efficacité est de 3% à 16% supérieure à celle de ces derniers. GIFT-128-128 voit son écart de taille avec son homologue s'élever à 53% en moyenne, avec une perte de près de 36% d'efficacité. De plus, il se place désormais comme étant le moins efficace des algorithmes légers étudiés. Il reste cependant en moyenne 6 fois plus petit que l'AES-128, pour un gain en efficacité d'un facteur 6. Ces changements de performances sont principalement liés aux différences de fréquences entre les algorithmes pour un même protocole. En effet, l'objectif ici est de fixer pour chaque couple chiffrement/protocole, la fréquence permettant d'atteindre le débit maximal défini. Le Tableau 2-12 répertorie ces résultats.

Tableau 2-12 - Fréquence des algorithmes de chiffrement suivant les protocoles utilisés

Algorithmes	Fréquence (KHz)			
	LoRa (@ 50 Kbps)	ZigBee (@ 250 Kbps)	BLE (@ 1 Mbps)	Wi-Fi (@ 11 Mbps)
AES-128	3.91	19.53	78.12	859.37
PRESENT-80	24.22	121.09	484.37	5 328.12
PRESENT-128	24.22	121.09	484.37	5 328.12
GIFT-64-128	21.87	109.37	437.50	4 812.50
GIFT-128-128	15.62	78.12	312.50	3 437.50

Une première conséquence de ces différences de fréquences entre les algorithmes implique des variations de taille, pouvant changer le classement des chiffrements. C'est le cas notamment du BLE, où PRESENT-80 est exceptionnellement plus petit que GIFT-64-128. Ces différences de fréquences expliquent également l'écart d'efficacité plus important que dans la section précédente (*e.g.* facteur 10 entre l'AES et la cryptographie légère précédemment, contre un facteur 2 ici).

Du point de vue efficacité énergétique, l'architecture particulière du système d'horloge dans ce scénario (association des blocs *MMCM* et *Div*), biaise l'estimation de la consommation. En effet, le bloc *MMCM* ne pouvant fournir de faibles fréquences, sa sortie est fixée à 100 MHz, horloge principale de la *Nexys-4* et le bloc *Div* se charge d'activer/désactiver la sortie du MMCM, afin d'atteindre la fréquence désirée. Toutefois, l'outil d'analyse énergétique de *Vivado* se base sur un cadencement à 100 MHz, donnant lieu à une puissance consommée plus élevée que précédemment : 42 mW pour l'AES et 4 mW pour les algorithmes légers (à 100 MHz), contre 6 mW et 1 mW respectivement (à 13,56 MHz). En se basant sur ces estimations, il est possible d'extrapoler l'évolution de la puissance en fonction de la fréquence, comme indiqué par les formules (2-5) et (2-6). L'histogramme Figure 2-41 illustre de manière théorique l'estimation de l'efficacité énergique des algorithmes.

$$Puissance_{AES-128} = 4{,}156 \times 10^{-10} \times Fréquence + 3{,}526 \times 10^{-4} \quad (2\text{-}5)$$

$$Puissance_{Algo_{Légers}} = 3{,}47 \times 10^{-11} \times Fréquence + 5{,}294 \times 10^{-4} \quad (2\text{-}6)$$

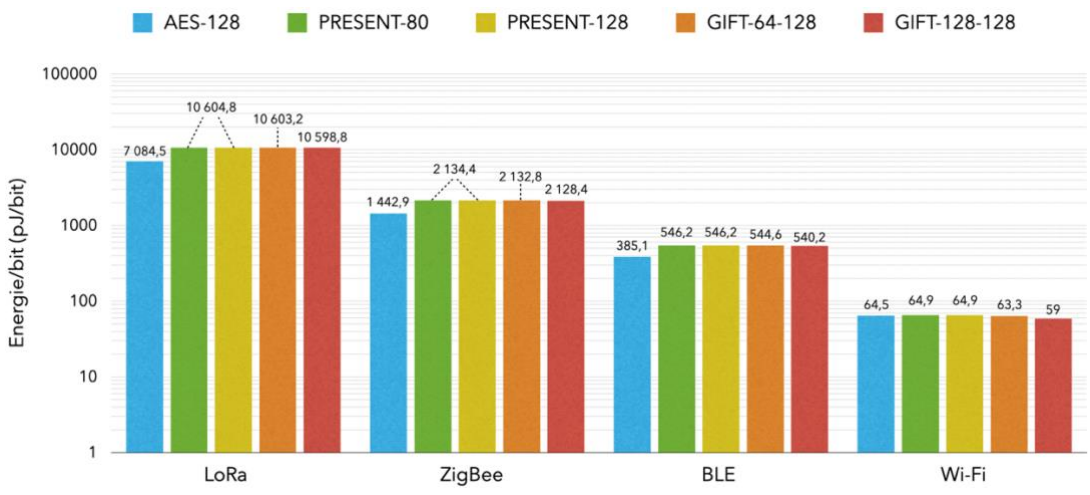


Figure 2-41 - Comparaison de l'énergie/bit des implémentations à débits fixés

Grâce à son bus de données sur 128 bits et son faible nombre de tours (10), la fréquence de l'AES permettant d'atteindre les débits des différents protocoles est jusqu'à six fois inférieure à celles de ses

homologues. Cela lui confère une puissance amoindrie, découlant sur une meilleure efficacité énergétique, jusqu'à 33% pour le protocole LoRa. À noter que cet écart s'amenuise lorsque la fréquence augmente, jusqu'à se montrer presque 10% moins efficace que GIFT-128-128 dans la configuration du Wi-Fi. Par ailleurs, ce dernier se positionne une nouvelle fois comme étant le moins énergivore, bien que la réduction maximale ne soit seulement de 7% et 10% et par rapport à GIFT-64-128 et PRESENT-80/128 respectivement (protocole Wi-Fi).

2.3.4. Synthèse

L'objectif de cette section 2.3 est de comparer les implémentations matérielles sur FPGA, d'algorithmes de cryptographies légers (PRESENT & GIFT) afin d'en souligner les gains obtenus par rapport à la cryptographie classique (AES). Grâce à des métriques communes, tels que le nombre de *slices* utilisées (taille d'implémentation), le rapport débit/surface (efficacité) ainsi que le ratio énergie/bit (efficacité énergétique), il est possible d'évaluer de manière fiable, les performances de chaque algorithme de chiffrement. Afin de fournir des résultats pertinents, une architecture susceptible d'être présente dans un système réel est proposée, embarquant une partie communication (*UART*) et le cœur cryptographique. La conception de ce dernier a elle aussi fait l'objet d'une attention particulière, rendant son intégration des plus simples possibles (« *Plug and Play* »). De plus, l'architecture interne de la partie chiffrement est là encore similaire pour l'ensemble des algorithmes étudiés, toujours dans le but de fiabiliser les comparaisons. L'évaluation est réalisée sur la carte d'évaluation *Nexys-4* de Digilent, embarquant un FPGA *Artix-7 XC7A100T-1CSG324C* de Xilinx, candidat potentiel pour le prototypage dans le contexte de l'IdO. Chaque algorithme est décliné en trois versions d'implémentations, nommées « Portable », « FF » et « SRL », en utilisant les différentes configurations des *slices* du FPGA. Enfin, la synthèse est effectuée afin d'optimiser la taille, tout en empêchant l'outil de procéder à des simplifications, pouvant impacter la sécurité de l'algorithme.

Deux types d'évaluations sont proposées. La première, à fréquence maximale des implémentations. Cela permet de quantifier les performances atteignables, mais également de comparer les résultats par rapport à l'état de l'art, utilisant majoritairement ce scénario. Les résultats ont permis de souligner l'efficacité de l'implémentation proposée « Portable », y compris par rapport aux différents travaux. La seconde évaluation se concentre sur les conditions réelles d'implémentations de l'IdO à savoir une fréquence de fonctionnement à 13,56 MHz mais aussi suivant différents protocoles de communication radio (LoRa, ZigBee, BLE et Wi-Fi). Bien que l'AES atteigne un meilleur débit grâce à son faible nombre de tours et la dimension de son bus de données, sa taille peut être jusqu'à 10 fois supérieure à celles de PRESENT et GIFT, réduisant son efficacité (ratio débit/surface). De par l'extrapolation de la puissance en section 2.3.3.b), l'AES se montre

cependant légèrement meilleur énergétiquement, pour les protocoles à faible débit. Toutefois, cela ne lui permet pas de compenser son empreinte matérielle, confirmant son caractère inadapté pour les objets fortement contraints. De manière générale, GIFT se positionne comme étant le plus efficace. La version GIFT-64-128 atteint les meilleures performances dans les différentes évaluations. Pour un même niveau de sécurité (en termes de taille de clé), GIFT-64-128 est entre 1,5 et 10 fois plus efficace que l'AES-128 (configuration 13,56 MHz et suivant les protocoles radio respectivement). De plus, il se montre entre 12% et 19% plus petit que PRESENT-128, pour un gain en efficacité compris entre 16% et 26% (suivant la configuration des protocoles radio et l'implémentation à 13,56 MHz respectivement), jusqu'à un gain atteignant les 53% (configuration à fréquence maximale). Pour un ajout maximal en taille de 53% et une baisse de 36% d'efficacité par rapport à GIFT-64-128, la version plus sécurisée GIFT-128-128 atteint une meilleure efficacité énergétique de 7% à plus de 28% que son homologue, suivant la configuration des protocoles radio et l'implémentation à 13,56 MHz respectivement. Cela est principalement dû à ses données sur 128 bits lui offrant un meilleur débit. Sa faible taille comparée à l'AES-128 lui permet d'être jusqu'à 6 fois plus petit et plus efficace que ce dernier. Ainsi, au travers de ces différents résultats, GIFT s'avère être un candidat idéal pour sécuriser les objets fortement contraints tout en optimisant l'utilisation des ressources.

2.4. Évaluations de l'AES, PRESENT et GIFT face aux attaques par canaux auxiliaires

La section précédente met en avant les gains de la cryptographie légère par rapport à la classique représentée dans ces travaux par l'AES-128. Cependant, les caractéristiques d'implémentation ne doivent pas être le seul critère de sélection d'un algorithme de chiffrement. Puisque son rôle est d'assurer la sécurité d'un système (*i.e.* confidentialité, intégrité et authenticité), sa robustesse doit faire l'objet d'une attention particulière. En effet, bien que les méthodes de contremesures soient les mêmes pour les deux types de cryptographie [76], [80], [81] et [82], de nouvelles attaques sont apparues sur les algorithmes légers, exploitant la structure binaire de leur fonction de diffusion (couche de permutation) [83], [84], [85] et [86]. Par conséquent, cette section propose de comparer le niveau de sécurité de l'AES, PRESENT et GIFT notamment contre les attaques par canaux auxiliaires. Bien sûr, de nombreux travaux évaluant la robustesse de ces algorithmes sont disponibles dans la littérature, comme ceux précédemment cités, mais les comparaisons sont souvent difficiles en raison des différentes technologies/implémentations [36], [63] et méthodes employées [84], [83]. Les travaux de cette thèse n'ont pas pour ambition d'introduire de nouvelles attaques, mais se montrent particulièrement intéressants grâce à l'utilisation critères communs à la fois d'un point de vue attaques et implémentations (*cf.* section 2.3). De ce fait, cette étude peut une nouvelle fois servir de guide à tout concepteur de systèmes embarqués dans le choix d'un algorithme de chiffrement.

2.4.1. Cryptanalyse et analyse des canaux auxiliaires

Dès sa conception, un algorithme cryptographique se doit d'être résistant à la cryptanalyse. Cette dernière constitue la science regroupant différentes techniques ayant pour but de retrouver le message clair à partir du chiffré, sans avoir connaissance du secret. La méthode naïve illustrant ce procédé est l'attaque dite par « force brute », qui consiste à tester toutes les possibilités de la clé. Toutefois, elle s'avère impossible en pratique lorsque les principes de Kerckhoffs sont respectés. En effet, l'un d'eux stipule que la sécurité d'une fonction cryptographique doit uniquement résider dans la clé. Cela signifie d'une part que le fonctionnement détaillé de l'algorithme ne doit pas faire l'objet d'un secret, et d'autre part, la clé doit être suffisamment grande pour qu'une recherche exhaustive soit impossible. Ainsi, la cryptanalyse vise à exploiter les potentielles faiblesses mathématiques d'une fonction cryptographique, offrant une complexité d'attaque inférieure à celle par « force brute ». Toutefois, ces algorithmes ont pour but d'être implémentés dans des systèmes réels, de manière logicielle ou matérielle. Cela introduit un nouveau vecteur d'attaque, cette fois lié au comportement de l'algorithme ou encore aux données qu'il traite. Ce type d'exploitation se nomme l'analyse des canaux auxiliaires (*Side Channel Analysis*).

Malgré leur résistance face à la cryptanalyse classique, toute fonction cryptographique, dès lors qu'elle est implémentée, induit des phénomènes physiques (*e.g.* consommation de courant, émanations électromagnétiques) qui peuvent être exploités par un attaquant. Afin de familiariser le lecteur avec ce principe, une analogie avec une situation du quotidien peut faciliter la compréhension : le contrôle d'accès à un immeuble par un digicode. En reprenant le principe de l'attaque par force brute (cryptanalyse), l'attaquant peut naïvement essayer toutes les combinaisons pour retrouver le code d'accès. Il devient évident que plus la taille de ce dernier est importante, plus l'attaque devient complexe. Face à son échec, l'attaquant peut alors orienter physiquement son approche, dans le but de contourner la sécurité apportée par le digicode. Une première solution consiste à démonter le boitier du digicode. De là, après analyse, il peut en déduire le mécanisme contrôlant l'ouverture de la porte (*e.g.* un commutateur) et forcer l'activation de celui-ci (*e.g.* en le court-circuitant). Le digicode étant physiquement atteint, ce type d'approche fait partie de la famille des attaques invasives [87]. Toutefois, il apparait évident que dans un tel scénario, l'attaquant doit se munir de matériels spécifiques (*i.e.* démontage physique du boitier, court-circuitage) mais aussi d'avoir un certain niveau de connaissance (*i.e.* analyse du système d'ouverture). Une seconde possibilité consiste à positionner par-dessus le digicode, un clavier factice capable d'enregistrer l'appuie sur chaque bouton. Ainsi, après le passage d'une seule personne possédant le code, ce dernier peut être révélé à l'attaquant, après analyse des enregistrements du faux clavier. On parle alors d'attaque semi-invasive [88], dans la mesure où une intervention physique sur l'élément est nécessaire. Enfin, une dernière approche s'offre à l'attaquant, cette fois de manière non invasive. En

effet, en utilisant un crimescope, lampe laser diffusant des rayons ultraviolets et infrarouges, il est possible de repérer les traces organiques telles que les empreintes digitales. Les numéros constituant le code d'accès sont alors révélés à l'attaquant. Ce dernier devra toutefois les remettre dans l'ordre, mais le nombre de possibilités est considérablement diminué par rapport à une attaque exhaustive sur l'ensemble des numéros. Ainsi, l'approche non invasive est souvent peu couteuse en raison du peu de matériel requis et ne demande pas un fort niveau de connaissance. De ce fait, elle est considérée comme étant la plus dangereuse, en raison de son abordabilité.

a) Attaques par perturbation

On distingue deux familles d'attaques non invasives. La première, dite par « perturbation », a pour objectif d'injecter des erreurs dans les calculs. Plusieurs méthodes sont utilisées à ce jour, comme la variation de tension d'alimentation [89] ou de l'horloge [90]. Plus récemment, l'attaque *Rowhammer* [91], exploite un effet secondaire de la technologie mémoire DRAM (*Dynamic Random Access Memory*), permettant d'inverser de manière déterministe les bits des cellules mémoires sans avoir les droits d'accès. Ces différentes techniques, ayant plus ou moins de précision, ont pour but d'exploiter ces erreurs afin de retrouver les informations secrètes d'une fonction cryptographique [92].

b) Attaques par observation

À l'inverse des attaques par perturbation, celles dites par « observation » constituent la seconde famille des non invasives. Là encore, plusieurs méthodes sont à l'œuvre comme l'analyse du temps d'exécution [32], [33], la consommation de courant [34], et le rayonnement électromagnétique [35]. Toutefois, cette dernière approche offre l'avantage de ne cibler qu'une partie précise du système à attaquer (*e.g.* la fonction cryptographique) ce qui la rend particulièrement efficace. De plus, elle bénéficie d'une installation des plus simples : il suffit de positionner au-dessus du système cible une sonde électromagnétique, couplée à un amplificateur pour augmenter l'amplitude du signal. Néanmoins, dans le cas où l'attaquant ignore l'architecture du système, il doit au préalable cartographier ce dernier en déplaçant la sonde, afin de détecter la position de la fonction cryptographique à attaquer. Dans le contexte de l'IdO, les systèmes sont généralement peu couteux et facilement atteignables physiquement. Leur nombre augmentant de manière exponentielle, ils interagissent de plus en plus avec nos données, les rendant propices aux attaques, notamment par canaux auxiliaires. Associant à la fois simplicité et faibles ressources matérielles, l'analyse par rayonnement électromagnétique s'avère être un vecteur d'attaque idéal. En conséquence, la suite de ce travail s'oriente sur cette approche.

2.4.2. Plateforme SECNUM

Les travaux de cette section 2.4 ont été effectués à l'aide de la plateforme SECNUM [93]. Dédiée à l'étude des vulnérabilités des circuits intégrés, notamment celles en lien avec les canaux auxiliaires, cette plateforme de recherche se matérialise par un banc d'attaque exploitant le rayonnement électromagnétique, présentée sur la Figure 2-42. Elle s'accompagne de son pendant éducation, porté sur la sensibilisation (présentations et démonstrations) ainsi que sur l'enseignement théorique et pratique (conférences, stages technologiques), adapté à différents profils (techniciens, ingénieurs et chercheurs).

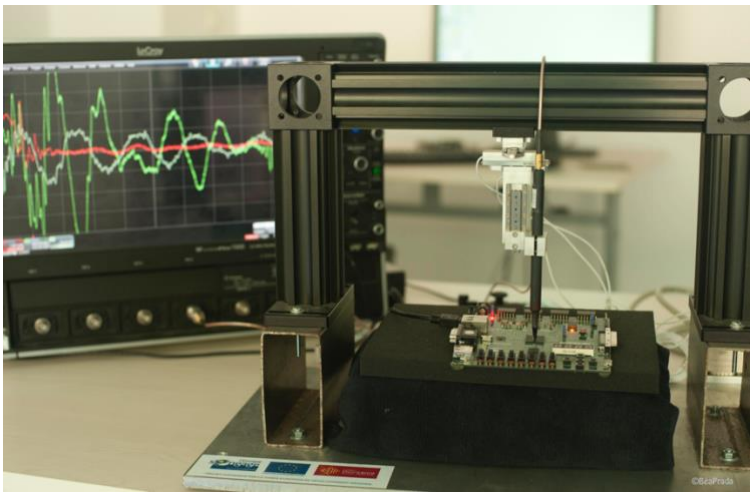


Figure 2-42 – SECNUM, plateforme d'attaques par canaux auxiliaires

2.4.3. Principe de l'attaque par observation du rayonnement électromagnétique

Lors de son fonctionnement, la consommation de courant d'un système, mais également les émanations électromagnétiques qui en résultent, sont directement liées aux données qu'il manipule. Cette propriété devient particulièrement intéressante lorsqu'il s'agit d'une primitive cryptographique, dont certaines opérations utilisent une partie de la clé de chiffrement. En exploitant ce phénomène, il est alors possible de retrouver cette dernière. Le principe général de l'attaque consiste à retrouver les résultats intermédiaires d'une opération spécifique du chiffrement, dépendant à la fois de données connues (*e.g.* message clair/chiffré) et de la clé (*i.e.* sous-clé utilisée lors d'un tour). En découpant cette dernière en petites parties (*e.g.* sur 8 bits), il est possible d'émettre de manière exhaustive des hypothèses sur la valeur de chacune d'elles, permettant de lister la totalité des résultats intermédiaires possibles de l'opération choisie précédemment. Enfin, par analyse statistique avec le rayonnement électromagnétique et un modèle dit « de fuite », la clé principale peut alors être

retrouvée, réduisant à néant la sécurité de l'algorithme de chiffrement. La Figure 2-43 illustre ce principe de fonctionnement. Les différentes étapes, numérotées de ① à ④ sont détaillées dans les sections 2.4.3.a) à 2.4.3.d) respectivement.

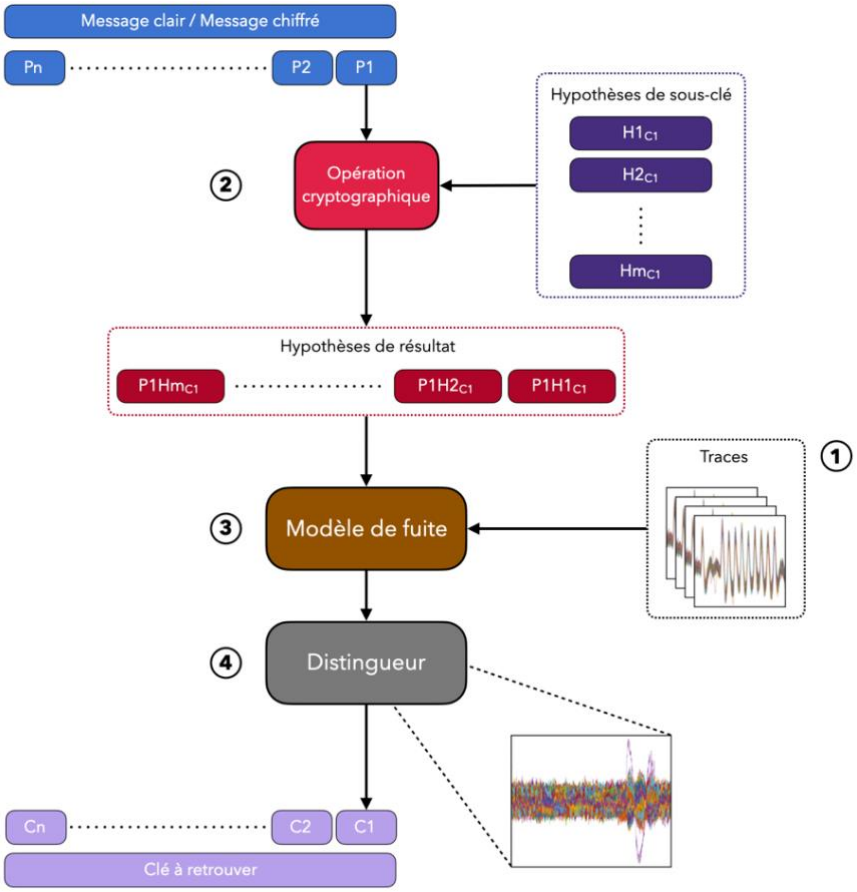


Figure 2-43 - Principe de fonctionnement d'une attaque par canaux auxiliaires

a) Enregistrement du rayonnement électromagnétique

L'étape ① consiste à enregistrer les fuites électromagnétiques de la fonction cryptographique, à l'aide d'un oscilloscope et d'une sonde. Pour cela, l'attaquant définit des messages clairs (*e.g.* aléatoirement) qu'il injecte dans l'algorithme afin d'en obtenir les messages chiffrés. Ce scénario est qualifié d'attaque « à textes connus ». Durant les chiffrements successifs, les fuites électromagnétiques sont captées et converties par la sonde en signaux électriques, lesquels sont ensuite enregistrés suivant le format choisi par l'attaquant (fichiers, matrices de points, etc.). Ces enregistrements sont communément appelés « traces ». Suivant leur précision (positionnement de la sonde, fréquence d'échantillonnage, amplitude, etc.), mais également de la nature de l'algorithme de chiffrement, le nombre de traces nécessaires pour extraire la sous-clé puis retrouver la principale

varie. Le résultat de cette procédure donne lieu au trio messages clairs, messages chiffrés et traces, qu'il est alors possible d'exploiter.

b) Sélection de l'opération cryptographique à attaquer

Le rôle de la phase ② est de sélectionner l'opération spécifique de l'algorithme de chiffrement, dont les résultats intermédiaires sont susceptibles de fuiter des informations sur la clé. Le choix dépend de plusieurs facteurs tel que l'algorithme lui-même, mais aussi son implémentation. La complexité de l'attaque, c'est-à-dire la taille et/ou le nombre de parties de sous-clé simultanément traitées, dépend directement de celle de l'opération choisie. C'est le cas des attaques dites « algébriques » [94], qui se basent sur des techniques algébriques pour reconstituer la clé principale à partir de plusieurs parties de sous-clé(s). Toutefois, une attaque d'une complexité inférieure n'est pour autant pas moins efficace, et peut s'avérer être plus dangereuse de par son accessibilité.

Comme expliqué précédemment, les fuites d'information sur la clé sous-entend que celle-ci doit être impliquée dans l'opération sélectionnée. En découpant judicieusement cette dernière en petites parties, il devient possible de dresser une liste exhaustive de toutes ses valeurs. Il convient également de s'assurer que l'opération utilise une donnée connue par l'attaquant, c'est-à-dire le message clair ou chiffré, ce qui positionne l'attaque proche du début ou de la fin du processus de chiffrement. Enfin, il est important de souligner que la propriété non linéaire d'une opération offre un avantage remarquable, de par le fort écart qu'il en résulte entre une bonne et mauvaise hypothèse de sous-clé. Ainsi, les opérations de substitutions (*SBOX*) s'avèrent être un point particulièrement intéressant.

c) Modèles de fuite

À ce stade, l'attaquant est en possession du trio messages clairs, messages chiffrés et traces associées (fuites électromagnétiques, *cf.* section 2.4.3.a). Après identification de l'opération à attaquer, il possède également les résultats intermédiaires de cette dernière, obtenus par hypothèses sur chaque partie de la sous-clé (*cf.* section 2.4.3.b). Afin de mener à bien l'attaque, l'étape ③ consiste à déterminer un lien entre ces éléments, grâce à un modèle de fuite. Ces derniers varient suivant l'architecture du système cible et du type d'attaque. Dans le cadre des fuites par consommation de courant (et par conséquent par rayonnement électromagnétique), deux d'entre eux sont particulièrement employés :

Distance de Hamming (*HD*) : ce modèle suppose que l'inversion de la valeur d'un bit ('0'→'1', '1'→'0') entraine une consommation de courant plus importante que celle requise pour le maintien de sa valeur actuelle ('0'→'0', '1'→'1'). En d'autres termes, la distance de Hamming correspond au

nombre de bits différents entre deux valeurs, comme en témoigne l'équation (2-7). En raison de sa cohérence avec le fonctionnement d'un inverseur CMOS, ce modèle est généralement utilisé.

$$HD(x, y) = \sum_{i=0}^{n-1} (x_i \oplus y_i) \tag{2-7}$$

Poids de Hamming (*HW*) : ce modèle suppose que la manipulation d'un '1' logique induit une consommation de courant différente que celle d'un '0' logique. En d'autres termes, cela signifie que la consommation liée à une valeur dans un circuit est corrélée au nombre de bits à '1' qu'elle possède, comme présenté par l'équation (2-8). Il est important de noter que ce modèle est privilégié lorsque l'attaquant n'a pas suffisamment d'information sur l'architecture du système. Il en est de même lorsque les précédents résultats intermédiaires de l'opération à attaquer ne sont pas accessibles sur les traces, rendant le modèle de distance de Hamming inefficace. Par ailleurs, certaines architectures utilisent des stratégies de précharge, fixant généralement à '0' le bus de données avant l'écriture de l'information [95]. De ce fait, la distance de Hamming est effectuée entre '0' et la valeur à écrire, ce qui n'est autre que le poids de Hamming (ou son complément lors d'une précharge à '1').

$$HW(x) = \sum_{i=0}^{n-1} x_i \tag{2-8}$$

Bien qu'ils diffèrent selon leur postulat de base, il existe un lien étroit entre ces deux modèles. En effet, la distance de Hamming correspond au poids de leur différence, comme le montre l'équation (2-9).

$$HD(x, y) = HW(x \oplus y) \tag{2-9}$$

d) Distingueur

Une fois le modèle de fuite choisi, le lien statistique entre la meilleure hypothèse de chaque partie de sous-clé et les traces est calculé à l'aide d'un distingueur, constituant l'étape ④. Parmi les plus utilisés se trouve la « différence des moyennes » DPA (*Differential Power Analysis*) [96] ou DEMA (*Differential Electromagnetic Analysis*) dans le domaine d'étude électromagnétique. Le principe est de partitionner les traces en plusieurs catégories, généralement deux, suivant un critère (*e.g.* la valeur d'un bit spécifique). Après quoi, la différence des moyennes de chaque groupe est effectuée pour chaque hypothèse de sous-clé, la meilleure donnant lieu au plus grand écart entre les moyennes. Cependant, la DPA est sensible au phénomène des « pics fantômes » produit par les propriétés de la couche de substitution [97], ce qui engendre un lien statistique anormalement élevé pour des hypothèses de sous-clé erronées. En conséquence, la méthode CPA (*Correlation Power Analysis*) ou

CEMA (*Correlation Electromagnetic Analysis*) a été proposée [98]. Son principe de fonctionnement reprend les étapes ② et ③, à savoir la prédiction de résultats intermédiaires en fonction des hypothèses de sous-clés, sur lesquels le modèle de fuite (*i.e.* distance ou poids de Hamming) est appliqué. Enfin, un coefficient de corrélation de Pearson est calculé entre les traces et les prédictions. Sa valeur est maximale pour l'hypothèse de sous-clé correcte. L'équation (2-10) définit le calcul du coefficient, avec les opérateurs « Cov » et « Var » signifiant la covariance et la variance respectivement.

$$Corr(Prédictions, Traces) = \frac{COV(Prédictions,\ Traces)}{\sqrt{Var(Prédiction)} * \sqrt{Var(Traces)}} \quad (2\text{-}10)$$

Contrairement à la DEMA, la méthode CEMA ne dépend pas d'un critère de partitionnement des traces, mais sur la pertinence du modèle de fuites utilisé. Elle se trouve ainsi moins sensible aux « pics fantômes » tout en augmentant son efficacité en termes de nombre de traces requises. De ce fait, la suite de ce travail se focalise sur cette méthode.

2.4.4. Protocole expérimental

Tous les algorithmes sont implémentés sur la carte *Nexys-4* [78], embarquant un FPGA *Artix 7 XC7A100T-1CSG324C*, présentée sur la Figure 2-44. Dans le cadre d'une évaluation de robustesse, il est généralement admis que l'étude soit réalisée dans les conditions idéales. Afin d'améliorer le rapport signal sur bruit, le cœur cryptographique (*pblock_CRYPTO*) est séparé du système de contrôle de l'architecture (*pblock_CTRL*, comprenant le bloc *UART*, la machine d'état, etc.), comme illustré par la Figure 2-45. De manière à augmenter la signature électromagnétique, l'ensemble du système est cadencé à 100 MHz, fréquence native de la *Nexys-4*. De plus, le débit de transmission du bloc *UART* est fixé à 230400 bauds, pour accélérer les campagnes de mesures. À noter qu'en raison du placement, mais aussi de la fréquence utilisée, les tailles d'implémentations peuvent légèrement varier par rapport à celles données dans la section 2.3.

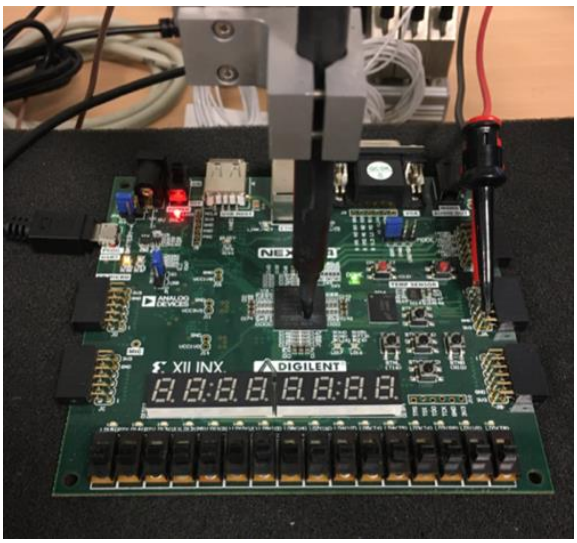


Figure 2-44 - Carte FPGA Nexys-4 [78] et la sonde électromagnétique [99]

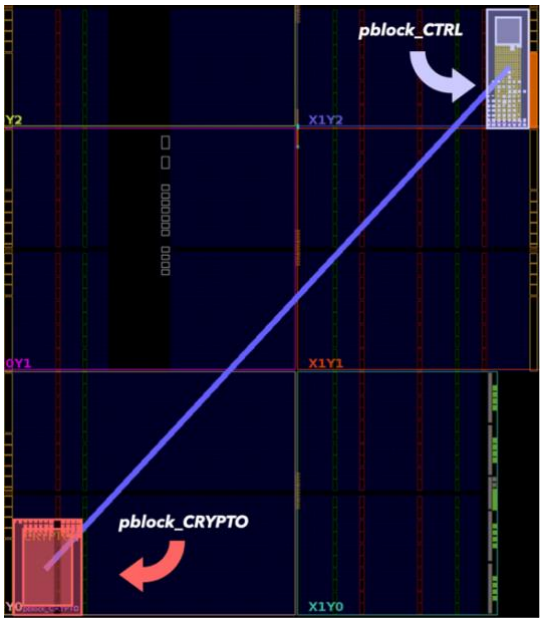


Figure 2-45 - Placement du cœur cryptographique pour l'évaluation de robustesse

Le rayonnement électromagnétique est capturé et converti en signal électrique par une sonde de champ proche passive Langer RF-U 2.5-2 [99], puis amplifié par un ampliateur FEMTO haute vitesse HSA-X-2-40 d'un gain de 40 dB [100]. Un oscilloscope LeCroy [101], présenté sur la Figure 2-46, permet d'acquérir et d'afficher le signal afin de trouver le meilleur point de fuite des algorithmes

sur le FPGA où le rayonnement est le plus intense. Dans le but d'améliorer l'expérience, un signal indiquant le début et la fin du chiffrement signal (en rouge sur la Figure 2-46) a été ajouté au système. Cela permet de déclencher l'acquisition des traces de manière à ne visualiser que les fuites du cœur cryptographique. De plus, afin d'augmenter la précision des mesures, chaque message clair est chiffré six fois consécutivement. Un moyennage des traces est ensuite effectué, permettant d'obtenir une image plus précise des fuites électromagnétiques. Cela est d'autant plus vrai lorsque le système cible est petit et consomme peu (*e.g.* les algorithmes légers).

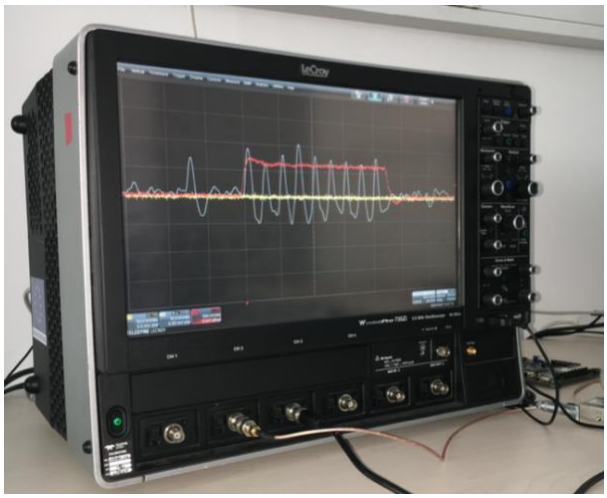


Figure 2-46 - Oscilloscope utilisé pour la récupération des traces

Comme évoqué précédemment, l'opération cryptographique spécifique et le modèle de fuite sont déterminants sur l'efficacité de l'attaque. Pour rappel, cette dernière peut être menée à partir des messages clairs, ou des messages chiffrés. Cela signifie que l'attaque s'effectue soit sur les *SBOX* du premier tour de l'algorithme de chiffrement, soit du dernier, respectivement. Bien que l'architecture et la technologie de la cible soient à prendre en compte, l'attaque sur le dernier tour est majoritairement privilégiée dans la littérature. Cela peut également s'expliquer par la qualité des traces plus précises en fin de chiffrement, car potentiellement moins perturbées par le reste du système (*e.g.* écritures du message clair et de la clé dans les registres du cœur cryptographique). Par conséquent, l'attaque CEMA sur le dernier tour sera utilisée dans cette étude. Il convient alors de déterminer le modèle de fuite à utiliser. En raison de sa cohérence avec le modèle de consommation de l'inverseur CMOS, la distance de Hamming s'avère être l'approche la plus pertinente. Cette dernière repose sur les changements de bits entre deux valeurs consécutives d'une donnée (*e.g.* en entrée ou sortie des *SBOX*). La Figure 2-47 présente l'exploitation de ce modèle, en rappelant les opérations de base constituant un algorithme de chiffrement.

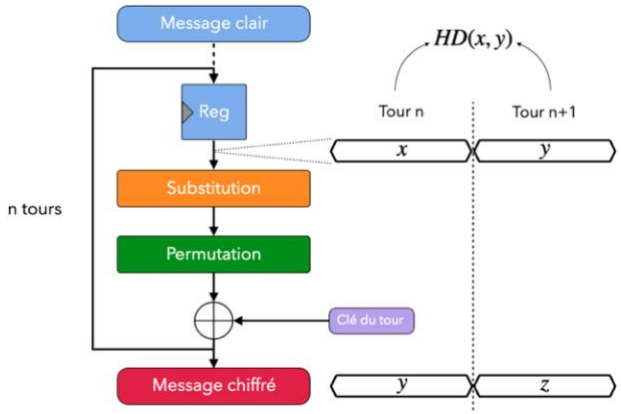


Figure 2-47 - Exploitation de la distance de Hamming

Puisque chaque tour dépend de son prédécesseur, la sortie du *XOR* (⊕) est rebouclée en entrée, et le registre (*REG*) est mis à jour avec le résultat. Lors du dernier tour, la valeur de celui-ci équivaut à x donnant lieu au message chiffré y. Le rebouclage s'effectue une nouvelle fois, recopiant la sortie du *XOR* sur l'entrée. La valeur du registre passera de x à y au cycle suivant, rendant alors possible le calcul de la distance de Hamming, à partir du message chiffré connu. Néanmoins, cette structure requiert toutefois un tour supplémentaire, nécessaire à la mise à jour du registre. Afin d'éviter cela, l'architecture proposée dans cette thèse conditionne le rebouclage, et plus spécifiquement, la mise à jour du registre (*REG*) lors du dernier tour, illustrée par la Figure 2-48.a). En effet, puisque le message chiffré est directement disponible en sortie du *XOR* à la fin du dernier tour, il est inutile d'opérer le rebouclage. De ce fait, le registre des résultats intermédiaires ne prend pas la valeur du message chiffré (y), rendant la distance de Hamming inadéquate, au profit du modèle de poids de Hamming.

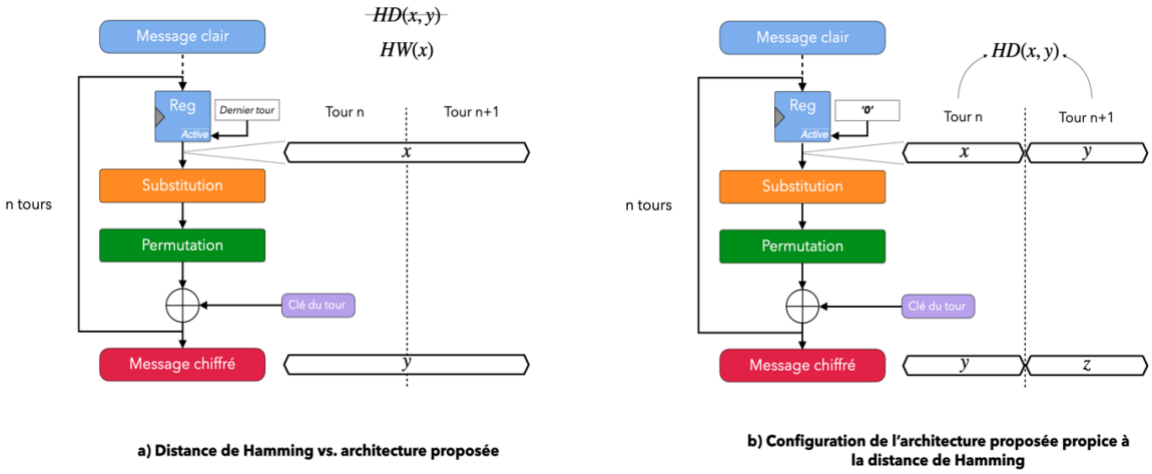


Figure 2-48 - Exploitation de la distance de Hamming sur l'architecture proposée

Cependant, la distance de Hamming étant le modèle de fuite le plus pertinent pour la technologie CMOS (*cf.* section 2.4.3.c), il est intéressant de l'exploiter, d'une part à des fins de performances d'attaque, mais aussi pour exposer une évaluation plus générale et non spécifique à une implémentation particulière. Pour cela, une simple modification de l'architecture proposée est nécessaire : le signal de contrôle du registre (*ACTIVE*) est fixé à '0', autorisant la mise à jour inconditionnelle de ce dernier, comme illustrée sur la Figure 2-48.b). Cette configuration permet de revenir à une architecture classique, telle que présentée précédemment (*cf.* Figure 2-47). Pour être complet, la partie de contrôle est légèrement adaptée pour supporter ce changement, mais n'entre pas en considération dans le scénario d'attaque. Ainsi, l'évaluation de la robustesse des algorithmes de chiffrement est effectuée suivant la méthode CEMA, mesurant les fuites électromagnétiques induites par les *SBOX* du dernier tour, en utilisant la distance de Hamming comme modèle de fuite.

Outre les éléments précédents, le processus de génération des sous-clés (bloc *Génération Sous-Clés*) joue un rôle déterminant dans le nombre de tours à attaquer pour retrouver la totalité de la clé principale. Les Figure 2-49 à Figure 2-53 illustrent cette caractéristique pour les algorithmes étudiés.

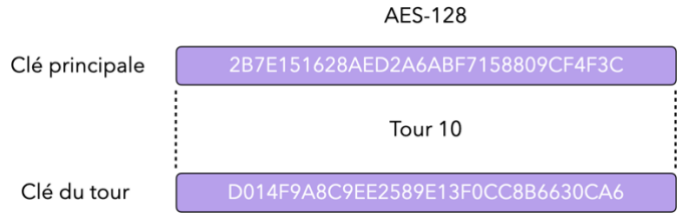


Figure 2-49 - Sous-clé du dernier tour de l'AES-128

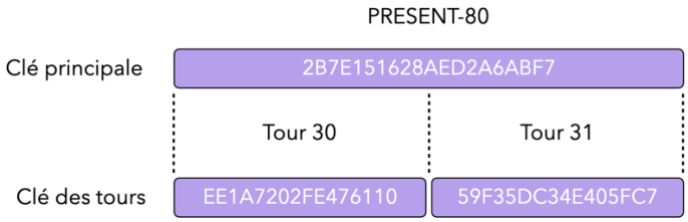


Figure 2-50 - Sous-clés des derniers tours de PRESENT-80

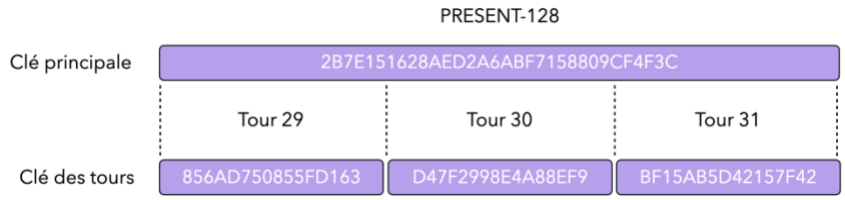


Figure 2-51 - Sous-clés des derniers tours de PRESENT-128

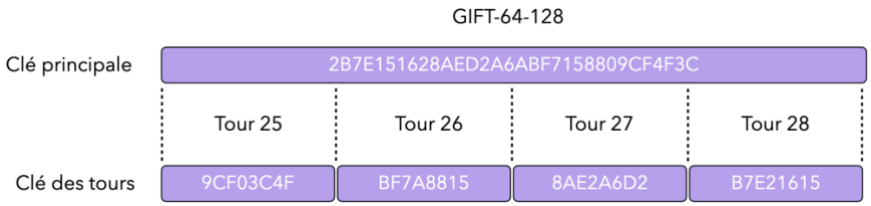


Figure 2-52 - Sous-clé des derniers tours de GIFT-64-128

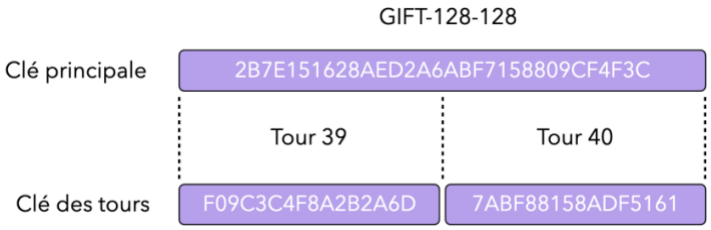


Figure 2-53 - Sous-clés des derniers tours de GIFT-128-128

Le cas de l'AES-128 est le plus simple. Ce dernier utilise l'intégralité des 128 bits de la clé principale, ce qui permet de n'attaquer qu'un seul tour pour retrouver cette dernière. Les deux versions de PRESENT utilisent principalement une rotation à gauche de 61 bits de la clé principale. Par conséquent, il faut attaquer les deux derniers tours pour PRESENT-80 (64 + 16 bits) et les trois derniers pour PRESENT-128 (64 + 61 + 3 bits). Enfin, GIFT utilise exclusivement des rotations à droite de 2, 12 et 32 bits. Par conséquent, il faut quatre tours pour retrouver la clé principale de GIFT-64-128 et seulement deux pour GIFT-128-128.

2.4.5. Évaluation de l'attaque CEMA

L'objectif de cette évaluation de robustesse est de quantifier le nombre de traces nécessaires pour retrouver la clé principale de chiffrement, communément appelé le *Success Rate*. Cette métrique correspond au pourcentage de cette dernière, retrouvée par un attaquant au cours du processus, en fonction du nombre de traces. Étant donné la réversibilité de leur génération, il est admis que lorsque suffisamment de sous-clés sont retrouvées (*cf.* section 2.4.4), l'attaquant est en mesure d'en déduire la clé principale. Afin d'avoir une comparaison fiable, l'attaque est effectuée dans les mêmes conditions pour les différents algorithmes, à savoir un positionnement optimal de la sonde, un moyennage de six chiffrements par message clair, ainsi qu'un ciblage sur 8 bits de clé à la fois. De plus, pour PRESENT et GIFT, nécessitant une attaque sur plusieurs tours, le choix a été fait de respecter l'interdépendance des tours, potentielle force de ces algorithmes. Cela signifie que l'attaque du tour *n-1* ne peut réussir que si celle du tour *n* l'est aussi. Enfin, une évaluation empirique est menée sur les traces, afin d'extraire celles qui fuitent le plus d'information (*i.e.* recherche du meilleur rapport signal sur bruit). Ces dernières sont représentées sur la Figure 2-54. La Figure 2-55 illustre la comparaison des *Success Rates*, image de la robustesse des algorithmes face à l'attaque CEMA. L'ensemble des fichiers sources et des traces est disponible en ligne (*https://gite.lirmm.fr/ldalmasso/LightweightCryptoCiphers.git*).

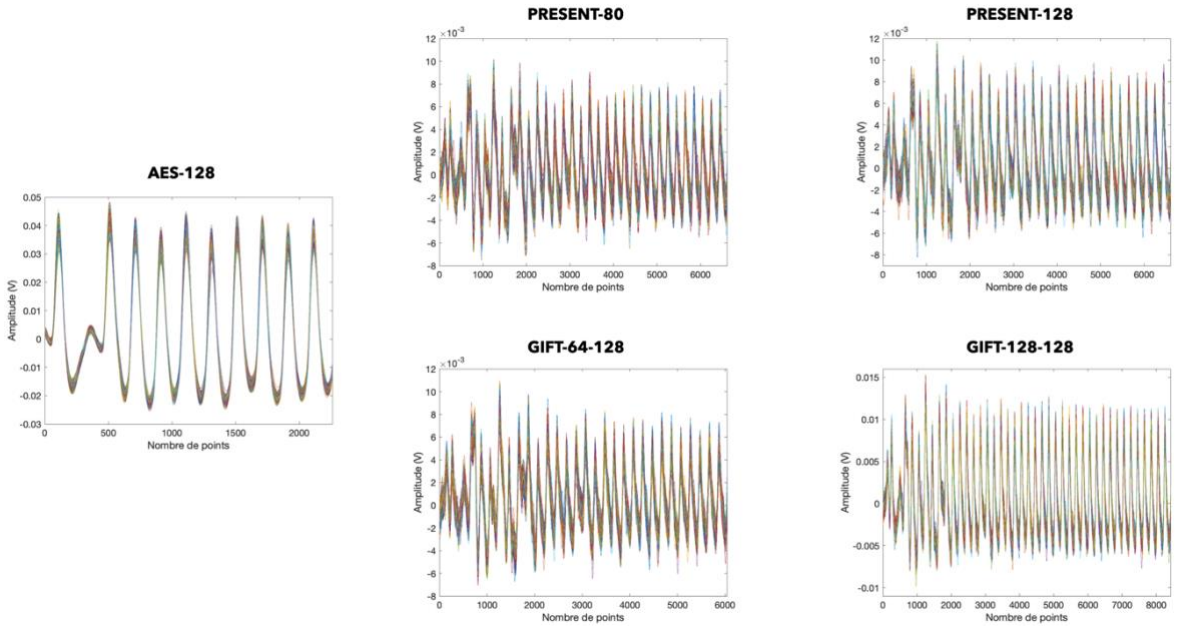


Figure 2-54 - Traces expérimentales de l'AES-128, PRESENT-80/128 et GIFT-64-128/128-128

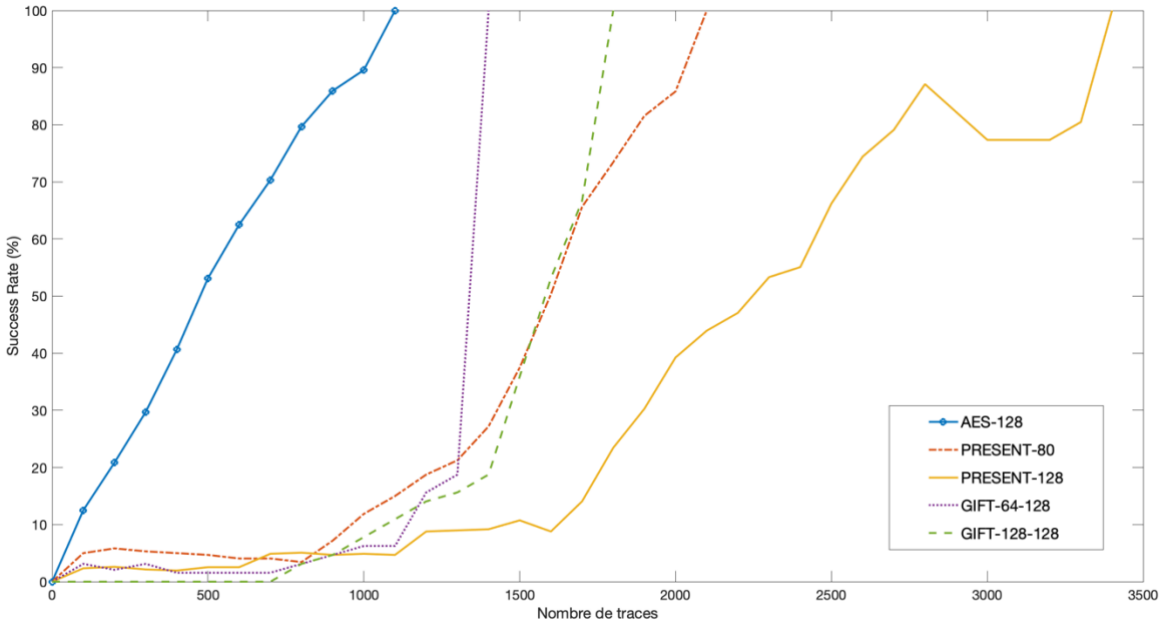


Figure 2-55 - Comparaison de robustesse de l'AES-128, PRESENT-80/128 et GIFT-64-128/128-128

D'après l'évaluation, PRESENT s'avère être le plus résistant face à l'attaque CEMA, dans ses deux versions, suivies par celles de GIFT. L'AES-128 se montre être le plus vulnérable, avec une résistance divisée jusqu'à un facteur 3 par rapport à PRESENT-128. Afin d'expliquer ces résultats, les Figure 2-56 à Figure 2-59 décrivent plus en détail le modèle d'attaque employé pour chaque algorithme.

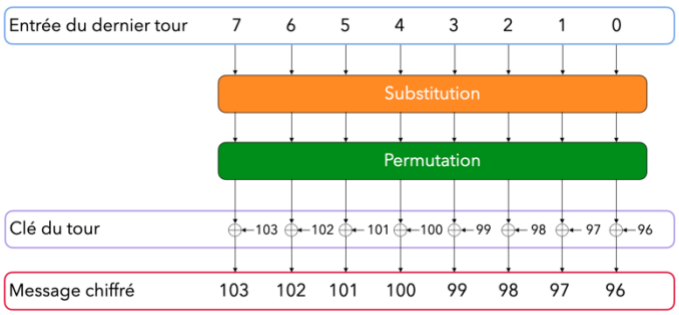


Figure 2-56 - 8 bits de poids faible du dernier tour de l'AES-128

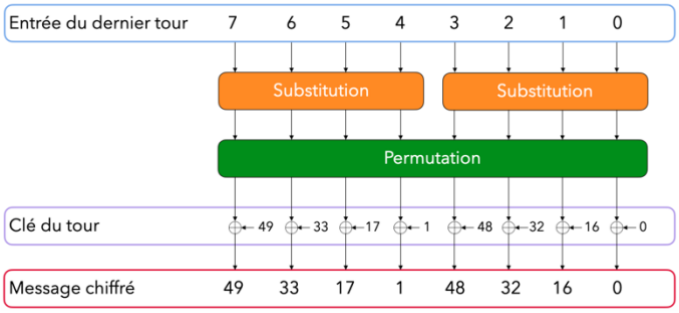


Figure 2-57 - 8 bits de poids faible du dernier tour de PRESENT-80 / PRESENT-128

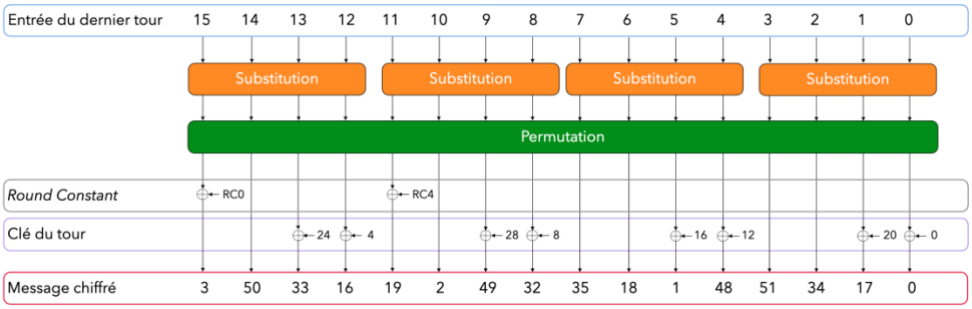


Figure 2-58 - 16 bits de poids faible du dernier tour de GIFT-64-128

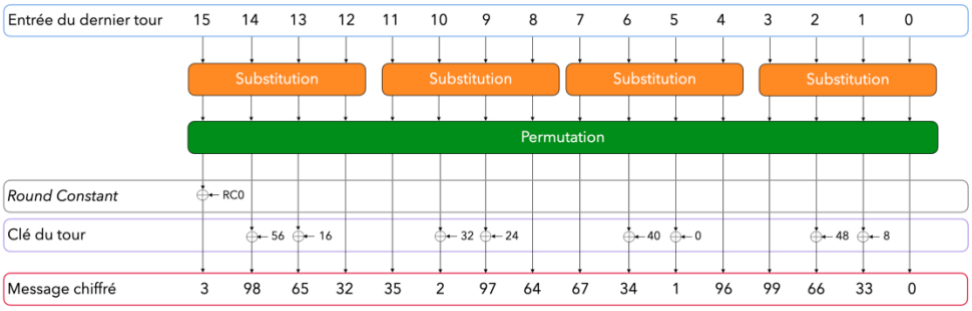


Figure 2-59 - 16 bits de poids faible du dernier tour de GIFT-128-128

Comme l'indiquent les Figure 2-56 et Figure 2-57, le modèle d'attaque de l'AES-128 et PRESENT est similaire : 8 bits de clé sont utilisés pour 8 bits du message chiffré. La principale différence réside dans le nombre de *SBOX* requises (couches de substitution). Pour rappel, celle de l'AES est définie sur 8 bits contrairement à celle de PRESENT sur 4 bits. Une attaque sur 8 bits de clé, requière donc deux *SBOX* pour PRESENT, contre une seule pour l'AES. Toutefois, l'intensité

des fuites est majoritairement liée à la taille de la couche de substitution en termes de ressources matérielles (*i.e. LUT*, *FF*). En conséquence, l'AES-128 se positionne comme étant le plus vulnérable de cette étude à l'attaque CEMA, y compris d'un facteur 2 par rapport à PRESENT-80, utilisant une taille de clé réduite.

En se focalisant désormais sur la cryptographie légère, il apparait cohérent que PRESENT-128 soit plus robuste que PRESENT-80 (d'environ 62%), du fait de leur différence de taille de clé. En effet, alors qu'il est nécessaire d'attaquer deux tours pour PRESENT-80, la version 128 en requiert trois. De plus, le nombre de bits à corréler se trouve réduit pour PRESENT-80 (64 bits, puis 16 bits de clé), par rapport à PRESENT-128 (64, 61 et 3 bits de clé respectivement). De ce fait, ce dernier exige plus de traces pour une attaque réussie.

Comme pour PRESENT, l'opération de substitution de GIFT est définie sur 4 bits. Expérimentalement, l'intensité des fuites entre ces deux algorithmes est similaire, avec un niveau légèrement supérieur pour GIFT-128-128. Cela corrobore les précédents résultats de tailles d'implémentation vus en section 2.3. Alors que l'AES et PRESENT appliquent respectivement quatre et huit bits de clé par *SBOX*, GIFT n'en utilise que deux, comme le montrent les Figure 2-52 et Figure 2-53. Les hypothèses de sous-clé ne sont alors réalisées que sur la moitié des données, contrairement à PRESENT et l'AES. La partie restante se comporte comme une constante, réduisant ainsi le nombre de fausses hypothèses. De plus, cela signifie que quatre *SBOX*, soit 16 bits de données au total (dont certains bits du *Round Constant*), sont nécessaires pour attaquer 8 bits de clé. Cette structure rend le processus de corrélation particulièrement efficace en augmentant le rapport signal sur bruit. Par conséquent, GIFT se trouve jusqu'à près de 60% plus vulnérable que PRESENT.

Deux versions de GIFT sont disponibles, utilisant chacune une clé de 128 bits, pour une donnée à chiffrer de 64 ou 128 bits. Bien que l'exploitation de leurs *SBOX* soit identique (*cf.* Figure 2-58 et Figure 2-59), GIFT-64-128 montre une plus faible propagation de sa clé pendant le chiffrement, ce qui nécessite quatre tours à attaquer (*cf.* Figure 2-52). Cette propriété, qui apparait au premier abord comme une force, est en réalité responsable d'une perte de robustesse d'environ 20% par rapport à son homologue. En effet, alors que l'attaque CEMA s'affaire à retrouver 64 bits de clé pour chacun des deux derniers tours de GIFT-128-128, elle ne cible que 32 bits pour ceux de GIFT-64-128. Étant donné qu'ils partagent les mêmes opérateurs, notamment les *SBOX*, il en résulte une probabilité d'échec divisée par deux pour chaque tour, améliorant ainsi l'attaque sur GIFT-64-128.

D'un point de vue pratique, le temps d'acquisition unitaire est compris entre quelques centièmes de secondes (AES-128, PRESENT-80/128 et GIFT-64-128), jusqu'à environ une seconde pour GIFT-128-128. Toutefois, pour effectuer les quelques milliers de traces, avec notamment le

moyennage de six chiffrements par message, le temps d'acquisition se compte en heures pour chaque algorithme. Quant à la durée de l'attaque, sa variation est beaucoup plus prononcée suivant l'algorithme, comme le révèle le Tableau 2-13.

Tableau 2-13 - Temps d'exécution de l'attaque CEMA

Algorithmes	Nombre de tours à attaquer	Nombre de traces	Durée minimale de l'attaque (s)	Durée de l'attaque sur 3 400 traces (s)
AES-128	1	1 100	1,23	3,60
PRESENT-80	2	2 100	8,22	10,73
PRESENT-128	3	3 400	16,42	16,42
GIFT-64-128	4	1 400	4,31	11,57
GIFT-128-128	2	1 800	7,09	11,21

Le temps de corrélation est obtenu sous *MATLAB R2017b* [102], exécuté sur un processeur *Intel Core i7* avec 16 Go de mémoire *DDR3*. Il convient de préciser qu'il dépend principalement du nombre de traces à corréler, mais également de l'échantillonnage de ces dernières (*i.e.* le nombre de points). Bien que ces résultats puissent varier d'une plateforme à l'autre, l'objectif principal est de fournir un ordre de grandeur du temps d'attaque. D'après le Tableau 2-13, il ne suffit que de quelques secondes pour mettre à mal ces algorithmes (sans compter le temps d'acquisition). Étant donné le faible nombre de traces, de points et de tours ciblés, le temps d'attaque de l'AES-128 se montre être le plus faible, soulignant sa véritable vulnérabilité à l'attaque CEMA. En raison de leur approche au niveau binaire, PRESENT et GIFT impliquent des procédures de formatage de données, allongeant le temps d'attaque. Ainsi, la cryptographie légère étudiée dans cette thèse nécessite en moyenne quatre fois plus de temps que l'AES-128.

2.4.6. Synthèse

Après avoir étudié les gains en termes de surface, d'énergie et d'efficacité offerts par la cryptographie légère, l'objectif de cette section 2.4 est de compléter l'évaluation en se focalisant sur l'aspect sécurité. Plus précisément, les attaques par canaux auxiliaires sont reconnues aujourd'hui comme étant une réelle menace. L'analyse du rayonnement électromagnétique offre à la fois simplicité, faible cout et efficacité, ce qui la rend abordable pour un grand nombre d'acteurs et lui vaut de figurer parmi les plus dangereuses. De plus, de par leur nature, les objets connectés sont également peu couteux et facilement accessibles physiquement, ce qui les rend d'autant plus vulnérables. C'est pourquoi il est important de prendre en considération la robustesse des algorithmes face à ce type d'attaque.

La mise en place de l'évaluation requiert plusieurs étapes tels que la sélection de l'opération cryptographique cible (les *SBOX*), le modèle de fuite (la distance de Hamming, en raison de sa cohérence avec le fonctionnement d'un inverseur CMOS), ainsi que d'un distingueur (la CEMA [98]). Côté implémentations, le cœur cryptographique est isolé du système de contrôle (*i.e.* UART, machine d'état, etc.), afin d'éviter toutes perturbations. De plus, pour permettre l'emploi de la distance de Hamming, une légère modification de l'architecture est effectuée (contrôle d'écriture du registre toujours actif). Enfin, l'analyse du processus de génération des sous-clés des algorithmes de chiffrement permet de déterminer le nombre de tour nécessaire à attaquer, afin de retrouver la clé principale. Alors que l'AES-128 n'en requiert qu'un seul, PRESENT-80 et GIFT-128-128 en exigent deux, PRESENT-128 trois, et GIFT-64-128 quatre.

L'évaluation de robustesse consiste à quantifier le nombre de traces nécessaires pour retrouver la clé principale de chiffrement (*Success Rate*). Les résultats mettent en avant la vulnérabilité de l'AES-128, faisant définitivement de lui, une solution inadaptée au contexte de l'IdO. GIFT, qui jusque-là montrait un fort intérêt en termes de taille et d'efficacité, se montre lui aussi peu résistant contre l'attaque CEMA. Sa version 128 bits de données augmente la résistance d'environ 29%. La faiblesse de GIFT provient principalement de son utilisation de la clé. Malgré sa taille de 128 bits, seulement un quart (ou la moitié suivant la version) n'est utilisé à chaque tour, réduisant fortement le mélange avec les données, en augmentant la corrélation de l'attaque CEMA. Enfin, PRESENT se montre le plus résistant. Bien que sa version 80 bits de clé ne soit guère plus robuste que GIFT, PRESENT-128 quant à lui l'est de près de 2 fois plus que GIFT-128-128 et de 3 plus que l'AES-128.

2.5. Conclusion

De par l'expansion des objets connectés, leur omniprésence les conduit à traiter de plus en plus de données, pouvant être sensibles (*e.g.* données médicales, compte bancaire, position géographique, etc.). Cela fait d'eux des cibles de choix dans le but d'extorquer des informations. La notion de confidentialité devient ainsi un enjeu majeur pour l'IdO, et la cryptographie permet de répondre à cette problématique. L'AES constitue le standard des chiffrements symétriques par bloc, mais n'est pas conçu pour être embarqué dans des systèmes fortement contraints, tels que les objets connectés. En conséquence, des algorithmes de chiffrements plus légers ont vu le jour, comme PRESENT et GIFT. Il est toutefois légitime de se demander dans quelles mesures ces derniers sont-ils plus adaptés au contexte de l'IdO, et quelle sécurité apportent-ils.

Ce premier chapitre propose une évaluation de ces trois algorithmes, suivant leur implémentation matérielle, afin de comparer leur taille, leur efficacité et leur empreinte énergétique. Les résultats démontrent un net clivage entre l'AES et ses homologues légers d'un facteur 10 en termes de taille

d'implémentation. De manière générale, GIFT atteint les meilleures performances. La version GIFT-64-128 offre la plus faible taille d'implémentation de cette étude, jusqu'à 19% de moins que PRESENT-128, exploitant une même taille de clé. Il en est de même pour l'efficacité, avec un gain de 26% par rapport à ce dernier. Par ailleurs, GIFT-64-128 permet d'atteindre un gain jusqu'à 10 fois supérieur à celui de l'AES-128. Enfin, la version GIFT-128-128 atteint la meilleure efficacité énergétique, jusqu'à plus de 28% par rapport à GIFT-64-128. Il permet également une implémentation 6 fois plus compacte que celle de l'AES-128. Ainsi, ces différents résultats positionnent GIFT comme une prometteuse alternative pour sécuriser les objets fortement contraints, tout en optimisant l'utilisation des ressources.

L'aspect robustesse est également pris en compte, en évaluant la résistance de ces algorithmes face aux attaques par canaux auxiliaires. L'étude se concentre notamment sur l'analyse du rayonnement électromagnétique via la méthode CEMA. Les résultats mettent en avant la vulnérabilité de l'AES-128, le positionnant comme le moins robuste. Alors que GIFT montre d'excellents résultats d'implémentation, il est cependant lui aussi peu résistant à l'attaque CEMA. Enfin, PRESENT, et notamment sa version PRESENT-128 est la plus robuste de l'évaluation, de près d'un facteur 2 et 3 par rapport à GIFT-128-128 et à l'AES-128 respectivement.

3. Sécuriser un écosystème décentralisé d'objets connectés

3.1. Contexte

L'objectif d'un réseau IdO est de connecter divers systèmes entre eux afin d'échanger des données et des ressources dans le but de créer des services. Cette hyperconnectivité entraine de nombreux défis sécuritaires, accrue par l'adoption d'une topologie réseau décentralisée, plus adaptée au contexte de l'IdO qu'une architecture centralisée. Le chapitre précédent a permis de souligner l'intérêt de la cryptographie légère pour assurer la confidentialité des données circulant sur le réseau. Toutefois, cette dernière ne suffit pas à elle seule à assurer une sécurité totale. En effet, son champ d'action ne concerne que les interactions entre les objets, c'est-à-dire la confidentialité, l'intégrité, l'authenticité ainsi que la non-répudiation des données. Bien que la cryptographie soit indispensable pour sécuriser les échanges, la viabilité d'un réseau d'objets connectés repose sur la fiabilité des services qu'il propose. Cela implique des mécanismes de contrôle sur la pertinence des données et sur la capacité du réseau à répondre aux exigences d'utilisation (*e.g.* temps réel). De plus, étant dans une architecture décentralisée sans organisme central de contrôle, le réseau doit être autonome. Cela signifie qu'il doit s'auto-adapter afin d'optimiser ses performances, mais aussi de faire face aux potentielles cyberattaques. Le tout, en prenant en considération les faibles ressources de calcul, de mémoire et d'énergie dont disposent les objets. Pour mener à bien cette mission, ces derniers doivent alors agir de concert, afin de ne former qu'une seule et unique entité. En d'autres termes, il est primordial d'instaurer la confiance entre les objets. Pour répondre à cet enjeu de taille, ce chapitre se focalise sur les deux grandes approches présentes dans la littérature, à savoir les systèmes de confiance et de réputation, et les systèmes blockchains.

3.2. État de l'art

3.2.1. Système de confiance et de réputation dans le domaine de l'IdO

En raison de la volatilité de l'IdO (multiples connexions/déconnexions) mais aussi de grand nombre d'objets présents, ces derniers n'ont a priori pas ou peu d'informations sur les autres. Suivant le vocabulaire humain, les objets sont des inconnus les uns par rapport aux autres. De ce fait, un mécanisme simulant la notion de confiance permet d'assurer la fiabilité, la performance, mais aussi la sécurité d'un système, en détectant et isolant les objets non viables, défectueux, voire malveillants. De manière plus concrète, un tel mécanisme peut être utilisé pour sélectionner un objet ayant des services plus performants et plus sécurisé qu'un autre [103] ou encore définir un routage réseau pair à pair (*e.g.* Ad Hoc) n'utilisant que les nœuds de confiance [104]. Ainsi, en instaurant la confiance,

il est alors possible de continuellement évaluer et améliorer l'efficacité et la sécurité d'un écosystème, à partir de multiples observations effectuées au cours du temps.

a) Concept de confiance

La notion de confiance a largement été étudiée comme le montrent les travaux [103]–[113]. Globalement, la confiance est un concept de haut niveau, regroupant à la fois l'aspect comportemental et la notion de sécurité. Par ailleurs, l'étude [105] stipule que la sécurité peut être considérée comme l'aspect technique (*i.e.* confidentialité, authenticité et intégrité), tandis que la confiance renvoie à une vision plus large, en prenant en compte le comportement des entités, la qualité du service, etc. Il s'agit là d'une notion complexe, influencée par de nombreuses propriétés mesurables et non mesurables. Cependant, il est possible de la définir comme l'attente qu'une entité accomplisse une tâche comme prévu. Pour appréhender correctement ce concept, il est important de souligner certaines de ses caractéristiques :

- La confiance est dynamique et dépend du contexte : elle n'est acquise que pour une certaine durée et dans une certaine situation.
- La confiance n'est pas transitive : si une entité *X* a confiance en une entité *Y*, qui a confiance en une autre entité *Z*, alors *X* ne fait pas nécessairement confiance en *Z*.
- La confiance est subjective et n'est pas obligatoirement réciproque : *X* peut avoir confiance en *Y* mais *Y* peut ne pas avoir confiance en *X* alors que *Z* fait confiance à *X* mais pas à *Y*.
- La confiance dépend de l'expérience : la qualité de la confiance varie en fonction des interactions en cours et passées entre les entités.

Trois sources d'information sont requises pour évaluer la confiance : la connaissance (*i.e.* sa propre compréhension d'une entité), l'expérience (*i.e.* interactions avec une entité) et la réputation (*i.e.* l'expérience acquise par d'autres entités). Dans leurs travaux [105], les auteurs définissent la connaissance comme la « confiance directe », ce qui signifie qu'aucune interaction n'est nécessaire pour évaluer la confiance d'une entité/objet. En d'autres termes, les caractéristiques intrinsèques sont utilisées, comme par exemple celles fournies par le constructeur d'un objet (*e.g.* précision des mesures, consommation énergétique ...). Toutefois, cette approche requiert une base de données commune et accessible à tous, ce qui repose fortement sur la bonne foi du constructeur. De plus, ce modèle ne tient pas compte du comportement des objets sur le réseau, c'est-à-dire l'expérience et leur réputation, définies en tant que « confiance indirecte » d'après [105]. Il s'agit de l'auto-jugement formé à partir d'observations au cours de multiples interactions avec l'objet. Ce modèle permet de compléter le précédent en apportant des informations supplémentaires sur les entités du réseau. Toutefois, il est important de noter que ces définitions de confiance ne sont pas unanimes dans la littérature. En effet,

la définition de « confiance directe » dans [105] est désignée comme la « confiance dans les capacités » dans l'étude [106], c'est-à-dire qu'une entité est évaluée en fonction des ressources dont elle dispose (*e.g.* puissance de calcul, stockage, etc.), lui permettant d'accomplir une tâche. Quant aux travaux [107]–[111], il est question d'évaluation basée uniquement sur les propres interactions d'un objet avec un autre (désignée comme l'expérience dans [105]). À l'opposé, la « confiance indirecte » ou « confiance dérivée » dans [112], fait référence à la notion de réputation, c'est-à-dire le niveau de confiance attribué à une entité, du point de vue d'une autre. Ce clivage rien qu'au niveau des définitions, souligne clairement le manque et la difficulté de standardisation du concept de confiance. Afin d'éviter toute confusion dans la suite des travaux de cette thèse, le terme de « confiance directe » fait référence aux interactions entre les objets, à l'opposé de la « confiance indirecte » désignant les recommandations provenant des autres objets.

b) Calculer le niveau de confiance : modèles et propriétés

La pertinence d'un système de confiance réside dans la sélection des meilleures métriques pour évaluer la confiance, ainsi qu'une méthode d'agrégation adaptée afin de convertir ces dernières en une valeur unique d'estimation globale du niveau de confiance attribué. En cumulant les précédences informations d'un appareil sur une période de temps donnée, le niveau de confiance que l'on peut lui attribuer peut être calculé, permettant à un autre de prendre une décision vis-à-vis dudit objet. Cette opération requiert ce qui est communément appelé un « modèle de confiance » (*Trust Model*). Les auteurs de l'étude [113] soulignent les principaux modèles utilisés tels que la somme pondérée, les modèles bayésiens ou encore la logique floue (*Fuzzy*), dont les valeurs des variables logiques sont comprises dans l'intervalle [0,1] contrairement à la logique booléenne (0 ou 1). Cependant, bien que le rôle du modèle soit important, la partie fondamentale réside dans les propriétés utilisées pour calculer la confiance. En raison de la large et contextuelle définition de cette dernière, chaque étude adopte une approche différente et emploi de multiples terminologies, pour parfois définir le même concept. Dans leur ouvrage [108], Bao et Chen sont les premiers à proposer une relation sociale pour la gestion de confiance dans le contexte de l'IdO. Ils définissent en tant que propriétés principales, les notions d'honnêteté (*i.e.* croyance de *X* que *Y* est honnête sur la base des interactions), de coopérativité (*i.e.* les objets amis sont probablement plus coopératifs que les autres) et d'intérêt communautaire (*i.e.* intérêts et objectifs communs). Bien que Bao et Chen proposent des lignes directrices pour calculer leurs métriques (*e.g.* compter le nombre d'interactions ne respectant pas les règles, pour la notion d'honnêteté), ce type d'approche basée sur les relations sociales reste pour le moins complexe à évaluer, pénalisant le modèle d'un manque d'efficacité et de fiabilité. Par conséquent, d'autres travaux de recherche se focalisent sur des métriques plus facilement quantifiables comme la taille des paquets réseau [109], le délai de transmission [109], l'énergie [111],

la qualité du signal radio [111] ou encore le taux de participation d'un objet dans son écosystème (*e.g.* nombre de tâches accomplies).

c) Vers un besoin de stockage sécurisé

Comme indiqué précédemment, l'évaluation de la confiance repose sur les informations relatives des objets connectés sur une certaine période, mais aussi sur l'historique des interactions passées. Cela nécessite un besoin d'espace de stockage dédié. Il devient évident que ce dernier doit être robuste et sécurisé, c'est-à-dire infalsifiable, afin d'assurer la fiabilité de l'évaluation de la confiance. En effet, la modification, volontaire ou non, d'une information peut fortement impacter l'évaluation du niveau de confiance, rendant ainsi le système inefficace, manipulable et donc non sécurisé. Il convient d'avoir un mécanisme permettant la vérification, l'authentification et la non-falsification des informations. En d'autres termes, la collecte des données doit être sécurisée. Cela est d'autant plus vrai dans une topologie réseau décentralisée tel que l'IdO sans organisme central, où le risque de manipulation est amplifié. Ce besoin de stockage sécurisé, étant l'une des principales limitations des systèmes de confiances traditionnels, constitue la première motivation d'allier ces derniers aux systèmes blockchains. Cette association a donnée naissance au concept de *Blockchain-based Trust Management*.

3.2.2. Introduction à la blockchain

a) Principe de la blockchain

Le concept de la blockchain peut a priori s'apparenter à un système complexe dont les rouages sont flous et difficiles à appréhender, telle que l'était la notion de Cloud lors de sa démocratisation. Pourtant, il n'en est rien. Dans sa définition la plus généraliste, la blockchain est un système de stockage décentralisé, qui ne repose sur aucune entité centrale. En d'autres termes, il s'agit d'une base de données répliquée sur un ensemble d'entités connectées à un réseau. Puisqu'il n'y a pas d'organisme central (*e.g.* administrateur), ce sont ces entités qui sont en charge de maintenir le réseau, et la base de données. Afin d'imprégner le lecteur du concept, mais aussi de l'utilité de la blockchain, cette section propose d'aborder successivement ses différents points clés, tout au long du scénario suivant : soit Alice et Bob, deux personnes souhaitant s'échanger de l'argent. Dans le système classique tel qu'illustré sur la Figure 3-1, lorsque Alice souhaite envoyer de l'argent à Bob, elle déclenche un ordre de transfert à sa banque, qui joue le rôle d'intermédiaire. Cette dernière s'assure d'une part qu'Alice possède bien la quantité d'argent qu'elle souhaite envoyer, et vérifie l'identité de Bob. Après quoi, la banque effectue le virement sur le compte de Bob. Ce scénario schématisé met en avant le caractère centralisé d'un tel écosystème, où la banque se présente comme l'autorité centrale, en charge du contrôle et de la sécurité du réseau. Du point de vue de l'utilisateur, celui-ci est

dans l'obligation de faire confiance à cet organisme central. À l'inverse, ce dernier est en mesure de contrôler toutes les informations circulant sur le réseau. Mais plus intéressant encore, si dans ce scénario la banque n'est plus en mesure de fonctionner correctement (*e.g.* saturation du réseau, attaques ...), tout le système entier est mis à mal. Cela peut se traduire par un arrêt de toute activité ou de vol d'informations confidentielles.

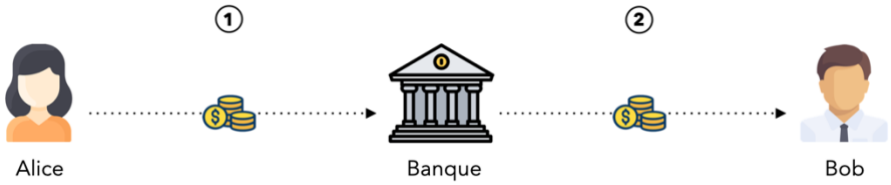


Figure 3-1 - Exemple de transfert d'argent centralisé

Dans le but de remédier aux problèmes engendrés par cette approche centralisée, l'énigmatique Satoshi Nakamoto propose en 2008 un concept de paiement décentralisé, basé sur la blockchain [114]. L'objectif consiste à rendre possibles des paiements directement de particulier à particulier, sans avoir recours à un intermédiaire de confiance, comme une banque. Bien que ce principe soit facilement compréhensible, il n'en soulève pas moins de nombreux défis sécuritaires. En effet, puisque la banque est mise à l'écart, il n'y a plus d'organisme en charge de contrôler les transferts. En reprenant l'exemple précédent, comment Bob s'assure qu'Alice possède bien l'argent qu'elle compte lui transférer ? Comment certifier que le transfert a bel et bien était effectué ? C'est pour répondre à ces problématiques que la blockchain s'avère être un atout de taille. La Figure 3-2 reprend l'exemple précédent dans ce nouvel écosystème décentralisé avec désormais deux autres participants Carole et Dave.

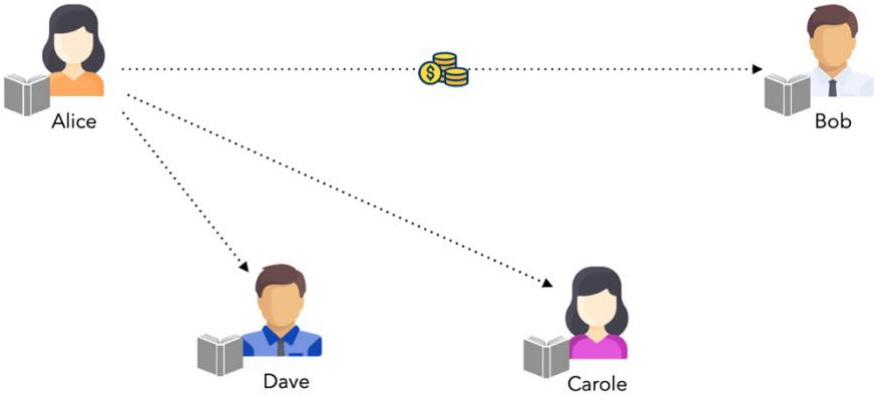


Figure 3-2 – Exemple 1 de transfert d'argent décentralisé utilisant la blockchain

La manière la plus simple d'appréhender la blockchain, est de l'assimiler à un livre de comptes, dans lequel chaque transfert y est enregistré. En raison de l'absence d'organisme central, chaque participant possède une copie de ce livre, qu'il est en charge de maintenir à jour. Lorsque Alice souhaite effectuer son transfert, elle en informe le réseau. Bob, Carole et Dave consultent alors leur livre de comptes afin de vérifier si Alice possède bien les fonds. Si tel est le cas, chaque participant l'indique dans sa propre copie du livre (*i.e.* le compte d'Alice est débité et celui de Bob crédité). Le transfert est désormais acté et irréfutable de manière décentralisée. De cette manière, Alice ne peut pas tricher, en faisant croire aux autres membres qu'elle dispose toujours des fonds. Par exemple en effaçant le transfert de sa propre copie du livre. Bob, Carole, et Dave auront toujours la trace du transfert d'Alice dans leur copie. C'est ce qui est communément appelé le problème de la double dépense que la blockchain permet de résoudre.

b) Structure et composition de la blockchain

Pour la suite, il convient d'étoffer le scénario précédent : supposons désormais que seule Alice possède initialement une quantité d'argent, et qu'elle souhaite effectuer un transfert à Bob. Ce dernier redistribue ensuite une partie des fonds reçus d'Alice, à Carole, qui transfère à son tour une partie à Dave, comme illustré par la Figure 3-3.

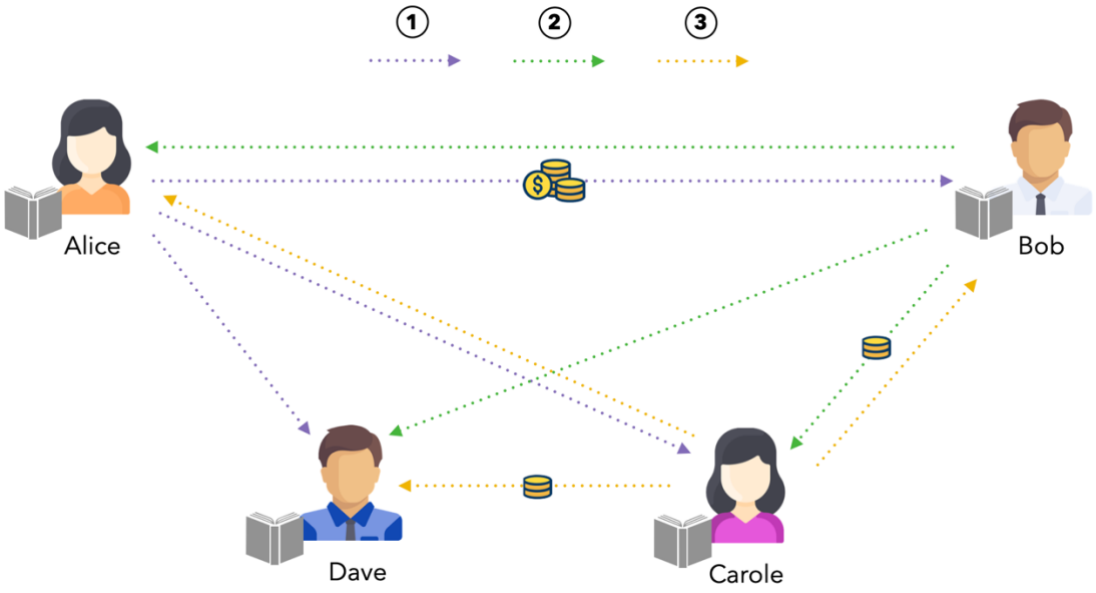


Figure 3-3 - Exemple 2 de transfert d'argent décentralisé utilisant la blockchain

Ce simple enchainement souligne une problématique indispensable à résoudre, à savoir l'ordre des transferts (*i.e.* des données). En effet, puisque Dave reçoit de l'argent de Carole, il faut s'assurer que cette dernière possède bien les fonds. Cela signifie qu'il faut d'une part prendre en compte le

transfert de Bob à Carole, mais également celui de Alice à Bob. Pour répondre à ce besoin, la blockchain est composée, comme son nom l'indique, de blocs, liés cryptographiquement les uns aux autres, comme illustrés sur la Figure 3-4.

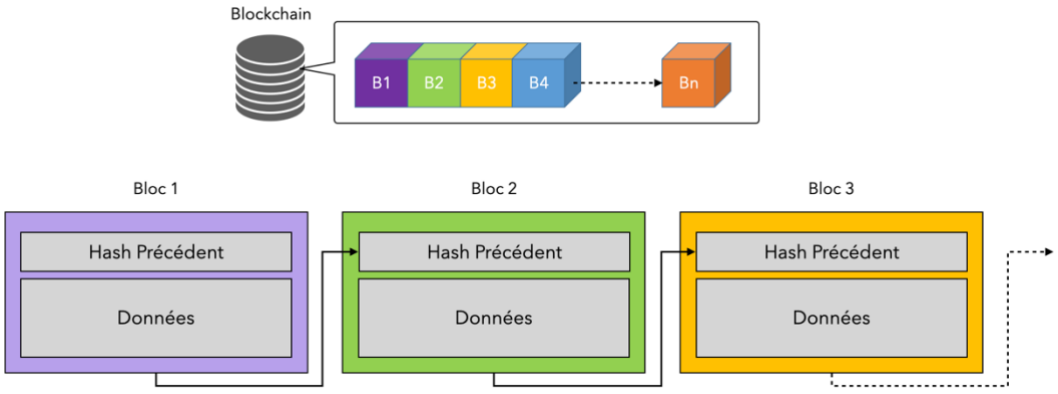


Figure 3-4 - Structure de la blockchain

Bien que la composition d'un bloc soit directement liée aux besoins de l'application, certains éléments sont indispensables pour construire un système blockchain comme il se doit. Puisque son rôle est de sécuriser la sauvegarde des données utiles de l'application, il va sans dire que ces dernières composent majoritairement la blockchain. Mais l'élément qui est au cœur du concept de la blockchain est l'inclusion de « l'empreinte » (*i.e.* hash) du bloc précédent dans le suivant. Brièvement, une fonction de hachage est un algorithme cryptographique permettant de générer une empreinte numérique unique d'une donnée, de sorte que la modification de seulement un bit de cette dernière, entraine une empreinte totalement différente. Ce principe assure ainsi l'intégrité de la donnée initiale. L'intérêt d'une telle fonction dans la construction d'une blockchain prend tout son sens, lorsqu'il est question d'ordonner les données. En effet, puisque chaque bloc contient les informations de l'application à sécuriser, il est souhaitable de générer une empreinte unique de ce bloc, afin de s'assurer aucune modification sur les données ne soit possible. En ajoutant cette empreinte unique dans le bloc suivant, l'empreinte de ce dernier dépend alors non seulement de ses propres données, mais aussi de l'empreinte du bloc précédent (et par conséquent de ses données). Il apparait alors un fort lien cryptographique entre les blocs, assurant l'inviolabilité des leurs données. Afin de mettre en image ce mécanisme, la Figure 3-5.a) présente la construction de la blockchain du scénario de transfert d'argent d'Alice, Bob, Carole de Dave. La Figure 3-5.b) met en avant la conséquence d'une modification de la donnée (*e.g.* changement de la valeur de transfert de Bob à Carole), alors que la Figure 3-5.c) souligne la conséquence d'un changement d'ordre des blocs (*e.g.* transfert de Carole à Dave avant celui de Bob à Carole). Ainsi, grâce à ce lien cryptographique entre les blocs, il est impossible de modifier ou même de supprimer une information inscrite dans la blockchain, sous peine d'impacter l'ensemble de la chaine.

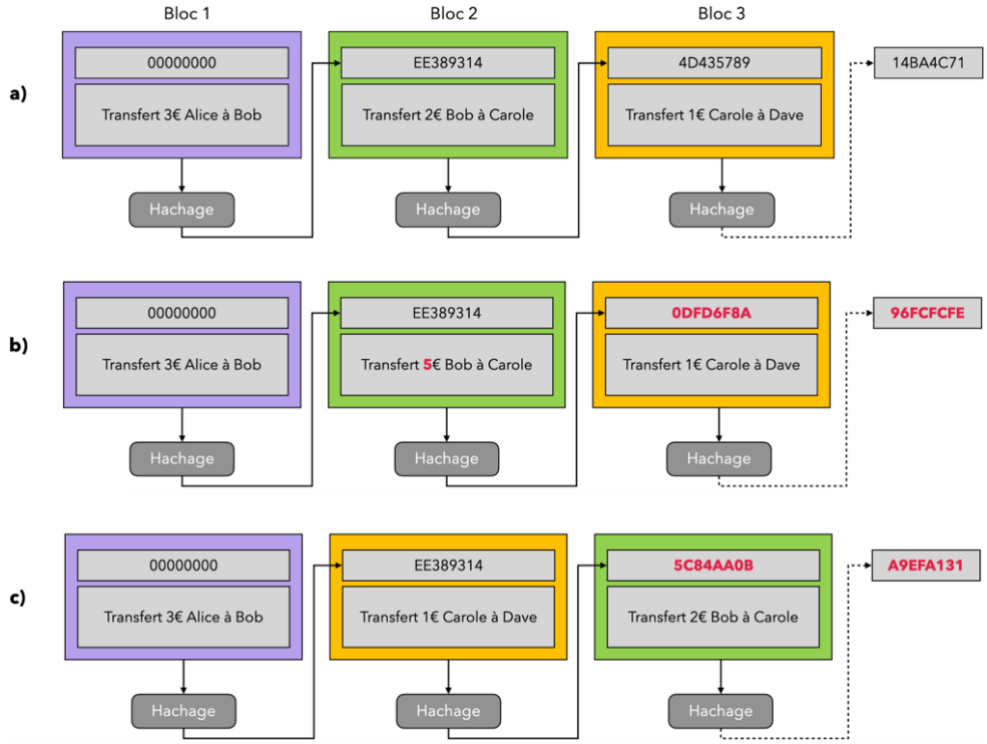


Figure 3-5 - Lien cryptographique entre les blocs

c) Consensus, une harmonisation et sécurisation de la blockchain

Sans mécanisme supplémentaire, la blockchain apparait comme étant non sécurisée, car la modification/suppression des données nécessite uniquement de recalculer les empreintes de chaque bloc consécutif. Il est vrai que le temps requis pour cette opération dépend du nombre de blocs à recalculer et donc de la taille de la blockchain. Toutefois, les algorithmes de hachage, utilisés pour la génération d'empreintes, requièrent peu de ressources de calcul, pouvant même être embarqués dans les objets connectés [115]. Cela ne suffit donc pas à assurer la sécurité de la blockchain.

D'autre part, étant donné qu'il n'y a pas d'autorité centrale, ce sont les participants du réseau eux-mêmes qui assurent la maintenance du système. Cela signifie que chacun d'entre eux est en mesure de valider ou d'invalider une nouvelle donnée (*i.e.* bloc) à ajouter à la blockchain. Telle que nous l'avons décrite, la mise en place réelle de cette blockchain serait soumise à plusieurs difficultés :

Coordination : tout d'abord, il n'y aurait aucune coopération entre les participants, c'est-à-dire que chacun mettrait à jour la version de sa blockchain (*cf.* livre de comptes), sans consulter les autres. Il faut donc un mécanisme permettant une coopération les différents acteurs afin de synchroniser leurs informations.

Latence : un réseau informatique étant imparfait, il existe une latence, c'est-à-dire un délai entre le moment de l'émission et de la réception d'une information. Ce retard fluctue en fonction de nombreux éléments telles que la position géographique de l'émetteur et du récepteur, de l'activité du réseau, etc. De ce fait, il est hautement probable que plusieurs entités réceptionnent des informations dans un ordre différent. Or, comme vu précédemment, l'ordre des données dans la blockchain est primordial (*cf.* Figure 3-5). Enfin, il est également possible qu'une information soit perdue et n'atteigne pas certaines entités. Par conséquent, chaque participant se retrouve avec une version différente de la blockchain.

Efficacité : un troisième point concerne l'efficacité des participants à se coordonner. Cela implique de mettre en place une méthode à la fois robuste et dont le processus de vérification des données soit efficace.

Pour doter la blockchain de telles capacités, il convient d'y intégrer ce qu'on appelle un algorithme de consensus. Ce dernier est en charge de coordonner chaque participant du réseau, afin que tous aboutissent à une même et unique décision commune. En apportant cette cohésion, il est alors possible de se mettre d'accord unanimement sur le prochain bloc à ajouter. Si l'on reprend le scénario précédent d'Alice et ses partenaires (*cf.* Figure 3-3) celle-ci informe les autres de son transfert à Bob, mais cette fois, Dave et Carole doivent donner leur accord, sans quoi le transfert n'est pas autorisé, ainsi que les suivants. Après autorisation, un bloc est créé contenant le virement d'Alice à Bob. Aussi séduisante que puisse être cette approche, elle s'avère inefficace puisque chaque transfert nécessite une autorisation du réseau et la création d'un bloc. Pour remédier à ce problème, une solution consiste à inclure dans un seul et même bloc plusieurs transferts. Dans notre scénario, les virements d'Alice à Bob, de Bob à Carole puis de Carol à Dave sont dépendants les uns des autres, par conséquent il est judicieux de les regrouper dans un seul bloc et de valider l'ensemble, comme illustré par la Figure 3-6. Ainsi, un seul accord global est effectué par les participants au lieu de trois initialement.

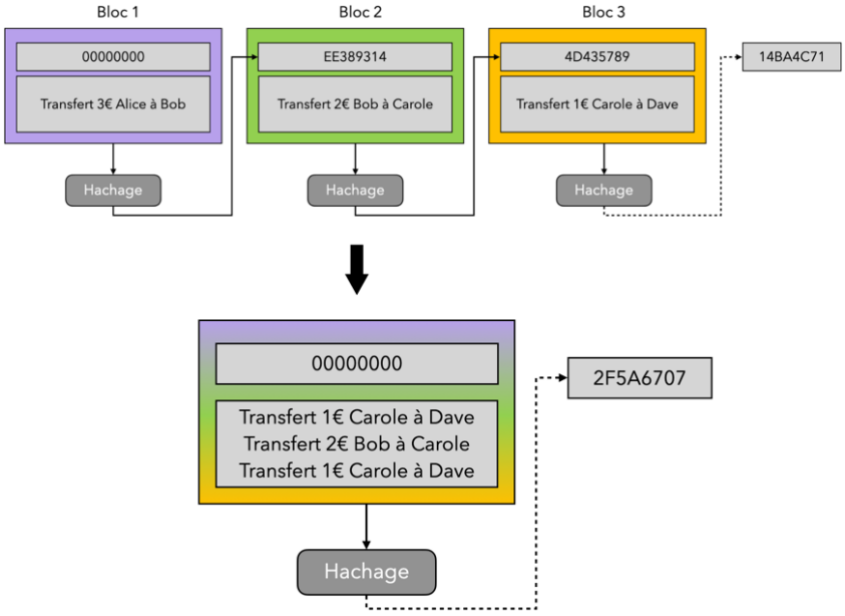


Figure 3-6 - Regroupement de données dans un seul bloc

En poursuivant l'analyse du mécanisme de consensus précédent, un second problème apparait : si l'un des participants n'est plus en mesure de donner son accord (*e.g.* perte de connexion, ou acte malveillant) alors tout le système s'arrête. Afin de pallier à ce type de scénario, l'algorithme de consensus dispose de règles, autorisant non plus à un accord unanime des participants, mais à une majorité d'entre eux. Par exemple, si Bob subit momentanément une perte de connexion, Alice, Dave et Carole peuvent toujours assurer la maintenance du système. Ce mécanisme permet de limiter l'impact de la latence du réseau (*i.e.* délai de transmission et/ou perte de l'information) et offre au système une robustesse face aux divers dysfonctionnements et attaques, dues à l'indisponibilité de certaines entités.

Il convient de noter qu'à première vue, un accord par la majorité semble moins sécurisé qu'un accord à l'unanimité. Pour limiter cet effet, une nouvelle règle dans l'algorithme de consensus introduit la notion de complexité dans la validation des blocs, exigeant aux participants de s'impliquer dans le processus. L'objectif sous-jacent est de dissuader les comportements malhonnêtes susceptibles de dégrader la sécurité du système. La manière la plus simple d'appréhender ce concept dans le scénario précédent est d'obliger les participants à apposer leur signature « bon pour accord » dans un bloc à ajouter. Grâce à cela, chaque entité est capable de déterminer de manière générale si un bloc a été approuvé majoritairement, mais aussi d'avoir la liste des participants l'ayant signé. Il est alors possible de promouvoir ceux qui ont donné leur accord sur des données viables, et à l'inverse de punir ceux ayant tentés de corrompre le système. Cette méthode d'incitation lie les participants dans un même but commun : assurer la pérennité du réseau.

Une des caractéristiques importantes de la blockchain est la transparence et la traçabilité des données stockées. Cela signifie que toute entité peut à tout moment réévaluer l'intégralité de la blockchain afin de s'assurer de sa validité. Cette propriété prend tout son sens lorsqu'un participant subit une perte de connexion (*cf.* exemple de Bob précédemment) ou encore lorsqu'un nouveau participant rejoint le réseau. Pour le cas de Bob, il doit synchroniser sa blockchain avec celle des autres, en demandant aux autres participants de lui transmettre les blocs manquants. Afin d'être certain de leur validité, Bob doit les contrôler. De la même manière, Ted souhaite pleinement intégrer le réseau et doit lui aussi se synchroniser avec les autres, c'est-à-dire mettre à jour sa copie de la blockchain, encore vierge. Une première approche consisterait à ne télécharger que les derniers blocs, comme pour Bob. Cependant, Ted ne dispose d'aucune information permettant de s'assurer de la validité des derniers blocs, qui dépendent eux-mêmes des précédents. Il apparait donc évident que Ted doit télécharger l'intégralité de la blockchain et la vérifier bloc par bloc. De ce fait, chaque bloc doit obligatoirement contenir sa preuve de validité. Dans le scénario d'Alice, cette preuve correspond aux signatures des participants, attestant leur « bon pour accord » sur les données d'un bloc. La composition finale de ce dernier correspond désormais à celle illustrée par la Figure 3-7.

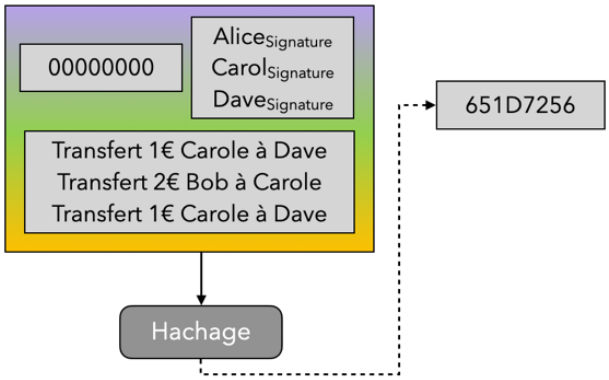


Figure 3-7 - Composition finale d'un bloc dans le scénario d'Alice

3.2.3. Les blockchains d'aujourd'hui

a) Une description plus formelle

De manière plus formelle, la blockchain est un registre distribué, fonctionnant sur un réseau pair à pair, où chaque entité possède un identifiant unique et s'échange des informations à l'aide de transactions. Ces dernières sont cryptographiquement signées par leur auteur, afin d'assurer leur authenticité (*i.e.* qui crée la transaction) et son intégrité (*i.e.* assurer son inviolabilité). Étant donné la nature du réseau, l'émetteur transmet sa transaction à ses voisins, avec qui il est directement connecté. Lorsque ces derniers la réceptionnent, ils s'assurent de la validité des informations avant de les

transmettre, le cas échéant, de la même manière à leurs proches homologues. Ce procédé se répète, autorisant ainsi uniquement la dissémination d'informations correctes sur le réseau. Toutes les transactions sont ensuite ordonnées et regroupées dans des blocs afin d'être définitivement actées par l'ensemble du réseau. Au travers d'un algorithme de consensus, les entités se mettent d'accord sur l'ordre et le contenu des blocs, à savoir les données, le lien cryptographique avec le bloc précédent et la preuve de validité. Disséminé de la même manière que les transactions, tout bloc considéré comme non valide est supprimé. Pour les autres, ils sont ajoutés à la suite des précédents, créant ainsi la blockchain.

Grâce à toutes ses caractéristiques, la blockchain s'avère être un système pertinent pour enregistrer et synchroniser des informations dans une base de données décentralisée, répliquée sur l'ensemble ou une partie des entités d'un réseau. Cette réplication offre à un tel système une plus grande robustesse face aux pannes et aux attaques en comparaison avec une approche centralisée. Par ailleurs, l'absence d'organisme central incombe aux participants eux-mêmes de maintenir la blockchain, apportant intrinsèquement transparence, traçabilité et sécurité au stockage de données. De manière générale la structure d'une blockchain et de ses blocs est résumée par la Figure 3-8

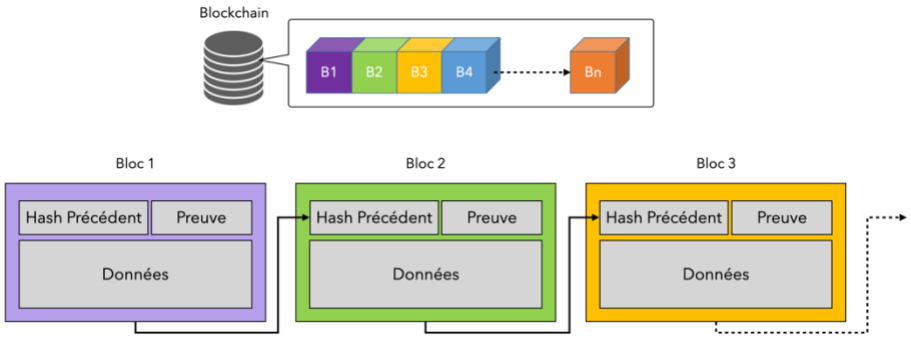


Figure 3-8 - Structure générale d'une blockchain et de ses blocs

Face à la démocratisation de la blockchain au cours de ces dernières années, une multitude de projets se sont lancés dans l'aventure. Bien qu'il ne soit pas d'un grand intérêt de les lister de manière exhaustive dans cette thèse, il est intéressant d'identifier les différents types de blockchains, leur structure ainsi que les différents acteurs.

b) Types de blockchains

Il existe plusieurs types de blockchains, directement liées à la nature des données, leur accessibilité et les actions qu'un utilisateur est en mesure d'effectuer. On parle alors de blockchain publique ou privée (*public* ou *private*), avec ou sans autorisation (*permissioned* ou *permissionless*), comme l'explique [116]. Les termes *public* et *private* font référence à l'accès des données d'une blockchain,

c'est-à-dire la manière dont les participants peuvent accéder au réseau. Une blockchain publique permet à quiconque d'obtenir un accès de manière inconditionnelle et sans aucune procédure d'acceptation. Bitcoin [114], Ethereum [117], IOTA [118] et Nano [119] sont des blockchains publiques. À l'opposé, les accès dans une blockchain privée, comme Hyperledger [120] et Quorum [121] sont soumis aux contrôles des propriétaires (*e.g.* les administrateurs). À l'inverse, *permissioned* et *permissionless* désignent les restrictions de certains membres à effectuer certaines tâches (*e.g.* création et validation des blocs). Bien que cela ne soit pas systématique, les blockchains privées usent souvent de ce type de restrictions. On parle alors de *private permissioned blockchain*. Un troisième type, nommé consortium, est également présent dans la littérature [116] [122], [123] et [124]. Ce concept est né en partant du postulat qu'une blockchain privée n'est constituée que d'une seule et unique organisation (*i.e. fully private blockchain*). Il convient de préciser que plusieurs entités peuvent être impliquées dans une telle blockchain, conservant l'approche distribuée, mais chacune d'elle est contrôlée par le même organisme. Le modèle consortium quant à lui, fait référence à une blockchain privée, ayant plusieurs organismes de contrôle. Par exemple, un groupe de plusieurs entreprises, dont chacune d'elles est en charge du maintien de la blockchain. Les projets Corda [125] et XDC [126] sont basés sur ce type de modèle.

c) Types de participants

Pour des raisons de simplicité d'approche, le terme « participant » a été employé jusqu'ici pour définir une entité connectée et interagissant sur un réseau blockchain. Il convient d'employer son équivalent plus formel : « nœud » (*node*), désignant un ordinateur ou tout autre périphérique capable de prendre part au réseau. Il en existe différents types, tous ayant des rôles bien précis comme les référencent [116] et [127]. À noter que d'un protocole à un autre, les noms attribués à ces différents types peuvent varier.

Les nœuds complets (*full nodes*) sont le cœur d'une blockchain, car ils supportent et assurent sa sécurité : ils maintiennent une copie complète de la blockchain et participent à la vérification et la propagation des transactions et des blocs. Attention, il est uniquement question de vérification de ces derniers, autorisant la propagation sur le réseau (*e.g.* la transaction et/ou le bloc sont-ils en accord avec les règles de consensus ou sont-ils frauduleux). Cela ne comprend donc pas la procédure de création de blocs. Cette dernière est assurée par les mineurs (*miner nodes*), parfois appelés validateurs (*validator nodes*) suivant l'algorithme de consensus employé. Ils sont identiques aux nœuds complets, à la différence qu'ils sont capables de générer des blocs afin de compléter la blockchain. Enfin, le troisième type correspond aux nœuds légers (*lightweight node*), ou parfois appelés simples nœuds (*simple node*) ou encore clients (*clients*). Comme leur nom l'indique, ces derniers sont de simples utilisateurs du réseau blockchain, dont leur rôle est de transmettre et recevoir des

transactions. Puisqu'ils ne possèdent pas de copie complète de la blockchain, ils ne participent pas à sa sécurisation ni à sa maintenance. Ils peuvent être vus comme de simples terminaux donnant accès aux informations. En conséquence, bien qu'il soit fortement intéressant d'avoir accès à des nœuds ne nécessitant que peu de ressources, la viabilité et les performances d'un système blockchain reposent entièrement sur ses nœuds complets, en charge de sa maintenance et de sa sécurité.

d) Bitcoin

Conceptualisé en 2008 et officiellement lancé le 3 janvier 2009 par Satoshi Nakamoto (celui dont on prononce le nom, mais dont on ne connait pas l'identité), le projet Bitcoin [114] constitue la première implémentation de la blockchain. Il est né d'une volonté de créer un moyen de paiement numérique pair à pair sécurisé et entièrement décentralisé. Ce concept de monnaie électronique, dont un jeton s'intitule *bitcoin*, portera le nom de « cryptomonnaie » (*cryptocurrency*), en raison de son utilisation intensive de fonctions cryptographiques (*e.g.* transaction, algorithme de consensus). La force d'un tel système, et plus particulièrement des propriétés offertes par la blockchain, réside dans la résolution d'un problème financier numérique de longue date : la double dépense (*double-spending*). Ce phénomène frauduleux consiste à dépenser plusieurs fois un même jeton numérique, par exemple en le dupliquant. Pour résoudre cette problématique, Bitcoin associe un algorithme de consensus avec une structure blockchain linéaire (*cf.* Figure 3-9), où une seule transaction du jeton numérique ne peut être présente (celle qui est acceptée par la majorité des nœuds).

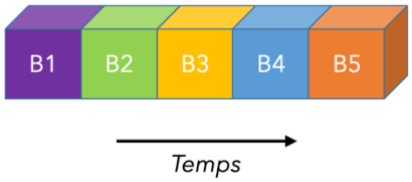


Figure 3-9 - Structure de blockchain linéaire (chaine unique)

Une autre caractéristique importante de Bitcoin concerne le format de transaction qu'il exploite, connu sous le sigle UTXO [128] (*unspent transaction outputs*) traduit par « transactions non dépensées », ou plus en adéquation avec notre langage, le « rendu de monnaie ». Par analogie avec un billet ou une pièce, une UTXO est indivisible et constitue une unité d'échange. Par exemple, soit un portefeuille contenant 25€ (un billet de vingt euros et un de cinq euros). Cela représente deux UTXO (une de 20€, et une autre de 5€). Lors de l'achat d'un objet valant 15€, il faudra utiliser l'UTXO de 20€, dont 10€ iront au vendeur, et les 5€ restants seront reversés dans le portefeuille de l'acheteur, sous la forme d'une nouvelle UTXO (ou billet) de 5€. Cette dernière pourra être utilisée

pour un nouvel achat. Ainsi, chaque UTXO est liée à une précédente, et le solde d'un compte est obtenu en effectuant la somme des UTXOs du portefeuille. Ce modèle de transaction permet d'effectuer plusieurs transactions en parallèle (autant de UTXO disponibles) et une forte traçabilité.

e) Ethereum

Ethereum [117] a été conçu en 2013 par Vitalik Buterin, un des développeurs de Bitcoin, dans le but de faciliter le développement d'applications décentralisées basées sur la blockchain. Ethereum possède comme Bitcoin sa propre cryptomonnaie, appelée *ether*, et est lui aussi construit sur une blockchain linéaire (chaine unique, *cf.* Figure 3-9). Mais à l'inverse, Ethereum adopte un modèle de compte/solde [128] (*Account/Balance model*) au lieu des UTXO. Son fonctionnement est similaire à celui des cartes bancaires que nous utilisons : un compte est associé à un solde, qui est contrôlé avant chaque transfert. Si le solde est suffisant alors l'opération est effectuée. En reprenant l'exemple précédent, la quantité d'argent disponible dans le portefeuille étant supérieure au prix d'achat (25€ > 15€) le transfert est autorisé, et ce, avec le montant exact de 15€. Ce modèle n'a plus besoin de conserver la liste complète des précédentes transactions, ce qui réduit la quantité de données à stocker. En revanche, puisqu'il faut au préalable vérifier le solde, seule une opération peut être effectuée à la fois.

Mais la grande particularité d'Ethereum réside dans sa capacité à exécuter du code sur la blockchain : les contrats intelligents (*smart contracts*). C'est en 1994 qu'ils sont introduits par Nick Szabo [129], qui les définit comme « un protocole de transaction informatisé qui exécute les termes d'un contrat ». De manière générale, ce sont des programmes informatiques, qui s'exécutent automatiquement lorsque les clauses préalablement définies sont remplies. Ils peuvent être vus comme une numérisation de n'importe quel type de contrat pouvant être passé entre des individus/entreprises, dans la mesure où cela peut faire l'objet d'une programmation informatique. Un *smart contract* peut prendre la forme de simples structure conditionnelles (*e.g.* « si les examinateurs valident ce travail, alors M. Dalmasso sera autorisé à soutenir sa thèse »), mais aussi des plus complexes, comme des contrats d'assurance (*e.g.* indemnisation automatique suivant des conditions préalablement définies), la mise en place d'un système de votes pour une élection, ou encore de la location de particulier à particulier. Les *smart contracts* agissent en tant que tiers de confiance entre plusieurs entités afin de garantir l'application des règles définies. Étant donné qu'ils sont instanciés, stockés et exécutés sur la blockchain (c'est-à-dire par les nœuds du réseau), ils jouissent de ses propriétés de transparence, de traçabilité, d'immuabilité et de décentralisation, ce qui est le fondement même d'une application décentralisée.

f) Nano

Intrinsèquement, les structures de blockchains linéaires/chaines uniques qu'emploient Bitcoin et Ethereum souffrent d'une faible scalabilité, en partie liée à l'ajout séquentiel des blocs. En effet, étant donné leur fort lien cryptographique entre eux, ils ne peuvent être ajoutés à la blockchain que les uns après les autres. C'est avec pour objectif de contrer ce problème de scalabilité qu'un premier pas a été initié avec le projet de cryptomonnaie Nano (anciennement RaiBlocks) [119] fin 2014. Sa structure est basée sur un graphe orienté acyclique (*directed acyclic graph*, DAG), organisé en blocs-tressés (*block-lattice*) comme illustré sur la Figure 3-10. À l'inverse du modèle précédent, où toutes les transactions des différents utilisateurs sont réunies dans des blocs, Nano crée une blockchain spécifique à chaque utilisateur. La Figure 3-10 représente trois participants (bleu, vert et orange), dont leur chaine de blocs respective contient uniquement l'historique du solde et des transactions monétaires de leur propre compte (*i.e.* utilisation du modèle *Account/Balance* précédemment défini). Par ailleurs, il convient de noter qu'à l'inverse de Bitcoin et Ethereum, un bloc ne contient qu'une seule transaction. Chaque blockchain ne peut être mise à jour que par son propriétaire, ce qui permet d'ajouter instantanément un bloc et ce de manière asynchrone par rapport aux autres chaines : un transfert monétaire du participant bleu vers l'orange n'exige la création que de deux blocs, un par l'émetteur (bleu) et un par le récepteur (orange), la transaction n'étant valide que lorsque les deux participants auront ajouté leur bloc respectif. Ainsi, ce système permet d'atteindre des vitesses de transactions presque immédiates et ne nécessitant l'accord que des deux entités en question. À noter toutefois que l'ensemble des autres nœuds doivent suivre les échanges afin de mettre en jour leur copie du registre distribué.

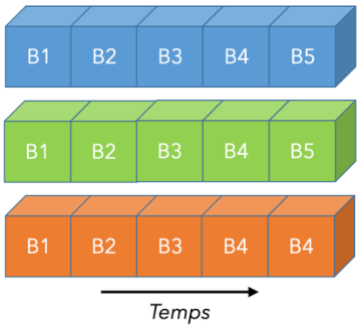


Figure 3-10 - Structure en blocs-tressés

g) IOTA

Toujours dans l'optique d'améliorer la latence des transactions, une autre approche, a été proposée, avec le projet IOTA [130] en 2016. Bien qu'il figure parmi les technologies de registres distribués, ses concepteurs le positionnent comme étant le successeur de la blockchain. Comme le

laisse sous-entendre son nom, IOTA a un fort penchant pour l'IdO avec des domaines d'applications tels que les transports, les chaines de productions, ou encore les villes intelligentes (*smart cities*). Pour atteindre ses ambitions, le système doit favoriser les micro-transactions, propices à l'IdO, tout en étant capable de gérer un grand nombre d'entre elles simultanément, de manière sécurisée. Le protocole IOTA repose sur *Tangle*, une architecture de registres distribués basée sur une autre forme de graphe orienté acyclique, comme illustré sur la Figure 3-11.

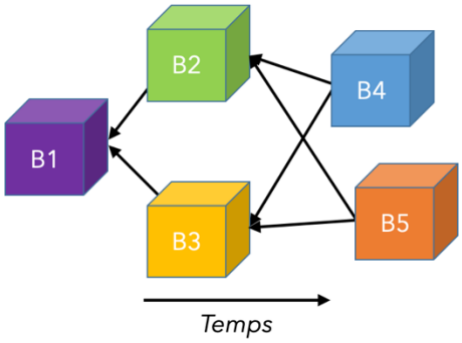


Figure 3-11 - Graphe orienté acyclique

Dans sa définition exacte, IOTA n'utilise pas de bloc, mais uniquement des transactions, liées les unes aux autres et basées sur le modèle UTXO vu précédemment (comme Nano, cela peut être vu comme un bloc contenant une seule transaction). Plus précisément, pour effectuer sa transaction sur le réseau, l'émetteur est dans l'obligation d'en vérifier et valider deux précédentes, sélectionnées aléatoirement et qui ne sont pas en conflit. Sur la Figure 3-11, B4 valide B2 et B3 pour être inscrit dans le graphe. C'est ce qui est appelé une validation directe. Au fur et à mesure que de nouvelles transactions sont ajoutées, les antérieures s'enfoncent dans le *Tangle*, les rendant de plus en plus actées par l'ensemble du réseau. Sur la Figure 3-11, B5 valide directement B2 et B3, ce qui valide indirectement B1. Ce fonctionnement s'avère remarquable, puisque plus le nombre de participants augmente, plus il y aura de transactions et par conséquent le temps de validation sera court. De plus, cette approche permet une parallélisation complète du traitement des transactions. Toutes ces propriétés font que la scalabilité théorique d'IOTA n'est limitée que par l'activité du réseau. En revanche, certaines études soulignent pour l'heure le manque d'évaluation concrète sur le niveau de sécurité qu'offre cette approche [116], [131], [132] et [133].

3.2.4. Blockchain et l'IdO

S'étant révélée dans le domaine de la cryptomonnaie [114], [117], [119] et [118], la blockchain se montre comme étant un mécanisme particulièrement intéressant dans un contexte de l'IdO pour plusieurs raisons, comme le soulignent les études [123] et [131]. En premier lieu, elle favorise la décentralisation d'un écosystème, en supprimant le besoin d'une entité centrale. Cette dernière pouvant d'une part être en mesure de contrôler l'ensemble du réseau, mais aussi subir tout type de pannes et d'attaques, rendant inopérant l'ensemble du système (*i.e. single point of failure*). L'autre avantage de cette décentralisation concerne la scalabilité. En effet, lorsqu'un réseau comporte un grand nombre d'éléments (*e.g.* plusieurs milliards dans le contexte de l'IdO), il apparait évident que si une seule entité doit gérer les communications, cette dernière doit être suffisamment puissante pour encaisser la charge de travail, tout en respectant les besoins de réactivité (*i.e. bottleneck*) [134]. Cela est d'autant plus vrai lorsque le réseau augmente. Cette approche s'apparente donc à une course à la puissance et aux performances, ce qui n'est pas viable sur le long terme, notamment avec l'expansion exponentielle de l'IdO.

Un autre aspect mis en avant par la blockchain concerne l'identité des utilisateurs/appareils du réseau. Cette dernière emploie des mécanismes (*e.g.* cryptographie asymétrique), permettant d'identifier de manière unique et immuable chaque entité du réseau [135]. Il est ainsi possible d'évaluer les interactions de chacun, de manière sécurisée, ce qui est à la base d'un système de confiance. De plus, cela offre un fort degré d'autonomie puisque le réseau est capable de s'autoévaluer.

Enfin, et c'est probablement la propriété principale de la blockchain qui en fait un moyen efficace en termes de sécurité, concerne l'immuabilité et l'auditabilité des données qu'elle stocke. Puisque toutes les informations inscrites dans la blockchain font l'objet d'un vote à travers un algorithme de consensus, aucune censure ne peut être appliquée, tout en apportant une transparence la plus totale. L'intégrité des données y est assurée grâce à sa structure de blocs liés les uns aux autres, et leur distribution en multiples exemplaires sur le réseau, permet à chaque entité d'y accéder à tout moment.

Grâce à ses propriétés, la communauté scientifique s'intéresse désormais à étendre le champ d'application de la blockchain, notamment dans le contexte de l'IdO, propice à l'utilisation d'architectures décentralisées. Cet attrait de la blockchain est aujourd'hui clairement acté de par la multitude de travaux présents dans la littérature ces dernières années, comme le résume le Tableau 3-1. À l'évidence, l'utilisation de la blockchain est envisagée pour la quasi-totalité des domaines

d'applications que permet l'IdO. Toutefois, si une telle association s'avère si prometteuse, pour quelles raisons n'est-elle à ce jour pas pleinement implémentée et déployée ?

Tableau 3-1 - Liste non exhaustive d'applications de la blockchain à l'IdO

Ville/Maison intelligente (*smart city/smart home*)	[136], [137], [138], [134], [139], [140], [141], [142]
Santé	[143], [144], [145], [146], [147], [148], [149], [150], [151]
Industrie	[152], [153], [154], [155], [156], [157], [158], [159], [160]
Transport	[161], [162], [163], [161], [164], [165], [166], [167], [168], [169], [170]
Énergie	[171], [172], [173], [174], [175], [176], [177], [178], [179], [180]
Agriculture	[181], [182], [183], [184]
Communication & Système d'informations	[185], [186], [187], [188]

Tout d'abord, il est intéressant d'identifier les grands axes de recherches sur lesquels se portent la communauté en ce qui concerne le couple blockchain et IdO. Une récente étude [131], réalise une synthèse de 62 travaux en les classant suivant leurs contributions : la sécurité, la gestion des identités et des données, la mise en place d'un environnement de confiance, le respect de la vie privée et enfin la monétisation des objets/données. Les résultats sont illustrés par le Tableau 3-2. À noter que certains travaux englobent plusieurs thèmes à la fois, d'où la distribution des pourcentages.

Tableau 3-2 - Axes de recherches de la communauté sur une adaptation de la blockchain à l'IdO

Axes de recherches	**Objectifs**	**Effort de la communauté**
Sécurité	Contrôle des accès, de l'intégrité, de la confidentialité et de la disponibilité des données	100%
Gestion des identités	Contrôle et gestion des identifications des objets	95%
Environnement de confiance	Élimination du besoin d'une entité centrale de confiance	84%
Gestion des données	Gestion d'un grand volume de données (disponibilité et stockage)	60%
Vie privée	Assurer l'auditabilité tout en préservant la vie privée	43%
Monétisation des objets/données	Création d'un modèle économique dédié à l'IdO	24%

D'après cette étude, l'intérêt de la blockchain concerne indéniablement la partie sécurité, à travers des règles de contrôle d'accès pour les objets, sans recourir à un service tiers, tout en préservant l'intégrité et la confidentialité des données. Un autre aspect fortement étudié concerne la gestion des identités (*e.g.* infrastructure de clés publiques) sans entité centrale pouvant être le point faible du système. Enfin, le dernier grand axe concerne la mise en place d'un environnement de confiance à l'aide de la blockchain, dans le contexte de l'IdO. Les travaux s'efforcent d'exploiter la nature décentralisée de cette dernière, en explorant différentes architectures utilisant des blockchains privées et/ou publiques existantes, ainsi que l'emploi de *smart contract*. Cet axe prend également en compte le déploiement de ces systèmes dans un écosystème IdO, en définissant les types d'interactions que chaque entité serait à même d'effectuer (*e.g.* simples requêtes par les objets, et support de la blockchain par les passerelles réseau). Force est de constater que peu de travaux s'orientent sur la monétisation des données et des objets. Cet aspect suggère la mise en place d'un écosystème décentralisé, où chaque entité est encouragée à partager ses données et ses ressources, ce qui constitue l'essence même de l'IdO. Malgré cela, la plupart de ces quelques études sur la monétisation ne sont que conceptuelles. Seulement 40% d'entre elles fournissent des résultats en simulation, contre 13% proposant une implémentation sur de vrais systèmes susceptibles de faire partie d'un réseau IdO, comme la carte Raspberry [189] et la carte Arduino Intel Edison [190]. Toutefois, ces derniers sont majoritairement utilisés comme nœud légers, n'étant en rien en charge de stocker, de gérer et de sécuriser la blockchain. De ce fait, cet état de l'art souligne une vision encore très étroite, positionnant la blockchain comme un outil, plutôt que de faire d'elle les fondations d'un nouvel écosystème. Mais plus important encore, la grande majorité des travaux se basent sur des blockchains déjà existantes (*e.g.* Ethereum), qui ne sont pas pensées à l'origine pour des systèmes fortement contraints tel que l'IdO. Néanmoins, il est important de les étudier, afin d'évaluer leur possible adaptation à ce dernier.

Avant tout, il convient de rappeler qu'aussi séduisante qu'elle puisse être, la blockchain n'est pas une solution absolue et ne convient pas à n'importe quelle application. C'est ce que souligne l'étude [191] à travers le diagramme illustré sur la Figure 3-12. Le premier élément concerne la répartition de la confiance dans le réseau. Doit-elle être assurée par une entité centrale ou doit-elle être répartie sur plusieurs entités ? Se pose alors la notion d'échanges. Est-ce que les participants du réseau peuvent s'échanger des informations directement entre eux (échanges pair à pair), ou une approche plus classique, regroupant les données vers un point central pour ensuite y être traitées puis redistribuées est-il plus adapté ? Enfin, il convient de définir qui a le droit d'accéder au réseau (publique ou privée) et qui est capable de mettre à jour les informations de la base de données (avec ou sans autorisation).

Figure 3-12 - Flot de décision pour définir la pertinence d'utilisation d'une blockchain [191]

De manière générale, l'IdO a pour but de connecter une multitude d'objets connectés, de différentes natures. C'est dans ce genre d'environnement hyperconnecté, fortement hétérogène et ouvert à tous qu'un réseau IdO devient particulièrement intéressant (*e.g.* les *smart cities*). En effet, plus ils sont nombreux et variés, plus l'intérêt est grand, car cela mène à la création de nombreux services. De ce fait, une architecture centralisée ne semble pas appropriée pour des raisons de réactivité (latence réseau, temps réel), mais aussi de sécurité (*i.e. single point of failure*). Chaque objet du réseau ne peut donc pas faire confiance à une entité centrale ni à ses homologues (rien ne garantit la fiabilité de ces derniers). Puisque leur but est de s'échanger des informations pour une utilisation immédiate et/ou future, une base de données est requise. En l'absence de point central, celle-ci doit être dupliquée et distribuée sur de multiples nœuds, afin d'être résiliente aux pannes ou corruptions. Enfin, étant donné le besoin d'interconnecter de nombreux acteurs en tout genre, l'IdO peut être considéré comme un réseau public, ouvert à tous, où chaque entité peut mettre à jour la base de données. Évidemment, dans de telles conditions, il est primordial d'assurer un fort niveau de sécurité. Ainsi, d'après ce cheminement, une « blockchain publique sans autorisation » (*public permissionless blockchain*) comme Bitcoin, Ethereum, Nano et IOTA, semble être l'approche la plus adaptée au contexte de l'IdO, comme le montre la Figure 3-12. Par conséquent, il convient d'étudier leurs caractéristiques, résumées dans le Tableau 3-3. Le nombre de blocs et de transactions des blockchains

ne cessant de croitre, les valeurs indiquées correspondent aux données recensées lors de l'écriture de cette thèse.

Tableau 3-3 - Caractéristiques de Bitcoin, Ethereum, Nano et IOTA

Propriétés / Blockchains	Bitcoin	Ethereum	Nano	IOTA
Taille	343 Go [192]	409 Go [192]	29,61 Go [193]	1,6 Go [194]
Nombre de blocs ou transactions	644 552 blocs [192]	10 697 176 blocs [192]	56 953 665 blocs [193]	1 000 000 TXs [194]
Énergie / transaction	569 kWh [195]	19 kWh [196]	0,1 Wh [197]	0,1 Wh [198]
Transaction / seconde (TPS) [199]	7	12 – 15	7 000	1 500
Temps de calcul d'un bloc (sur Raspberry Pi3 B+ [189])	~ 7 milliards d'années / bloc[1]	Impossible[2] [200]	> 4 heures[1]	82,9 s / TXs [201]

1 : Évaluation effectuée dans cette thèse

2 : Sans disque dur externe hautes performances

En premier lieu, les blockchains publiques les plus utilisées, Bitcoin et Ethereum montrent des caractéristiques totalement impossibles à entrevoir pour une application à l'IdO, tels que l'espace de stockage requis (se comptant en plusieurs centaines de Go), la consommation énergétique (en kWh) ainsi que les faibles débits atteints (jusqu'à 15 transactions par seconde). Il est clairement évident que ces blockchains ne sont pas adaptées à l'IdO. D'autres tentent de surmonter ces problèmes, comme Nano et IOTA, affichant une réduction de l'espace de stockage de plus d'un facteur 10, une diminution de la consommation énergétique (de 0,1 Wh), pour un débit nettement supérieur (d'un facteur 1000). En revanche, ces blockchains sont à ce jour beaucoup moins utilisées, limitant ainsi le nombre d'études réalisées en termes de viabilité, de sécurité, mais aussi d'application possible à l'IdO. Par ailleurs, en guise de prémices, il est intéressant de comparer les temps de calcul que requiert la création d'un bloc pour les différentes blockchains, sur un système réel composant un réseau d'IdO. La comparaison est ici faite sur une carte Raspberry Pi 3B+ [189], dont les caractéristiques peuvent être similaires à celles disponibles sur les passerelles réseau (*gateways*), à savoir un processeur quatre-cœurs Cortex-A53, 1 Go de RAM et une capacité de stockage variable (supérieure à 256 Go, suivant la carte SD). Alors qu'il est impossible d'exécuter le protocole Ethereum sans un disque externe, la génération d'un bloc Bitcoin est tout simplement impossible à l'heure actuelle. La documentation officielle du protocole Nano [202] stipule un minimum de 4 Go de RAM pour ses *full nodes*, ainsi qu'un disque SSD ayant un minimum de 80 Go d'espace de stockage disponible. Pour faire le rapprochement avec la comparaison précédente d'un système réel de l'IdO, la carte Raspberry Pi 3B+, cette dernière ne satisfait pas les spécifications de Nano, ni même la toute nouvelle Raspberry Pi 4 [203], embarquant 2 Go de RAM. Cependant, en implémentant uniquement sa fonction du

PoW (Blake2b [204]), il faut plus de 4 heures à une carte Raspberry Pi3 Modèle B+ pour accomplir cette tâche avec la difficulté actuelle du réseau [205], ce qui représente en moyenne plus 469 millions d'itérations. Enfin, IOTA requiert plus d'une minute de calcul, ce qui est en désaccord avec le besoin de temps réel de l'IdO. De plus, d'après son guide officiel [206], un nœud complet IOTA exige au moins 2 Go de RAM et un disque SSD pour le stockage des données, de minimum 50 Go, d'après un guide communautaire [207]. Rappelons que les cartes Raspberry correspondent aux types de systèmes pouvant être considérés comme étant les plus puissants dans un environnement d'IdO. Car si l'on se réfère plus précisément aux objets connectés tel que le JN5189 [208] de chez NXP, les capacités sont drastiquement inférieures (Cortex-M4, 152 Ko de RAM et 640 Mo de FLASH). Cela met ainsi en évidence le gap existant entre les ressources requises par les blockchains actuelles et celles dont disposent les entités constituant un écosystème d'objets connectés. Par conséquent, il est primordial d'analyser précisément les différents éléments, permettant d'entamer l'adaptation de la blockchain à l'IdO.

3.2.5. Les défis d'une adaptation de la blockchain à l'IdO

a) Architecture

Comme évoqué au début de cette thèse, les systèmes de Cloud ont été fortement utilisés pour outrepasser les faibles capacités des objets connectés, comme le souligne [131]. En revanche, cette approche centralisée oblige l'ensemble du réseau à faire confiance à cette entité (*e.g.* utilisation des données). D'un point de vue sécurité, cette architecture souffre d'une faible résistance aux pannes et aux attaques, de par sa concentration de toute l'intelligence en un seul point. Toutefois, établir un écosystème entièrement décentralisé, en s'appuyant sur la blockchain, n'est pas chose simple. En effet, l'IdO interconnecte toute une panoplie d'appareils divers et variés, avec des ressources inégales et souvent très limitées, comme la comparaison précédente entre la carte Raspberry et le JN5189. De plus, cette diversité est fortement liée à l'application en question. Il est donc primordial de clairement définir l'architecture réseau employée, ainsi que l'implication des différents types d'objets sur la blockchain et dans la sécurisation du système. La Figure 3-13 reprend les configurations actuellement proposées dans l'état de l'art, rapportées par [123] et [131].

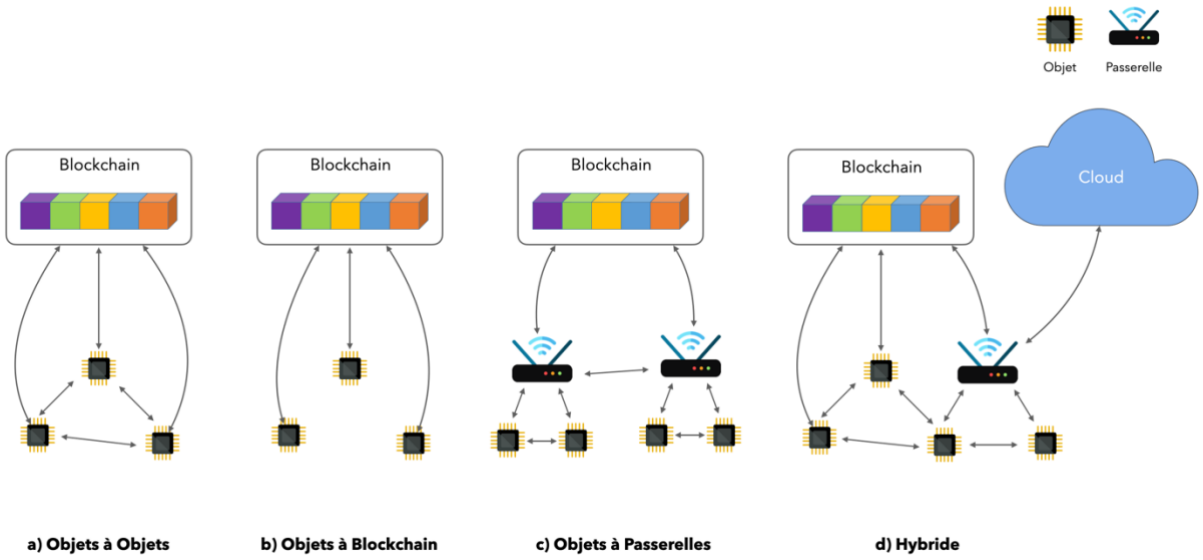


Figure 3-13 - Architectures d'intégration de la blockchain à l'IdO (inspiré de [123] et [131])

Objets à Objets (a): [123] définit cette approche comme étant la plus rapide en termes de latence, en raison de ses interactions directes entre les objets, offrant une faible latence, idéal lors de communications fréquentes. Cela implique que ces derniers soient en mesure de gérer le routage d'un point A à un point B. De plus, la blockchain n'entre pas en compte lors de chaque interaction et seulement une partie des données y est stockée (les plus pertinentes). [131] complète ce modèle en précisant que les objets ne possèdent pas de copie locale de la blockchain et qu'ils peuvent uniquement effectuer des transactions. Cela signifie qu'ils sont donc en mesure d'interagir directement avec la blockchain, sans intermédiaire, mais aussi qu'ils intègrent des fonctions cryptographiques (*e.g.* signature). Le compromis est ici de maximiser l'autonomie des objets, au détriment d'une complexité de calcul accrue.

Objets à Blockchain (b) : cette architecture reprend sensiblement les mêmes caractéristiques que la précédente, avec toutefois une plus forte sollicitation de la blockchain. En effet, ici toutes les interactions y sont enregistrées permettant de les tracer, ce qui augmente la latence. Le compromis s'oriente cette fois sur un besoin plus fort de sécuriser les interactions, dégradant la performance du réseau.

Objets à Passerelles (c) : proposé par [131], cette approche est celle se rapprochant le plus des architectures réseau actuelles, centrées autour des passerelles (*gateways*). Ces dernières permettent de relier les réseaux sans fils, au monde IP, rendant possible une interconnexion complète. Dans ce modèle, chaque objet est enregistré sur une passerelle, qui transmet alors les interactions à la blockchain. Comme l'indique [131], cela permet d'avoir une traçabilité des communications, mais aussi de relier et sécuriser les interactions entre différents réseaux/blockchains. Étant donné les

ressources plus conséquentes dans les passerelles par rapport aux objets, elles peuvent être utilisées pour stocker et maintenir une copie de la blockchain. Par ailleurs, elles sont également en mesure de traiter les données provenant des objets et de soumettre dans la blockchain uniquement les informations pertinentes (*i.e. fog computing*). Enfin, l'utilisation des *smart contracts* devient alors possible, apportant une autre dimension dans les possibilités d'applications. En contrepartie, cette approche est plus centralisée que les précédentes, puisque ce sont uniquement les *gateways* qui sont en mesure d'interagir avec la blockchain.

Hybride (d) : Enfin, cette dernière architecture est un mélange des précédentes. Les objets comme les passerelles peuvent accéder directement à la blockchain et communiquer entre eux sans intermédiaire. [123] et [131] soulignent qu'une telle approche représente la meilleure solution d'intégrer la blockchain à l'IdO puisqu'elle permet de tirer pleinement profit des capacités de cette dernière, tout en conservant la possibilité aux objets d'interagir directement entre eux. De plus, les auteurs incluent le Cloud, permettant de compléter le modèle, en y apportant ses grandes capacités de calcul/stockage. Toutefois, la grande difficulté de cette solution hybride réside dans le choix des interactions qui doivent faire l'objet d'une plus grande sécurité (*i.e.* via blockchain), et de définir des métriques permettant de faire ce choix, en temps réel.

b) Stockage

Par définition, la blockchain est une base de données à ajout unique, comme le montrent Bitcoin [114] et Ethereum [117]. Cette propriété est la conséquence directe de sa plus grande force, à savoir le lien cryptographique entre chaque bloc. En effet, la sécurité d'un tel système réside dans l'immuabilité des données écrites, qui dépendent les unes des autres. Dès lors, il devient évident que le simple fait de modifier, ou de supprimer une information (et donc un bloc) remet en cause toute la validité de la chaine. Au cours de son utilisation, la blockchain devient donc de plus en plus imposante. Cette conséquence s'avère être une problématique de taille dans son adoption à l'IdO, où d'une part l'espace de stockage des objets est fortement limité et d'autre part, leur fulgurante expansion s'accompagnera d'une quantité de données encore jamais atteinte [209].

[191] met en avant l'idée de ne répartir la blockchain que sur certaines entités, faisant référence aux différents types de participants (*full nodes* et *lightweight node*, *cf.* section 3.2.3.c). Par exemple, un nœud complet possèdera l'intégralité de la blockchain, alors qu'un nœud léger n'en conservera qu'une partie (*e.g.* uniquement les *block headers* [114]). Dans le même état d'esprit, ce qui est appelé « mini-blockchain » [210], consiste à fixer une taille limite de blocs au bout duquel les anciens sont supprimés. Toutefois, pour assurer la sécurité de leur système, les auteurs stipulent que seules les transactions des anciens blocs sont supprimées, et que les *block headers* doivent toujours être

conservés. Alors que la première approche sous-entend un amoindrissement de la décentralisation, ces deux stratégies permettent, dans une certaine mesure, de réduire l'espace de stockage nécessaire, mais ne résolvent pas entièrement le problème puisqu'il faudrait toujours avoir une capacité de stockage illimitée. De ce fait, l'approche souvent actée dans la littérature consiste à ne stocker uniquement les informations jugées nécessaires afin de ne pas alourdir inutilement la blockchain [123], [131].

Malgré tout, ces méthodes ne constituent pas de réelles solutions au problème du besoin de stockage grandissant de la blockchain. Qu'il soit réparti sur plusieurs entités ou que seulement une partie des données ne soit stockée ou conservée, la taille de la blockchain ne cessera de croitre. La seule solution consiste à trouver un compromis entre espace de stockage et sécurité, en supprimant complètement les anciens blocs. C'est ce que propose IOTA [130], en effectuant un *snapshot* tous les trois mois [207]. L'objectif est de ne conserver uniquement les comptes contenant des jetons. Tout le reste est effacé. En revanche, la force de Nano [119] en termes de stockage réside dans sa structure associée à un modèle de *Account/Balance*, dans lequel chaque bloc contient le solde actuel d'un nœud. De ce fait, il est possible de ne conserver uniquement le dernier bloc de chaque compte, afin de réduire drastiquement l'espace de stockage. Ce dernier ne dépend alors que du nombre de nœuds sur le réseau (*i.e.* 1 bloc/nœud). Évidemment, dans ce cas de figure le niveau de sécurité d'une telle blockchain, à savoir son immuabilité, est moindre. Toutefois, bien que l'auteur indique cette possibilité de configurer la politique de stockage des nœuds, il préconise de télécharger dans un premier temps l'intégralité de la blockchain, de la vérifier et ensuite de supprimer les blocs. Cela sous-entend que l'entièreté de cette dernière est conservée et par conséquent, la présence de *full nodes*. C'est par ailleurs ce que l'on retrouve sur l'explorateur officiel de Nano [193].

c) Scalabilité

Un autre élément important du Tableau 3-3 concerne le débit de transactions par seconde (*transaction per second*, TPS). En effet, puisque Bitcoin (7 TPS), Ethereum (12 à 15 TPS) et leurs homologues ont pour but de proposer un moyen de paiement alternatif, ils doivent être en mesure de concurrencer les systèmes existants tels que VISA (24 000 TPS) ou encore PayPal (193 TPS) [211]. Bien que cela ne soit pas suffisant, il est intéressant de constater l'amélioration qu'apportent la structure à base de graphe utilisée par IOTA (1 500 TPS) et celle de blocs-tressés par Nano (7 000 TPS), en termes de transactions par seconde (d'un facteur 100 à 1 000). Malgré cela, il persiste un gap important qu'il est primordial de résoudre pour que ces cryptomonnaies soient massivement adoptées. Mais cette faculté de supporter un important nombre de transactions ne s'applique uniquement pas aux systèmes monétaires, mais également à d'autres, comme l'IdO. En effet, en raison de son expansion, le nombre d'objets connectés se comptera prochainement en dizaine de

milliards [9]. Il faudra par conséquent que le système sur lequel ils reposent soit en mesure de gérer cette quantité astronomique de données.

Plusieurs alternatives sont pensées, étudiées et mises à l'épreuve. Certaines d'entre elles adoptent la technique de « diviser pour mieux régner » tel que le *sharding* [212], [213], [214]. Cette méthode consiste à découper une base de données en plusieurs parties. C'est en quelques sortes une décentralisation d'une décentralisation. De manière schématique, le *sharding* peut s'apparenter à un sous-réseau. Le principe est de créer plusieurs chaines alternatives, dont chacune d'elles se comporte comme une vraie blockchain, à la différence qu'elles ne sont validées que par une partie du réseau. Ce procédé permet d'une part de désengorger le réseau en parallélisant les opérations (*e.g.* chaque *shard* traite uniquement les transactions concernant ses participants). La première difficulté de ce mécanisme concerne la bonne décentralisation de chaque *shard* sous peine d'impacter fortement sa sécurité. De plus, pour ne pas perdre le bénéfice d'une telle approche, la répartition doit être efficace afin d'éviter une trop forte fréquence de communication entre les *shards*, qui par ailleurs, doivent faire l'objet de vérifications.

Dans le même esprit, les *sidechains* [215], [216], [217] sont également une solution permettant d'améliorer la scalabilité, en apportant de l'interopérabilité entre plusieurs blockchains. Une *sidechain*, ou « chaine enfant » [215], est une blockchain à part entière, avec ses propres caractéristiques, mais bénéficie du réseau et de la sécurité de la chaine parente. Ainsi, toute transaction déportée sur la chaine secondaire allège la charge de travail de la principale.

Enfin, une dernière alternative concerne les systèmes dits *off-chain*, tel que le *lightning network* [218] de Bitcoin. L'idée derrière ce concept est d'ouvrir un canal de communication entre deux ou plusieurs entités, leur permettant d'effectuer des transactions en minimisant au maximum l'intervention de la blockchain. Cette dernière est uniquement sollicitée lors de l'ouverture et de la fermeture du canal. Par exemple, si Alice et Bob veulent s'échanger de l'argent, ils ouvrent un canal et une fois validé par la blockchain, ils peuvent tout deux effectuer autant de transferts qu'ils le souhaitent. À la fin de leurs interactions, le canal est clôturé et seulement le résultat final (c'est-à-dire le solde d'Alice et Bob) est inscrit dans la blockchain. Ce système permet ainsi de réduire considérablement le nombre de transactions à traiter sur la blockchain.

d) Latence

Au-delà du nombre de transactions à la seconde qu'il est possible de traiter, le temps de confirmation est également important à prendre en compte. Comme le souligne [191], lors d'un paiement par le réseau VISA, le temps de vérification, à savoir la latence, n'est que de seulement quelques secondes. Dans le cas d'une blockchain, une opération n'est considérée comme valide

uniquement lorsqu'elle figure dans un bloc. De plus, la synchronisation de la blockchain sur l'ensemble du réseau étant imparfaite, il est possible qu'un bloc (et donc les transactions qu'il contient) ne soit pas reçu et accepté par l'ensemble des nœuds au même moment. En raison du lien immuable qui les lie, plus un bloc a de successeurs, plus la probabilité qu'il soit accepté par la majorité des nœuds est importante. De ce fait, il est fortement conseillé, pour des questions de sécurité, d'attendre un certain nombre de blocs, pour s'assurer que la transaction a bien été acceptée par l'ensemble du réseau. Par exemple, il est souvent recommandé dans le protocole Bitcoin d'attendre six blocs, soit environ une heure [219]. Pour une application à l'IdO, où il y a un fort besoin de temps réel, cette latence n'est pas acceptable.

Pour contrer ce problème, certaines directives dans l'état de l'art suggèrent soit d'accélérer le temps de génération d'un bloc, soit d'augmenter le nombre de transactions qu'il peut contenir [116]. Alors qu'un tel raisonnement semble cohérent, ces deux solutions ne résolvent pas le problème. En effet, la réduction du temps de génération d'un bloc signifie une simplification dans la procédure de création. Il est donc plus facile pour les nœuds, et pour un plus grand nombre d'entre eux, de proposer de nouveaux blocs à ajouter. La conséquence est que le réseau se retrouve surchargé de propositions, qu'il convient de répandre sur l'ensemble des participants, mais aussi de les trier afin de n'en conserver qu'une seule et unique, de manière unanime. Dans la pratique, un réseau est imparfait, ce qui signifie qu'il existe une latence irréductible, ralentissant la vitesse de propagation des propositions. De ce fait, elles n'arrivent pas en même temps à tous les nœuds, mais surtout, leur ordre d'arrivée peut être différent. Par conséquent, cette approche cause d'une part un gaspillage des ressources du système dans la création de blocs qui seront finalement rejetés, mais ne résout en rien le problème de la latence, puisque, la probabilité qu'un bloc ne soit pas accepté de manière unanime est fortement augmentée, exigeant toujours d'attendre un certain temps/nombre de blocs pour s'assurer de l'acceptation définitive de la transaction [116]. La seconde approche consiste cette fois à augmenter la taille maximale d'un bloc, lui permettant d'inclure plus de transactions. Là encore, cela ne constitue pas une solution. En effet, une taille de bloc plus importante conduit à un temps de propagation sur le réseau plus long. Les nœuds, ne voyant pas de nouveaux blocs à ajouter, proposent le leur, ce qui conduit une nouvelle fois à plus de propositions de blocs qu'il fait trier, comme dans le scénario précédent. De plus, une taille plus importante implique de plus grandes capacités (*e.g.* mémoire, bande passante) ce qui restreint la génération de bloc aux entités les plus puissantes, réduisant ainsi le niveau de décentralisation. Par conséquent la solution du problème de la latence ne peut être résolue par une modification de la taille d'un bloc ou de leur temps de génération. Un compromis doit être trouvé entre performances et sécurité.

e) Énergie

Bitcoin a souvent fait parler de lui pour sa tendance à être énergivore. Pour donner une dimension de leur consommation, lors de l'écriture de cette section, Bitcoin se montre équivalent à la consommation énergétique annuelle de la République Tchèque [195], 43ème au classement mondial des consommations des pays, sur un total de 197 reconnus par l'ONU. Ethereum quant à lui, est proche de celle de l'Uruguay [196], à la 102ème position. L'analyse de [195] estime la consommation annuelle de Bitcoin à presque 67 TWh, pour un total d'environ 10 400 *full nodes* [220]. Ethereum atteint les 8 TWh [196], avec en moyenne 8542 nœuds [221]. Évidemment, cette dépense énergétique est en partie liée au nombre de participants connectés. Rapporté sur un nœud, cela représente 6 GWh/an pour Bitcoin, contre 936 MWh/an pour Ethereum, soit respectivement l'équivalent de l'énergie moyenne consommée par plus de 500 et 86 foyers américains sur un an [196].

Comme évoqué plusieurs fois au cours de cette thèse, les objets composant un écosystème d'IdO sont souvent fortement contraints en termes de ressources de calcul et de mémoire. De plus, leur nature les conduit à être déployés dans un environnement où leur seule source d'énergie provient d'une batterie, parfois rechargeable (*e.g.* via panneaux solaires). Par conséquent, l'adoption de la blockchain à l'IdO doit impérativement tenir compte de la dépense énergétique. Aux vues des estimations précédentes, il apparait clairement une incompatibilité des blockchains classiques avec un environnement de l'IdO, comptant des milliards d'objets. [191] et [122] soulignent les deux principales sources de cette consommation énergétique, à savoir comme étant d'une part la communication entre les entités, mais surtout le processus de génération des blocs. Toutefois, comme le montre le Tableau 3-3, IOTA et Nano se montrent moins exigeant énergétiquement ce qu'il convient d'approfondir pour l'adaptation à l'IdO.

f) Incitation

Dans un écosystème comportant d'innombrables entités, il s'avère très complexe (voire impossible) de s'assurer du bon respect des règles de chaque participant, le tout de manière décentralisée. De plus, le risque qu'un acteur soit malveillant est d'autant plus élevé lorsque cet environnement est accessible à tous. Pour remédier à cela, il est primordial d'instaurer un mécanisme d'incitation (*incentive*) au bon respect des règles. Par exemple, dans le contexte des cryptomonnaies telles que Bitcoin et Ethereum, tous les nœuds sont incités à vérifier et créer de nouveaux blocs en accord avec le protocole, afin d'obtenir une récompense (*i.e.* rémunération). Dans le cas contraire, les nœuds auront dépensé leurs ressources (*e.g.* électricité) sans aucun gain. Mais ce concept d'incitation peut aller plus loin, notamment dans le cadre de l'IdO. En effet, en plus d'encourager au bon respect

des règles, un réseau offrant tout un panel de services, peut attirer de nouveaux acteurs, qui à leur tour proposeront de nouvelles possibilités, venant enrichir le système, devenant encore plus attractif. C'est donc au travers de cette notion d'incitation, qu'un écosystème de l'IdO a les moyens de se développer, tout en assurant sa sécurité. Il s'agit là d'une propriété majeure, qui malgré cela, est souvent négligée dans l'état de l'art lorsqu'il est question d'utilisation des blockchains à l'IdO.

g) Sécurité et vie privée

Il n'est pas sans rappeler que la volonté d'associer la blockchain à l'internet des objets est avant tout motivée à des fins de sécurité, pour ses propriétés de données immuables, d'authentification et d'intégrité, ainsi que sa transparence et sa tolérance aux fautes. Toutefois, un système n'étant jamais sans faille, il est judicieux de recenser les principales attaques connues sur les blockchains, afin d'évaluer leur criticité pour le contexte de l'IdO. De ce fait, cette section résume les principales et les plus générales d'entre elles, extrait des études [191], [122] et [116].

Attaque des 51% : aussi appelée *attaque de la majorité*, elle représente l'attaque ultime de la blockchain. En effet, elle consiste à prendre le contrôle total du système, en possédant plus de la moitié des capacités permettant la génération des blocs. Suivant le type d'algorithme de consensus utilisé, cela se traduit par de la puissance de calcul, de pouvoir de vote, etc. L'attaquant peut alors procéder à de la censure (blocage de certaines transactions), ou même réécrire la blockchain.

Double dépense : le but est de dépenser deux fois le même jeton. Dans un système de cryptomonnaie tel que Bitcoin, un attaquant souhaitant faire un achat, effectue simultanément une transaction au marchand et une autre sur son propre compte afin de se rembourser. Lorsque la transaction vers le marchand est acceptée par ce dernier, l'attaquant reçoit son achat. Mais en raison de la latence dans le processus de consensus, il est possible que ce soit la transaction de remboursement qui soit acceptée par la majorité du réseau, invalidant alors la transaction d'achat. Par conséquent, l'attaquant récupère son argent tout en conservant son achat. Il y a là une double dépense.

Attaque Sybil : un attaquant tente de créer une multitude de fausses identités afin d'obtenir une grande influence sur le réseau, se présentant alors comme étant légitime. Il lui est alors possible de manipuler les participants à sa guise. Par exemple, dans un scénario de vote, l'attaquant effectuera plusieurs votes, sous des noms différents, afin d'obtenir la majorité.

Attaque par rejeu : le but de cette attaque est d'écouter les transmissions entre deux entités et de les retransmettre par la suite, sans aucune modification, afin d'usurper l'identité de l'une des entités.

Attaque par déni de service (DoS/DDoS) : l'objectif est de saturer un système/service, en lui transmettant une grande quantité de requêtes simultanément afin de le rendre inutilisable. Cette inondation du réseau peut s'effectuer à partir d'une seule source (*denial of* service, DoS), ou de plusieurs (*distributed denial of service*, DDoS).

Comme indiqué précédemment, la propriété de transparence d'un système blockchain permet d'assurer une traçabilité et une vérification des interactions entre les entités. Cela s'avère être une force pour la sécurité, mais révèle toutefois des problèmes en termes de vie privée (*privacy*). Par exemple, si Alice décide d'effectuer un transfert à Bob, le réseau doit être en mesure de vérifier qu'Alice possède effectivement les fonds. Si Alice souhaite que cela soit secret, il faudrait qu'elle chiffre ce transfert, mais dans ces conditions, les autres participants ne seront pas en mesure de vérifier la validité de l'opération. Outre l'aspect de sécurisation, l'accessibilité aux données est également un enjeu, comme le montre l'exemple de l'agriculture connectée [184] : si la collecte d'informations provenant des capteurs (*e.g.* composition de l'eau, température, humidité, pH, etc.) n'est pas protégée, un attaquant est en mesure de suivre l'évolution de ces données lui permettant non seulement d'identifier le type d'agriculture dont il est question, mais aussi de planifier des actes malveillants (*e.g.* sabotage par manipulations de données spécifiques). Ces simples scénarios illustrent parfaitement le compromis entre vie privée et sécurité que peut apporter une blockchain.

Comme le soulignent [191] et [131] les cryptomonnaies comme Bitcoin et Ethereum sont pseudo anonymes car les identifiants (*e.g.* adresse Bitcoin) ne relient aucunement à l'identité réelle d'une personne (ou d'un appareil). Toutefois, certaines études dans l'état de l'art tentent de relever ce défi, comme les protocoles de preuve à divulgation nulle [222] (*Zero Knowledge Proof*) dont le principe est de prouver au cours de nombreuses interactions que deux entités connaissent et partage un secret commun, sans le divulguer. Une alternative, appelée *Non-Interactive Zero-Knowledge Proof* [223] se base sur des primitives cryptographiques afin de réduire ces échanges, au détriment d'une plus grande complexité de calcul. Une autre approche est la signature d'anneau [224] (*Ring Signature*), qui consiste à utiliser la clé publique de plusieurs entités d'un groupe dans la signature d'un message. Bien qu'ils soient prometteurs, leur utilisation s'avère inadaptée en termes de ressources requises, dans le contexte de l'IdO.

b) Consensus

Les précédentes sous-sections mettent en avant les éléments clés de l'adaptation de la blockchain à l'IdO. Toutefois, il en reste un, qui est le garant de tous les autres, à savoir les performances (cf. Scalabilité, Latence, Énergie), la viabilité (cf. Incitation), mais aussi la sécurité (cf. Sécurité et vie privée) : l'algorithme de consensus. En effet, ce dernier est le cœur même d'un système blockchain,

car il est le mécanisme définissant l'ensemble des règles permettant à toute entité du réseau de s'accorder sur une même et unique vision (*i.e.* ordre des transactions/blocs). De par son importance cruciale, il est indispensable d'étudier les différents algorithmes de consensus déployés dans les blockchains actuelles, en prévision d'une adaptation à l'IdO, comme le soulignent les travaux [191], [123], [116], [131] et [132].

Proof-of-Work (PoW) [114]: certainement un des plus célèbres en raison de son utilisation dans Bitcoin par Satoshi Nakamoto, le PoW consiste à élire un seul et unique nœud en charge de proposer le prochain bloc à ajouter dans la blockchain. Pour que cela se fasse de manière sécurisée et aléatoire, l'idée est d'imposer les participants à trouver la solution d'un puzzle cryptographique, complexe à résoudre, mais simple à vérifier. L'analogie la plus fréquente est le jeu du Sudoku : la résolution d'une grille exige temps et réflexion (*i.e.* puissance de calcul et énergie), alors que sa vérification est quasiment instantanée. Cette approche incite donc les nœuds à respecter les règles, sans quoi ils gaspilleraient leurs ressources sans aucun bénéfice, c'est-à-dire la récompense pour avoir trouvé un bloc correct. Dans le cas de Bitcoin, ce puzzle consiste à trouver un nombre aléatoire (*nonce*), permettant à l'empreinte du futur bloc (obtenue via la fonction de hachage SHA-256 [225]) de commencer par un certain nombre de zéros. Puisque la seule manière de trouver ce nombre est de tester toutes les combinaisons possibles, ce protocole constitue ainsi une « preuve de travail ». Toutefois, un inconvénient majeur de cette approche est le gaspillage des ressources. En effet, lorsqu'un nœud trouve une solution au puzzle, il la transmet à ses homologues. Cette opération implique un temps de propagation sur l'ensemble du réseau, pendant lequel les autres nœuds tentent toujours de résoudre « inutilement » le puzzle, jusqu'à ce qu'ils reçoivent la solution. Pour limiter cet effet, Satoshi introduit la notion de difficulté, qui, en fonction de la rapidité du système à résoudre le puzzle, évolue de sorte qu'une solution soit trouvée à intervalle de temps régulier. Comme l'explique l'article [226], ce choix suppose qu'il faut environ une minute pour qu'un nouveau bloc soit entièrement distribué sur l'ensemble du réseau. En prenant un temps de résolution de dix minutes, cela signifie que 10% des ressources sont gaspillées (1min / 10min = 10%). Ainsi, plus le temps de résolution est grand, plus le gaspillage diminue, au détriment de la latence. Bien que le PoW ait montré sa robustesse, il endosse à lui seul les nombreux inconvénients des blockchains basées sur ce modèle, à savoir la consommation excessive d'énergie, son faible débit, mais aussi d'un point de vue global, un fort gaspillage des ressources. La popularité grandissante de Bitcoin a conduit à une forte augmentation de ses participants (notamment les *full nodes*) ce qui a conduit à une augmentation de la difficulté du PoW, garant de la sécurité du système. À l'origine possible sur un simple processeur (CPU) seulement, puis sur processeur graphique (GPU) et FPGA, aujourd'hui seuls des ASICs sont en mesure d'exécuter ce PoW [227]. De plus, cette course à la capacité de calcul pousse les acteurs à se regrouper dans ce qui est communément appelé des *pools*, où ils mettent en

commun leurs ressources, ce qui conduit à une recentralisation non prévue et dangereuse du réseau (*cf.* attaques des 51%). Pour contrer ce phénomène, Ethereum emploie une autre implémentation du PoW appelée Ethash [228], basée sur une forte utilisation de la mémoire (*memory-hard hashing*), ce qui le rend résistant aux approches matérielles dédiées (*i.e.* difficile ou économiquement inefficace à mettre en œuvre par des ASICs). Toutefois, cette approche requiert initialement 1 Go de données et 16 Mo de mémoire cache, qui augmentent respectivement de 8 Mo et de 131 Ko tous les 30 000 blocs. Nano utilise lui aussi le PoW, non pas en tant qu'algorithme de consensus, mais comme un moyen de limiter les spams. Une nouvelle version prévoit d'évoluer sur un PoW exploitant la mémoire, dont les premiers résultats d'évaluation en termes de ressources sont du même ordre de grandeur qu'Ethash [229].

Proof-of-Stake (PoS) [230]: face à l'utilisation intensive de ressources et d'énergie du PoW, l'intérêt d'une alternative s'est rapidement emparé de la communauté. Le *Proof-of-Stake* figure parmi les plus populaires. Ce protocole se base sur le fait que les entités les plus actives sur un réseau sont celles qui sont le moins susceptibles de l'attaquer. Dans ce schéma, au lieu de dépenser sa puissance de calcul (et son énergie) à la création de blocs, les participants ont simplement besoin de prouver leur implication dans le réseau. La technique de preuve la plus répandue consiste à miser de l'argent (*e.g.* des Ethers) pour être sélectionné comme étant le prochain publieur de bloc et ainsi gagner la récompense. Cette sélection s'effectue de manière aléatoire, mais avec une pondération sur la quantité d'argent misé. En effet, puisque les participants les plus riches sont ceux qui ont le moins d'intérêt à perturber le système, ces derniers sont plus dignes de confiance. Plus aucun calcul complexe n'est nécessaire, ce qui permet au PoS d'être moins énergivore que le PoW mais aussi de produire plus rapidement des blocs. En revanche, cette simple version souffre d'un problème de centralisation, car les nœuds les plus riches sont plus souvent élus dans la génération de blocs. Pour remédier à cela, il est possible d'ajouter des conditions en plus de la quantité d'argent misée, dont les plus connues sont les méthodes « *Coin Age Selection* » [231] et « *Randomized Block Selection* » [232]. La première consiste à favoriser les nœuds ayant misé le plus d'argent, et ce sur une longue durée (quantité × temps de mise). Une fois désigné pour publier le prochain bloc, le temps de mise du nœud est réinitialisé, l'empêchant d'être à nouveau sélectionné. Quant à la seconde, il s'agit d'une combinaison entre la quantité de mise et la plus faible valeur d'un hash qui permet de déterminer quel nœud sera élu.

Delegated Proof-of-Stake (DPoS) [233]: ce protocole est similaire au *Proof-of-Stake* précédent, à la différence qu'au lieu de miser pour être le prochain publieur de bloc, les nœuds élisent des représentants qui seront en charge de générer et valider les prochains blocs à leur place. Le poids de ce vote dépend de la mise, comme dans le PoS. Lorsqu'un représentant ne respecte pas les règles, il est automatiquement destitué par les nœuds, qui élisent un remplaçant. Le principal avantage de ce

protocole réside dans la restriction du nombre de publieurs de blocs ce qui permet d'accélérer le processus de consensus. En revanche, cette réduction doit être contrôlée et bien définie pour éviter de dégrader le niveau de sécurité d'une telle blockchain. Dans son implémentation, Lisk [234] fixe le nombre de représentants à 101, 51 pour le projet ARK [235], alors que Bitshares [233] laisse le réseau choisir, avec toutefois un minimum fixé à 11. Par ailleurs, l'algorithme de consensus de Nano [119] peut être vu comme un hybride entre le PoS et DPoS. En effet, chaque entité possède un compte, auquel est associé un représentant : cela peut être le propriétaire du compte lui-même (configuration de PoS) ou une autre entité, choisie par ce dernier (configuration de DPoS). Dès lors qu'un conflit apparait entre des blocs, chaque représentant avertit ses homologues sur le bloc qu'il considère comme valide. À la fin du processus, le bloc qui aura obtenu le plus de voix sera accepté unanimement. À noter que le poids de vote d'un représentant dépend du nombre de nœuds l'ayant choisi comme tel. En revanche, le protocole Nano ne mentionne aucune méthode de récompense pour la participation des nœuds au consensus. Cela réduit drastiquement l'incitation et particulièrement la sécurité de manière générale. Si aucune motivation n'incite les nœuds à participer au consensus, c'est toute la pérennité du réseau qui est mise à mal [236].

Proof-of-Importance (PoI) [237]: il s'agit d'une version plus poussée du PoS, où chaque entité se voit attribuer une valeur d'importance en fonction de sa mise, mais aussi d'autres facteurs comme sa réputation, son activité (*e.g.* nombre de transactions émises/reçues), suivant les besoins de l'application.

Proof-of-Elapsed Time (PoET) [238]: outre les systèmes basés sur le PoS, le *Proof-of-Elapsed Time* est une autre alternative moins énergivore que le PoW. Comme dans ce dernier, les nœuds doivent résoudre un puzzle cryptographique, à la différence qu'au lieu d'être en compétition sur la génération du prochain bloc (ce qui exige une forte consommation énergétique), chaque nœud génère un nombre aléatoire qui définit son temps d'attente. Le gagnant, désigné pour publier le prochain bloc, est celui dont ce temps d'attente expire en premier. Bien que cette approche apparaisse comme idéale sur le plan énergétique, l'inconvénient majeur est qu'elle exige une méthode de certification sur les temps d'attente (*i.e.* comment s'assurer qu'un nœud à bel et bien tiré un nombre au hasard et a respecté son temps). Pour fournir cette preuve, le système a besoin d'un « environnement d'exécution de confiance » (*Trusted Execution Environment*, TEE), tels que le *Software Guard Extension* (SGX) de Intel [239] ou la *TrustZone* de ARM [240]. Cela implique d'une part que chaque nœud embarque ce type de technologie, mais plus important encore, la sécurité de l'algorithme de consensus repose sur la viabilité des solutions proposées par les constructeurs. Par ailleurs, certaines failles sur la plateforme SGX ont récemment été publiées [241]. De plus, Intel et ARM s'apparentent alors comme des tiers de confiance, ce qui est en inadéquation avec la philosophie d'un système blockchain.

Tangle [118] : constituant la base du projet IOTA [130], le *Tangle* n'est pas à proprement parlé un algorithme de consensus, mais une structure de graphe orienté acyclique (DAG). Toutefois, il inclut une procédure permettant l'ajout de transaction (pour rappel IOTA se compose uniquement de transactions et non de blocs). Chaque nœud doit valider deux précédentes (mais récentes) transactions avant d'émettre la sienne, ce qui constitue une excellente source de motivation à sécuriser l'écosystème. Ensuite, de la même manière que dans Nano, la transaction fait l'objet d'un PoW léger pour être considérée comme valide. Ce schéma de validation permet aux anciennes transactions d'être au fur et à mesure validées par les nouvelles, augmentant ainsi leur viabilité et donc leur niveau d'acceptation sur le réseau. Toutefois, la communauté s'accorde à dire qu'il est pour l'heure difficile de déterminer quand une transaction est considérée comme irréfutable (comme les six blocs dans Bitcoin). De plus, en raison de la faible difficulté du PoW employé, il est plus facile que dans Bitcoin pour un attaquant, d'obtenir suffisamment de puissance de calcul (majorité de la puissance de hachage du réseau). Bien que cela ne soit que temporaire, la fondation IOTA emploie un nœud spécial, appelé « coordinateur » dont elle a le contrôle. Son rôle est d'émettre régulièrement des transactions de références validant directement ou indirectement les précédentes. C'est à ce jour, la seule méthode permettant aux nœuds de s'assurer de l'irréfutabilité d'une transaction [242]. Comme l'explique le document officiel [243] et l'article [242], le coordinateur sera supprimé dès lors que la fondation estimera le protocole et le réseau suffisamment autonome pour assurer sa sécurité de manière autonome.

Practical Byzantine Fault Tolerance (PBFT) [244]: comme son nom l'indique, le protocole PBFT fait partie de la famille des consensus qui tente de résoudre le problème des généraux byzantins (*byzantine fault*), étudié en profondeur pour la première fois dans l'étude [245], remettant en cause la fiabilité des transmissions et l'intégrité des parties prenantes. L'objectif est de déterminer comment et dans quelle mesure une information peut être considérée comme fiable, dans un réseau où non seulement la source de cette information, mais aussi le canal de transmission peut ne pas être digne de confiance. En accord avec cette définition, le système blockchain est un parfait environnement pour ce type de problème. Le protocole PBFT est basé sur un processus de vote à cinq tours, impliquant tous les nœuds du réseau, afin d'atteindre un consensus. Lorsqu'un nœud souhaite ajouter un bloc, il transmet sa proposition à ses homologues pour validation. Ces derniers donnent leur accord ou désaccord par l'intermédiaire de votes émis sur le réseau. Si la majorité accepte la proposition, alors le bloc est accepté de manière unanime et définitive. Pour résoudre le problème des généraux byzantins, le protocole nécessite qu'au moins 2/3 des participants soient honnêtes, sans quoi le consensus ne peut être atteint. Le seuil de tolérance aux fautes est donc de 33%. L'avantage notable de ce système de votes est qu'il permet d'atteindre rapidement un consensus, sans nécessiter une forte consommation d'énergie et de puissance de calcul. En revanche, il requiert une forte bande

passante en raison de la transmission de tous les votes émis par les participants. Comme l'illustre [116], avec un réseau comportant seulement quatre nœuds, le protocole exige 32 interactions. Par conséquent, cette approche souffre d'un manque de scalabilité, en termes de nombre de participants. De plus, un système implique de connaitre la taille du réseau et d'identifier de manière unique chaque entité (contrôles des votes, estimation du seuil pour atteindre la majorité, etc.).

Delegated Byzantine Fault Tolerance (DBFT) [246]: de la même manière que le *Delegated Proof-of-Stake*, des représentants sont élus par les nœuds pour voter en leur nom et atteindre un consensus sur le prochain bloc à ajouter. Le DBFT suit le même protocole que le PBFT, mais cette fois, uniquement les représentants sont impliqués dans le processus de vote, ce qui améliore sa scalabilité, au détriment d'une plus forte centralisation. Comme précédemment, un bloc est définitivement accepté lorsque la majorité est atteinte, soit $N - f$ votes, avec N le nombre total de représentants honnêtes et f les fautifs/malicieux. Ainsi, le DBFT assure un consensus tant qu'il n'y a pas plus de $(N - 1)/3$ représentants malhonnêtes.

Tendermint [247]: considéré comme un protocole hybride associant le PoS et le PBFT. En effet, Tendermint repose sur un système de votes, pondérés par la quantité d'argent misée par les nœuds, devenant ainsi des « validateurs ». Ces derniers sont alors en charge de voter et d'atteindre un consensus sur le prochain bloc à ajouter. Contrairement au PBFT, les votes des nœuds ayant une mise plus conséquente auront plus de poids lors du dépouillement. De la même manière, un bloc est accepté de manière unanime et définitive lorsque 2/3 des validateurs l'approuvent, et tant que le nombre de validateurs n'excède pas les 33%.

Algorand [248]: portant également le nom de sa cryptomonnaie associée, le protocole de consensus d'Algorand a été développé pour répondre au problème de latence du PoW et de la scalabilité du PBFT et ses dérivés. Pour cela, un petit groupe de nœuds (*e.g.* 26 [248]) est sélectionné aléatoirement, avec une pondération sur la quantité d'argent qu'ils possèdent. La particularité de cette sélection est qu'elle s'effectue de manière privée, à l'aide d'une « fonction aléatoire vérifiable » [249] (*verifiable random function*, VRF). Chaque nœud exécute localement cette fonction, avec comme paramètres leur clé publique et privée. Le résultat lui indique s'il est sélectionné pour participer au processus de consensus. Puisque cette procédure est privée, la VRF fournit également une preuve, afin que les autres membres du réseau puissent s'assurer de la sélection légitime de leurs homologues. À ce stade, les nœuds du groupe publient leur proposition de bloc, dont leur probabilité d'acceptation par le reste du réseau dépend du poids monétaire de son créateur. Ensuite, un nouveau groupe, plus grand (*e.g.* 2000 [248]), est formé de la même manière avec cette fois pour but de voter les propositions précédentes. Dès lors qu'un bloc atteint la majorité (2/3 du groupe), un troisième groupe est formé afin de le certifier, en respectant toujours l'accord de la majorité. À la fin de ce processus, le bloc est

ajouté de manière définitive dans la blockchain. L'avantage majeur de ce protocole réside dans la sélection privée des membres du groupe. En effet, choisir un faible nombre de nœuds en charge de maintenir la blockchain permet certes d'améliorer la rapidité du consensus, mais au détriment de la sécurité : il est alors plus facile d'isoler ces nœuds (*e.g.* DoS) ou d'altérer la majorité (*e.g.* attaque Sybil). En ne révélant l'identité des membres du groupe uniquement lors de leur unique participation durant tout le processus, un attaquant est dans l'incapacité de prédire à l'avance les nœuds qui seront élus et donc de mettre en péril le système.

i) Finalité et fork

Plusieurs algorithmes de consensus ont été présentés tout au long de cette section de manière non exhaustive. En effet, il en existe autant que d'applications. Toutefois, ils peuvent être classés suivant deux catégories : les probabilistes, et les systèmes de votes. Comme son nom l'indique, cette dernière stipule que le consensus s'effectue de manière équivoque et immédiate, au travers de votes émis par les participants du réseau. Par exemple, le PBFT, DBFT, Tendermint et Algorand en font partie. À l'inverse, les algorithmes probabilistes n'affichent pas de collaboration explicite, dans le sens où l'acceptation unanime et définitive d'une information prend forme à mesure que les participants en ajoutent de nouvelles, basées sur les précédentes. Il est alors question de probabilité d'acceptation, qui augmente au cours de l'avancement du système. Par exemple, le PoW/PoS et leurs dérivés ainsi que le *Tangle* sont probabilistes.

Cette classification s'avère être primordiale, car elle souligne la caractéristique majeure d'un algorithme de consensus, à savoir sa finalité. Cette propriété désigne le temps nécessaire au protocole pour tendre vers un résultat définitif et irrévocable. Dans le contexte d'une blockchain, cela signifie quand et comment un bloc, ainsi que les transactions qu'il contient, est accepté unanimement et définitivement. La conséquence d'un consensus ayant une faible finalité (*e.g.* les probabilistes), conduit à l'apparition de ce qui est communément appelé un *fork*. Ce phénomène se produit lorsque plusieurs nœuds proposent simultanément un nouveau bloc à ajouter. Par exemple, soit les blocs *A* et *B* proposés en même temps. En raison des imperfections du réseau, ils n'arrivent pas dans le même ordre sur l'ensemble des nœuds. Ces derniers, dans l'attente du prochain bloc, vérifient le premier qu'ils reçoivent et s'il est valide, l'ajoute dans leur copie locale de la blockchain. Par conséquent, suivant l'ordre d'arrivée, deux versions de la blockchain (*fork*) se forment, comme illustrées sur la Figure 3-14.

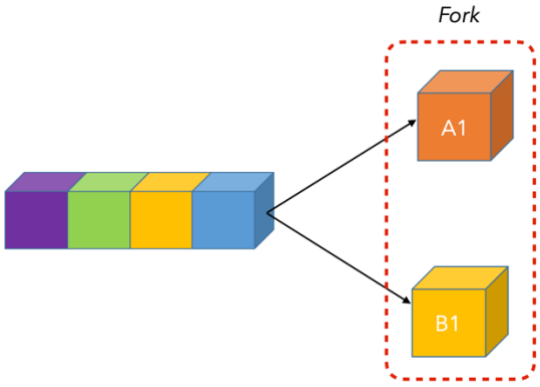


Figure 3-14 – Illustration d'un *fork* dans une blockchain

Quelle que soit la structure de la blockchain, l'algorithme de consensus doit impérativement intégrer une procédure capable de résoudre ce conflit. Dans le cas de Bitcoin et Ethereum, leur protocole définit que « la branche comportant le plus de calcul l'emporte ». Aussi simple soit-elle, cette règle est d'une efficacité remarquable. En effet, le processus de génération de bloc représente à lui seul la quasi-totalité d'utilisation des ressources de la blockchain. Par conséquent, « le plus de calcul » équivaut à « le plus de blocs ». Pour qu'une branche grandisse, il faut que les nœuds y ajoutent des blocs, et donc qu'elle soit légitime à leurs yeux. De ce fait, plus il y a de nœuds qui travaillent sur une même branche, plus les chances pour que celle-ci grandisse sont importantes. Ainsi, « la branche comportant le plus de calcul l'emporte », désigne implicitement celle la plus répandue sur le réseau et donc la plus légitime. La Figure 3-15 présente la résolution du *fork* selon cette règle.

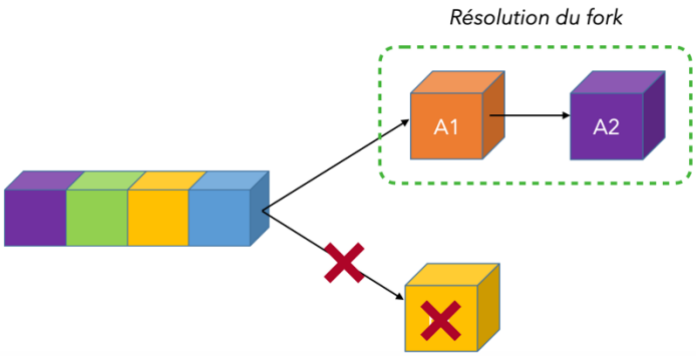


Figure 3-15 - Résolution d'un *fork*

Puisque tout est intimement lié, la finalité du consensus impacte celle des blocs et transactions. En effet, comme l'illustre la Figure 3-15, il est évidement que les transactions se trouvant dans la branche non conservée, seront supprimées. C'est pour cette raison qu'il est recommandé d'attendre plusieurs blocs afin d'être certain de l'acceptation définitive de la transaction au détriment de la

latence (souvent 6 blocs dans Bitcoin, ce qui équivaut à une heure [219], [226]). Ainsi, dans le contexte de l'IdO où le besoin de réactivité est primordial, un algorithme exigeant étant capable de gérer un grand nombre d'entités est préférable. En revanche, le délai trop important pour s'assurer de la viabilité d'une information n'est pas adapté.

Par définition, les algorithmes de consensus basés sur un système de votes sélectionnent explicitement et unanimement le prochain bloc à ajouter. De ce fait, l'apparition de *fork* est considérablement réduite, offrant par conséquent, une forte finalité de consensus, ce qui s'avère être idéal dans le cadre de l'IdO. Mais cet avantage n'est pas sans inconvénient. En effet, ils nécessitent un système d'identification unique et efficace des participants, afin de rendre les votes nominatifs pour éviter les multiples votes. De plus, ils doivent connaitre le nombre d'entités à l'avance afin d'estimer le seuil de majorité à atteindre. Enfin, puisque les nœuds prenant part au consensus doivent échanger les votes de chacun, cela implique d'intenses interactions entre les parties prenantes, d'autant plus importantes lorsque le nombre de participants augmente. Alors que l'approche probabiliste précédente ne souffre pas de ce problème de taille de réseau, les systèmes de votes souffrent d'une faible scalabilité de ce point de vue, avec une forte utilisation du réseau, impactant grandement l'énergie très limitée des objets connectés.

3.2.6. Synthèse

Instaurer la confiance dans un système, lui fournit les moyens d'assurer sa fiabilité, sa performance et sa sécurité, en détectant et isolant les entités non viables, défectueuses, voire malveillantes. La confiance est une notion complexe à appréhender, influencée par divers éléments, et dont la définition varie suivant les études. Globalement, elle peut être vue comme l'attente qu'une entité accomplisse une tâche comme prévu, évaluée au moyen de trois sources d'informations : la connaissance (*i.e.* compréhension d'une entité), l'expérience (*i.e.* interactions avec une entité) et la réputation (*i.e.* l'expérience acquise par d'autres entités). L'état de l'art contient différentes méthodes permettant d'estimer le niveau de confiance d'un acteur du réseau, alimentées par plusieurs propriétés comme la coopérativité, l'intérêt communautaire ou encore des notions plus facilement mesurables telles que l'énergie, la qualité du signal radio. Évidemment, la fiabilité d'un modèle dépend des données qui lui sont fournies. C'est pourquoi elles doivent être contrôlées, disséminées et stockées de manière sécurisée sur l'ensemble du système. Étant donné les contraintes de scalabilité (*e.g.* nombre d'entités, temps réel, etc.) et de fiabilité (sécurité, résistances aux pannes), une architecture décentralisée, s'avère être plus en adéquation pour un environnement d'IdO, qu'une approche centralisée.

Dans ce contexte, la blockchain apporte une solution particulièrement intéressante au besoin de sécurisation des données. Fonctionnant sur un réseau décentralisé, elle assure l'immuabilité, la disponibilité et la synchronisation des informations, répliquées en de multiples copies, la rendant ainsi résistante aux pannes. Parmi les plus connues, Bitcoin [114] et Ethereum [117], mais aussi Nano [119] et IOTA [118] sont toutes des implémentations de blockchain publiques, c'est-à-dire que le réseau et les données sont accessibles à tous. Cette approche étant sensiblement liée à ce qu'est un environnement d'IdO, l'étude de ces blockchains permet d'identifier les possibilités d'adaptation à ce dernier. Tout d'abord, il convient de sélectionner la meilleure architecture, à savoir quelles entités effectuent quels types d'interactions (*cf.* Architecture). Par définition, la blockchain a besoin d'un espace de stockage en constante augmentation, ressource très limitée dans les objets connectés. Il est donc primordial d'adapter la structure du stockage (chaine unique, graphe orienté acyclique ou blocs-tressés) mais également sa gestion (*cf.* Stockage). Au-delà de ces points certes essentiels, le cœur même d'un système blockchain, responsable à la fois de ses performances (*cf.* Scalabilité, Latence, Énergie), de sa viabilité (*cf.* Incitation) mais aussi sa sécurité (*cf.* Sécurité et vie privée), repose sur son algorithme de consensus. L'analyse de plusieurs d'entre eux (*cf.* Consensus), montre qu'il est inéluctable de faire des compromis entre ces divers éléments, comme illustrée sur la Figure 3-16. Dans le cadre de l'IdO, l'accent devra indéniablement être mis sur la scalabilité, la latence et l'énergie, avec les quelques milliards d'objets connectés prévus [9]. Enfin, la capacité d'intégrer des *smart contracts* [129] est une caractéristique particulièrement intéressante, ajoutant de l'intelligence et de l'autonomie au réseau. Toutefois, parmi les projets cités précédemment, seul Ethereum les supporte à ce jour. Un effort doit également être porté sur cet axe, pour profiter pleinement des opportunités qu'offre la blockchain.

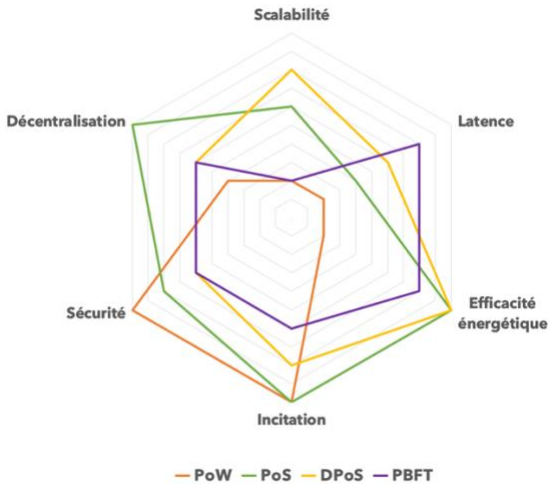


Figure 3-16 - Compromis sur la conception d'un algorithme de consensus (inspiré par [212])

3.3. Approche : un écosystème intelligent, autonome et sécurisé pour l'IdO

3.3.1. Passerelles intelligentes, point clé du réseau

Par définition, un écosystème d'IdO regroupe différents types d'objets, comme de simples capteurs de température, de mouvements, des actionneurs tels que des alarmes et des variateurs, ou encore des appareils plus complexes comme les smartphones. À cette pléiade d'acteurs s'ajoute une multitude de protocoles de communication (*e.g.* WiFi, LoRa, BLE, ZigBee), rendant cet environnement fortement hétérogène. Malgré leur disparité, ils partagent tous un but commun, interagir les uns les autres de manière autonome et sécurisée afin de s'échanger des données et créer des services.

Une architecture réseau centralisée peut être appréhendée en deux mondes distincts : les objets (*i.e.* capteurs/actionneurs) qui numérisent et agissent sur leur environnement ; le Cloud, qui endosse toute la puissance de calcul et de stockage. Toutes les données y sont alors centralisées, afin de le traiter et d'appliquer une réponse au système en conséquence. Les services, émanant de l'agrégation de ces données, sont donc endossés par le Cloud. Ce cheminement de l'information d'une extrémité du réseau à une autre implique une forte consommation énergétique (transport de la donnée de l'objet jusqu'au Cloud), mais aussi de potentielles failles de sécurité (exposition de la donnée). Dans ce modèle, représenté sur la Figure 3-17, les passerelles réseau (*gateways*) représentent le lien qui unit ces deux mondes, en relayant les informations des objets, souvent émises par signal radio, vers le Cloud, qui emploie le protocole standard TCP/IP.

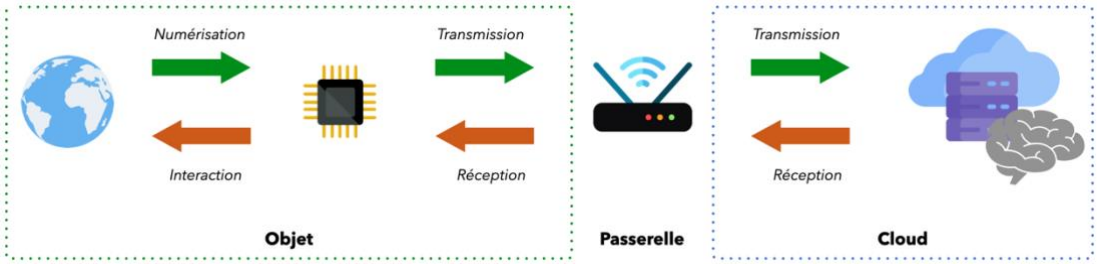


Figure 3-17 - Architecture réseau classique

Grâce à l'avancée de la microélectronique, le cout de l'intelligence et de l'espace de stockage, notamment en termes de surface silicium, est tel qu'il est désormais possible d'en doter les objets, mais aussi les passerelles. Ces dernières, déployant alors plus de capacités, deviennent alors un groupe à part entière, au même titre que les objets et le Cloud, comme le montre la Figure 3-18. Ce nouveau

modèle constitue le fondement même du *Fog Computing* [250]: tous les traitements et décisions principales sont pris par les passerelles, sans sollicitation du Cloud. Ce dernier n'est impliqué que pour traiter les données jugées nécessaires, améliorant ainsi la latence et la bande passante, mais également l'interopérabilité et la consommation d'énergie. De plus, puisque la quasi-totalité des données transite par les passerelles, désormais dotées de capacités accrues, elles sont capables de créer et supporter des services. En outre, puisque seules certaines informations transitent sur l'ensemble du réseau (de l'objet jusqu'au Cloud), l'exposition de toutes les données s'en trouve réduite, ce qui est intéressant d'un point de vue sécurité. Par exemple, l'étude [251] propose une preuve de concept d'une caméra connectée, dont l'objectif est d'identifier le sens de déplacement d'une personne (vers la gauche/droite). L'approche réseau précédente (*cf.* Figure 3-17) exige de transmettre l'ensemble du flux vidéo de la caméra, à un serveur pour analyse via la passerelle. Cela implique une forte utilisation de la bande passante (250 Mo/s) qui induit une latence, une consommation d'énergie et expose la vidéo complète sur le réseau, pouvant poser des problèmes de sécurité et de vie privée. En adoptant maintenant une architecture intelligente (*cf.* Figure 3-18), la passerelle est alors en mesure de récupérer, analyser et identifier elle-même les mouvements à partir du flux vidéo. Dès lors, elle peut transmettre uniquement le résultat (gauche ou droite) sur le réseau, n'exigeant qu'un seul octet de donnée. De plus, étant donné cette drastique réduction de donnée, l'information peut alors être transmise via des réseaux de faible consommation et bande passante limitée, tel que le LoRa. Enfin, la vidéo ne circulant plus qu'entre la caméra et la passerelle, le risque de sécurité et d'atteinte à la vie privée en est fortement réduit. Ainsi, l'ajout d'intelligence au plus près du capteur permet de raccourcir la chaine de communication, améliorant les performances, l'efficacité et la sécurité du réseau. Cette distribution des capacités constitue les bases d'une architecture réseau décentralisée, dont les passerelles en sont le cœur.

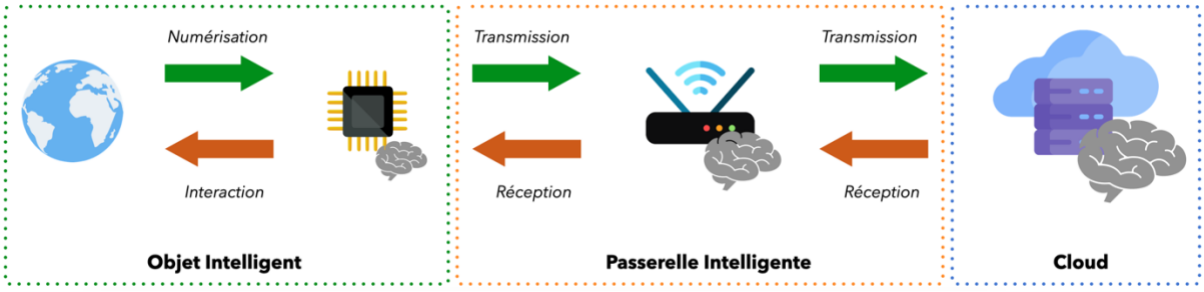


Figure 3-18 - Architecture réseau intelligente

3.3.2. Plateforme de prototypage d'un réseau d'IdO

Comme le soulignent les études [250] et [251], le rôle des passerelles s'élargit, en offrant de nouvelles capacités de calcul, de stockage et de sécurité, tout en effectuant les tâches d'origine tels que le routage réseau et l'interopérabilité des protocoles de communication. Cette particularité ouvre la voie d'un nouveau paradigme, centré sur la reconfiguration des passerelles, les dotant d'une meilleure flexibilité et augmentant leurs performances. Étant particulièrement intéressant dans un contexte d'IdO, une partie des travaux de cette thèse ont été dédiés à la réalisation d'une plateforme de prototypage d'un réseau de passerelles intelligentes. La première étape consiste à déterminer quel système serait le plus représentatif de ces dernières en termes de ressources (puissance de calcul, stockage et consommation d'énergie), tout en étant entièrement reconfigurable. Pour remplir cette mission, le choix s'est porté sur la carte Raspberry Pi3 Modèle B+ [189], constituée d'un processeur quatre-cœurs Cortex-A53 à 1.4 GHz, 1 Go de RAM et une capacité de stockage variable (supérieure à 256 Go, suivant la carte SD). La version actuelle de la plateforme, représentée sur la Figure 3-19, comporte 48 cartes Raspberry, connectées sur un même réseau, créer par deux routeurs TP-Link Archer C50 [252] (l'un en mode routeur, l'autre en point d'accès. 23 cartes Raspberry sont connectées en Ethernet par l'intermédiaire d'un switch Cisco Catalyst [253] et les 25 restantes en WiFi 2.4 GHz. Cette répartition offre l'avantage d'évaluer les performances suivant les types de connectivités (filaire vs. radio). Enfin, un ordinateur est également connecté, en guise de moniteur de contrôle pour interagir en temps réel avec la plateforme et effectuer les évaluations. Une vidéo de présentation de la plateforme est disponible ici (*Présentation-Plateforme-DATE2020*) et a été valorisée aux Rencontres Cybersécurité d'Occitanie 2020 (*cf.* Valorisations).

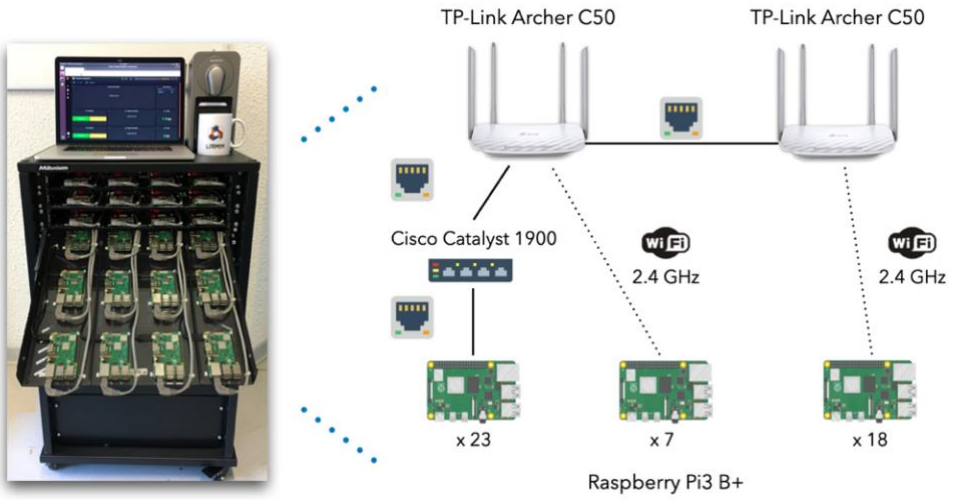


Figure 3-19 - Plateforme de prototypage d'un réseau d'IdO

3.3.3. Modèle de l'écosystème

Bien que les objets connectés puissent communiquer directement entre eux et sans intermédiaire (*i.e.* communication point à point), la possibilité d'élargir les interactions à plus grande échelle, c'est-à-dire via une connexion au réseau IP, représente une opportunité de taille, comme expliqué précédemment. Par ailleurs, puisque les passerelles sont en charge de récolter et d'analyser les données des objets pour fournir des services, la quantité d'interactions objets à passerelles s'avère être plus conséquente que celle d'objets à objets. De ce fait, une première approche dans cette thèse propose de se focaliser sur les interactions entre les passerelles, en mettant en place un modèle « économique » incitant chacune d'elles à coopérer, tout en assurant leur sécurité.

Puisque l'objectif d'un réseau d'IdO est de tout connecter afin de partager des données et créer des services, le principal défi est d'instaurer la confiance, sans tiers et de manière décentralisée. Le modèle proposé suit l'essence même du concept de l'IdO : toute entité peut intégrer le réseau pour accéder aux ressources/données et services disponibles, en échange d'une contribution à son développement et à sa sécurité. En effet, plus le système connecte de participants, plus il y a d'échanges de données, venant enrichir l'écosystème. La problématique est alors d'inciter le partage et d'estimer la contribution de chacun afin de contrôler ses demandes d'accès, et ce, de manière unanime et sécurisée. Par exemple, n'être qu'utilisateur du réseau, sans partager aucune donnée de valeur (*i.e.* qualité de service), la participation au développement du système est nulle. Une possibilité pour dissuader ce type de comportement, consiste à limiter les accès de l'entité en question, aux ressources/données/services du réseau. Ce dernier étant décentralisé, le contrôle des contributions doit faire l'objet d'un consensus et d'un stockage sécurisé, propice à un système blockchain, dont les passerelles en auront la charge, moyennant récompenses pour inciter sa pérennité. Ce modèle permet ainsi de mettre en place un écosystème à la fois équitable, durable et sécurisé. Toutefois, il convient de construire une blockchain adaptée aux contraintes de l'IdO et aux ressources limitées des passerelles.

3.3.4. Synthèse

L'ajout d'intelligence dans de nombreux systèmes, notamment au niveau des objets et des passerelles réseau, permet de traiter au plus proche les données, sans avoir recours à d'autres entités plus performantes. Ce gain de puissance de calcul, mais aussi de mémoire permet l'émergence d'un nouveau paradigme, le *Fog Computing*, pierre angulaire d'une architecture décentralisée. Bien que les performances et la sécurité en soient accrues, les résultats en sont notamment perçus au niveau des passerelles, ces dernières étant en charge du lien entre le monde des objets, souvent sans fil, et du réseau global IP.

L'IdO, interconnectant des milliards d'objets avec un fort besoin de réactivité, requiert une architecture efficace et sécurisée. Aussi importante et pertinente soit-elle, toute idée théorique visant à améliorer un système se doit d'être évaluée en conditions réelles. Dans cette optique, une plateforme de prototypage a été mise en place durant cette thèse, pour supporter les diverses approches d'un écosystème d'objets connectés. Afin d'instaurer la confiance dans un tel système, un modèle d'interactions doit être mis en place. Dans un environnement ouvert tel que l'IdO, l'incitation est une propriété primordiale, car elle régit à la fois l'évolution et la sécurité de ce dernier. Pour cela, le partage de données/ressources/services ainsi que la pérennité du système, doit être mis en avant, de manière sûre et décentralisée. Dans ce contexte, la blockchain se montre particulièrement intéressante. En revanche, cette dernière n'est actuellement pas prête à intégrer l'IdO, notamment en termes de puissance de calcul, de stockage et de consommation énergétique. C'est pourquoi un premier pas vers l'adaptation consiste à explorer différentes améliorations possibles, permettant la mise en œuvre d'une blockchain au niveau des passerelles, cœur d'une architecture distribuée et plus à même d'endosser une charge de travail que les objets.

3.4. Vers une adaptation de la blockchain pour l'IdO

3.4.1. Mise en place d'une blockchain classique

a) Mise en place du modèle d'IdO

Le modèle présenté précédemment propose d'évaluer la contribution de chaque passerelle, en charge de collecter les données provenant de leurs objets (capteur/actionneur), de les traiter si besoin, pour ensuite les propager sur le reste du réseau et/ou les utiliser pour créer des services. Lorsqu'il est question d'évaluer une donnée, plusieurs aspects peuvent être pris en compte, comme sa taille, sa précision, sa viabilité ou encore la valeur ajoutée qu'elle apporte dans l'écosystème. Toutefois, une première approche pour évaluer la contribution des passerelles, consiste à certifier de manière unanime, décentralisée et sécurisée, la quantité de données qu'elles produisent et partagent sur le réseau. En fonction de cette dernière, les passerelles peuvent accéder à différents types de ressources, services ou données. Bien que cette vision semble à première vue limitée, elle offre néanmoins une excellente preuve de concept permettant de valider ce type de modèle pour l'IdO, à savoir la monétisation de la donnée. Les différents aspects évoqués plus haut entreront progressivement en compte, afin de perfectionner le modèle.

b) Modèle de transactions de la blockchain

Avant tout, il convient de préciser que cette section traite uniquement les transactions en lien avec la blockchain et non celles de l'application IdO en elle-même (*e.g.* transmission de données capteurs).

Deux modèles de transactions sont utilisés dans les blockchains actuelles : les UTXO et les comptes/soldes (*Account/Balance model*) [128] comme Bitcoin [114] et Ethereum [117] respectivement. Les UTXO constituent un historique complet de transactions, les unes liées aux autres, offrant une traçabilité remarquable (*e.g.* Alice transfère à Bob, qui transfère à Carole, etc.). Toutefois, dans un contexte d'IdO, chaque donnée transmise n'est pas forcément dépendante de la précédente. Par exemple, la valeur de température d'une pièce ne dépend pas de sa valeur précédente. De plus, ce modèle requiert une conservation totale et définitive de toutes les anciennes transactions, pour s'assurer de la validité des suivantes. Étant donné l'intensité des communications attendues dans le contexte de l'IdO mais aussi l'espace de stockage limité des passerelles, l'approche UTXO ne semble pas adaptée. À l'inverse, un modèle de compte, contenant le solde actuel d'une entité, est plus flexible quant à la suppression d'anciennes transactions. En effet, bien que l'historique complet assure l'inviolabilité du solde, le lien entre les précédentes valeurs du compte est moindre que précédemment. Il peut donc être plus facilement envisagé de supprimer une partie de l'historique permettant ainsi une diminution de l'espace de stockage requis.

c) Génération de blocs et consensus, 1ère approche

En accord avec le modèle d'IdO choisi, une passerelle est récompensée en fonction de la quantité de données qu'elle produit/partage. Cela doit faire l'objet d'un consensus, afin que l'ensemble de l'écosystème ait la même vision. Comme dans le scénario des transferts d'argent d'Alice (*cf.* 3.2.2.c), effectuer cette procédure pour chaque transaction de donnée n'est pas envisageable, pour des raisons d'efficacité. Une solution simple à mettre en œuvre consiste à déterminer un seuil de quantité au bout duquel la génération d'un bloc sera déclenchée, ainsi que le consensus. Toutefois, en raison du nombre important de transferts de données dans un écosystème d'IdO, mais aussi de la latence du réseau, chaque passerelle ne reçoit pas toutes les informations en même temps ni dans le même ordre. Pour limiter ce phénomène, une légère modification à la solution proposée s'impose. Le seuil ne s'applique pas sur l'ensemble des données, toutes confondues, mais sur chaque groupe émetteur – type de données. Par exemple, pour un seuil fixé à 10, si les passerelles *A* et *B* publient respectivement 10 et 4 valeurs de température, seules celles de *A* seront prises en compte dans le prochain bloc. Celles de *B* seront traitées dans le suivant, si le seuil est atteint. Ce système exige donc de stocker toutes les transactions. Cependant, il convient de signaler que ce stockage est d'une part que temporaire (attente de confirmation du bloc), et potentiellement déjà présent pour le fonctionnement de l'application IdO embarquée.

À partir du modèle de ses transactions, un bloc contient les mises à jour des comptes des passerelles, sur lesquels un consensus doit être atteint. La transparence étant de rigueur dans une blockchain, chaque bloc doit contenir la justification pour toute nouvelle valeur des comptes (*e.g.*

justifier que « *A* a bien publié 10 valeurs de température »). De plus, les passerelles ne publient pas uniquement un seul type de données (*e.g.* température, pression, etc.), et il se peut que plusieurs d'entre elles atteignent le seuil simultanément. Encore une fois, à cause de la latence certaines entités du réseau peuvent être en décalage par rapport à d'autres. C'est pourquoi il est primordial d'indiquer précisément quels jeux de données sont pris en compte. Une méthode efficace consiste à appliquer une fonction de hachage (*e.g.* SHA-256 [225]) sur ces données, associées à l'identifiant de l'émetteur (*A*), et à la date de publication de chacune d'elles. Cela assure l'intégrité des informations, tout en constituant une preuve. La Figure 3-20 illustre la composition d'un bloc.

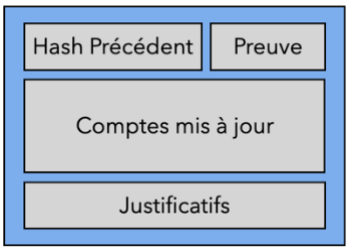


Figure 3-20 - Composition d'un bloc

L'empreinte du bloc est générée à l'aide du PoW [114], algorithme de consensus de Bitcoin, mais ici avec une faible difficulté dans le but de dissuader la publication de blocs frauduleux, comme le font Nano [119] et IOTA [118]. Pour rappel, cela consiste à calculer l'empreinte du bloc, en fonction de ses informations et d'un nombre aléatoire, de sorte que cette dernière respecte un format particulier. Par exemple, elle doit débuter par un certain nombre de 0. Cette opération a un coup en termes de puissance de calcul et d'énergie, qu'il convient de dimensionner pour être suffisamment important pour dissuader un comportement malveillant, sans être pénalisant pour le reste du système. Une fois l'empreinte calculée, le bloc est inscrit dans la blockchain locale de la passerelle et transmis sur le réseau.

d) Consensus, 2ème approche

Bien qu'une première approche basée sur le PoW léger soit intéressante, elle apporte son lot d'inconvénients, à savoir son manque de finalité. En effet, en raison des imperfections du réseau, plusieurs blocs créés en même temps, faisant donc référence au même prédécesseur (*i.e.* « hash précédent »), peuvent arriver dans un ordre différent aux passerelles, causant l'apparition de *forks* qu'il convient de résoudre (*cf.* 3.2.5.h)). Naïvement, il serait tentant d'augmenter le temps de génération d'un bloc, de sorte à réduire la probabilité que plusieurs d'entre eux soient publiés en même temps. Toutefois, cela implique une augmentation de la latence et donc une perte de réactivité du système. Une autre approche consisterait à synchroniser parfaitement chaque entité du réseau.

Malheureusement, cela s'avère être impossible, notamment sur vaste réseau tel que l'IdO [254]. En conséquence, comme dans Bitcoin [114] et Ethereum [117], la présence de *forks* engendre une latence de confirmation sur l'acceptation et sur l'ordre définitif des blocs, non appropriée à un contexte d'IdO. De plus, le caractère sécurisé du PoW réside dans la difficulté de génération des blocs, ce qui, aux vues des ressources limitées des passerelles, ne peut être très complexe. De ce fait, un autre algorithme de consensus doit être envisagé. Pour remédier à ces problématiques, un système de votes à la majorité permettrait, comme évoqué dans la section 3.2.5.h), de valider immédiatement un bloc. De plus, la sécurité serait cette fois liée à la taille du réseau, et donc au nombre de participants. En effet, plus il y a de votants, plus le seuil permettant d'atteindre la majorité est grand, rendant plus difficile sa manipulation, et augmentant la confiance dans le réseau. D'autre part, cela constitue un nouveau critère de motivation d'évolution de l'écosystème.

La première étape de cette nouvelle approche reste la même que précédemment : les passerelles créent un bloc contenant l'empreinte du précédent, les mises à jour de comptes, les justificatifs, et la solution du PoW (*cf.* Figure 3-20), avant d'être transmis sur le réseau. En revanche ici, le bloc fait l'objet d'un vote explicite, avant d'être ajouté à la blockchain. Il ne peut être accepté que lorsque la majorité des passerelles (*e.g.* 2/3 du réseau), donne leur accord, en publiant des blocs identiques. Le premier bloc valide et atteignant la majorité prend alors place dans la blockchain, de manière unanime et définitive. De plus, la liste de toutes les passerelles ayant voté un bloc est inscrite dans celui-ci, en guise de preuve de majorité. La nouvelle composition est illustrée par la Figure 3-21. Enfin, l'empreinte définitive du bloc est calculée, en prenant en compte toutes ses informations.

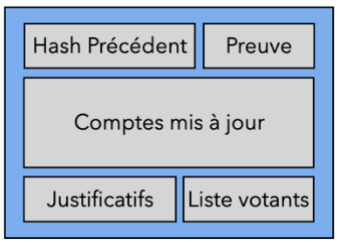


Figure 3-21 - Nouvelle composition d'un bloc

Lorsqu'il est question de majorité, cela sous-entend d'être en mesure d'estimer le nombre de participants présents dans le système. Grâce à son modèle de compte, l'estimation s'avère triviale, car il suffit de parcourir la blockchain et de procéder à un comptage. Une possibilité serait d'ajouter le nombre de participants total dans chaque nouveau bloc, ce qui permettrait plus facilement de retrouver le seuil de majorité. Toutefois, cela exigerait une utilisation supplémentaire de l'espace de stockage, dont l'impact augmenterait avec la taille de la blockchain.

Cette seconde approche a le mérite d'assurer la validation unanime et immédiate de chaque mise à jour de compte, résolvant a priori le problème de *forks*. En revanche, tel qu'il l'est décrit, ce protocole comporte quelques inconvénients. Tout d'abord, en raison de la latence de propagation sur le réseau, la liste des passerelles formant une majorité peut être différente. Par exemple, dans un système constitué de 6 participants (nommés de *A* à *F*), une majorité peut être représentée par les votes de *A B*, *C* et *D* ou bien par *C*, *D*, *E* et *F*. Plusieurs listes sont alors possibles, résultant sur différentes empreintes de bloc, menant à nouveau à l'apparition de *forks*, bien que moins fréquente par rapport à précédemment. Autre point important, intrinsèque à un système de majorité : tant qu'un bloc ne récolte pas suffisamment de votes, il n'est pas ajouté à la blockchain. Dès lors, cela signifie que le système peut rester bloqué indéfiniment si aucune procédure n'est prévue. Il convient donc de prendre en considération ces éléments afin d'élaborer un algorithme de consensus adapté et efficace.

3.4.2. Inciter à la participation et au respect des règles

Inciter les entités à respecter les règles est un point essentiel dans la conception d'un écosystème sécurisé, autonome et attractif, favorisant son développement. Le premier intérêt à intégrer un environnement d'IdO concerne bien évidemment l'accès à toutes ses données, ressources et services. Une entité verra ainsi ses opportunités multipliées. En revanche, puisque la sécurité et le contrôle du système dépendent de ses participants, il est important que ces derniers soient incités à y prendre part. Pour rappel, le modèle précédemment proposé (*cf.* section 3.3.3) stipule que les accès aux données/ressources/services sont régulés en fonction de la participation de chaque passerelle sur le réseau. De la même manière, il est possible d'évaluer leur implication dans la sécurisation du système. En effet, au travers de l'algorithme de consensus, chaque vote contient l'identifiant de la passerelle donnant son accord sur le prochain bloc. De cette manière, lorsque ce dernier atteint la majorité, il est trivial de retrouver la liste des entités ayant correctement voté, afin de les récompenser. À l'inverse, il est également possible de punir celles dont le comportement serait défectueux ou malveillant. Ainsi, chaque passerelle est incitée à partager ses données et à sécuriser son écosystème, rendant ce modèle particulièrement intéressant pour l'IdO.

3.4.3. Réduire l'utilisation de l'espace de stockage

Par définition, une blockchain contient les informations à sécuriser, ici les comptes des passerelles, répliquées sur l'ensemble du réseau. L'IdO s'apparente à un vaste écosystème, comportant plusieurs dizaines de milliards d'objets, générant une quantité colossale de données. Dans un système blockchain classique, cette activité entrainerait une création frénétique de nouveaux blocs, pouvant très rapidement saturer l'espace de stockage des passerelles. Comme vu précédemment, la suppression

des anciens blocs est séduisante, mais impacte fortement la viabilité et la sécurité de la blockchain, puisque la validation des nouveaux dépend des précédents.

Certaines techniques de compression de données peuvent réduire l'utilisation du stockage, toutefois cela ne présente qu'un moyen de le ralentir, ce qui en ne l'empêchera pas d'arriver à saturation. En complément, un découpage de la blockchain peut être envisagé. Une première approche consiste à ne répliquer les blocs que sur une certaine partie des nœuds et non sur tous comme initialement. Il en résulte un gain de l'espace de stockage, au prix d'une diminution de la décentralisation et de la sécurité. En effet, puisque certaines entités ne possèdent désormais qu'une partie des informations, elles ne peuvent participer qu'à la validation des blocs en lien avec ces dernières. De plus, en raison du lien cryptographique entre les blocs, le découpage peut s'avérer complexe et doit se faire scrupuleusement pour limiter les dégradations de la sécurité. Enfin, le seuil de réplication doit lui aussi être correctement défini pour assurer la disponibilité des données, lors des déconnexions des nœuds.

Une autre méthode reprend les concepts des *shards* [212], [213], [214] et des *sidechains* [215], [216], [217]. Bien que ces méthodes soient employées à des fins de scalabilité, l'espace de stockage local se voit également moins sollicité puisque seulement les informations des *shards*/*sidechains* sont traitées. Là encore, la décentralisation et la sécurité sont impactées. Quelles que soient les techniques employées, dès lors que des interactions ont lieu entre différents découpages, une procédure de synchronisation doit être effectuée pour assurer les contrôles. La complexité de ce processus peut mettre à mal le gain qu'apporte cette réduction temporaire de l'espace de stockage, au prix d'une plus lourde charge de calculs.

3.4.4. Synthèse

Cette section retrace les prémices d'une adaptation de la blockchain au contexte de l'IdO. En partant de ses principes fondamentaux, un modèle prônant le partage et le contrôle d'accès aux données/ressources et services semble pertinent. Une première approche dans sa mise en place consiste à évaluer la contribution de chaque passerelle, au travers de la quantité de données partagées, mais aussi de son implication dans la sécurisation de l'écosystème. Grâce à la blockchain, il est possible de certifier cette participation de manière unanime, décentralisée et sure. Pour cela, une première structure de bloc est proposée, comportant le lien cryptographique avec son prédécesseur, les transactions et leurs justificatifs basés sur le modèle de compte, ainsi que la solution d'un PoW léger. Ce dernier est employé afin de dissuader les comportements frauduleux et malveillants. La présence de *forks* est un problème majeur, exigeant une latence dans la confirmation des blocs. Afin de limiter leur probabilité, un algorithme de consensus à base de votes est finalement adopté. Ce dernier permet

également de cibler et récompenser précisément les entités ayant correctement participé, mais aussi celles qui ont tenté de nuire. Toutefois, cette méthode n'élimine pas entièrement la présence de *forks*, notamment à cause de la latence et de l'ordre d'arrivée des votes. La structure blockchain linéaire (à chaine unique) semble ainsi complexe à adapter aux contraintes de l'IdO, dont l'espace de stockage représente un frein majeur. Alors que certaines méthodes tentent seulement de ralentir leur utilisation, il est indispensable de limiter la taille de la blockchain, en limitant l'impact sur la sécurité. Au travers de ce cheminement d'une première tentative d'adaptation, plusieurs idées voient le jour, donnant naissance à un prototype, détaillé dans la section suivante.

3.5. Wallance, une alternative légère à la blockchain pour l'IdO

3.5.1. Principe

Cette section se propose de détailler le protocole Wallance, une alternative légère à la blockchain pour l'IdO, mis en place durant cette thèse. Fruit des investigations détaillées précédemment, l'objectif est de promouvoir le partage et l'utilisation de données/ressources et services, fondement d'un système d'IdO, tout en sécurisant les interactions des entités de manière décentralisée, et en prenant en compte les contraintes de stockage, de puissance de calcul et d'énergie.

À l'inverse de la majorité des études de l'état de l'art, Wallance utilise la blockchain comme la fondation de son écosystème. En accord avec le modèle de la section 3.3.3, et de sa première mise en place (*cf.* 3.4.1.a), ce protocole se focalise sur les interactions entre les passerelles, en monétisant la quantité de données partagées au travers du *DCoin* (*Data Coin*). Chaque nœud gagne une division de *DCoin*, le *DCoin Rate*, pour chaque partage. Cet argent virtuel leur donne le droit d'accès à de nouvelles données, ressources et services. Par exemple, louer le protocole radio de plus grande portée ou de la puissance de calcul d'une autre entité. Employant un protocole de consensus basé sur le vote majoritaire, le réseau est en mesure de contrôler toutes ces rémunérations et les accès. Enfin, pour encourager la sécurité de l'écosystème, un *DCoin Reward* est attribué aux nœuds ayant correctement voté. La Figure 3-22 met en image le positionnement de Wallance dans un écosystème d'IdO. À noter que les serveurs/Cloud sont des éléments à part entière du réseau, offrant des services plus complexes (*e.g.* analyses par intelligence artificielle, stockage, etc.) mais ne constituent en rien une autorité centrale.

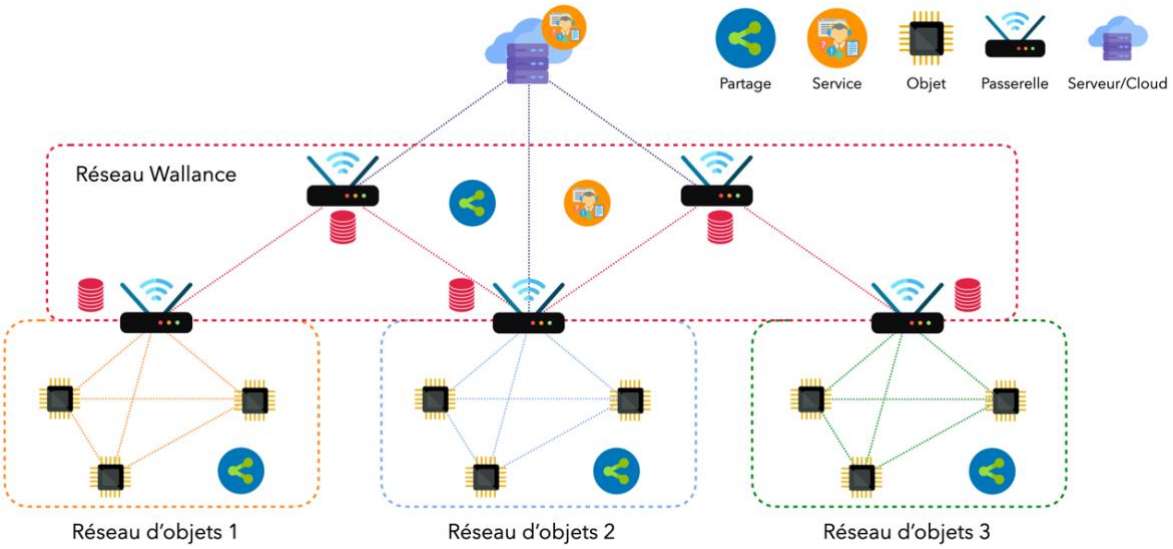


Figure 3-22 - Positionnement de Wallance dans un écosystème d'IdO

3.5.2. Structure

L'une des contraintes majeures de la blockchain est l'espace de stockage qu'elle requiert. En effet, cette dernière stocke l'historique complet des transactions depuis le tout premier bloc. Toutefois, cette approche n'est pas adaptée au contexte de l'IdO, composé de dizaines de milliards d'entités, ayant une mémoire très limitée. Pour remédier à cela, mais aussi aux problèmes liés aux *forks*, Wallance est constitué d'une matrice de blocs-tressés, inspirée de Nano [119] et du modèle de transaction de compte [128]. Comme le montre la Figure 3-23, chaque bloc, désigné sous le nom de portefeuille (*Wallet*), représente le compte d'une entité, contenant son identifiant, la quantité d'argent qu'elle possède (*DCoin*), ainsi qu'un *état*, empreinte du compte actuel, identique à celle d'un bloc d'une blockchain classique. Chaque nouvel *état* dépend de sa valeur précédente et de la nouvelle opération de l'entité, ce qui garantit l'intégrité du portefeuille. Ces derniers sont complètement indépendants, ce qui signifie qu'ils sont mis à jour de manière totalement asynchrone, sans impacter d'une quelconque manière les autres. Comme dans tout système blockchain, les nœuds doivent toujours stocker l'ensemble des portefeuilles du réseau, sans quoi les vérifications ne peuvent être effectuées. Pour pallier au problème de l'augmentation continue de l'espace de stockage, la structure de Wallance offre la possibilité de supprimer l'historique complet de chaque portefeuille et de ne conserver uniquement la dernière mise à jour. Toutefois, l'intégrité de chaque portefeuille au cours des modifications est continuellement assurée grâce à leur *état*. Ce choix est un compromis entre une sécurité de haut niveau des blockchains classiques et l'utilisation de l'espace de stockage limité dans l'IdO.

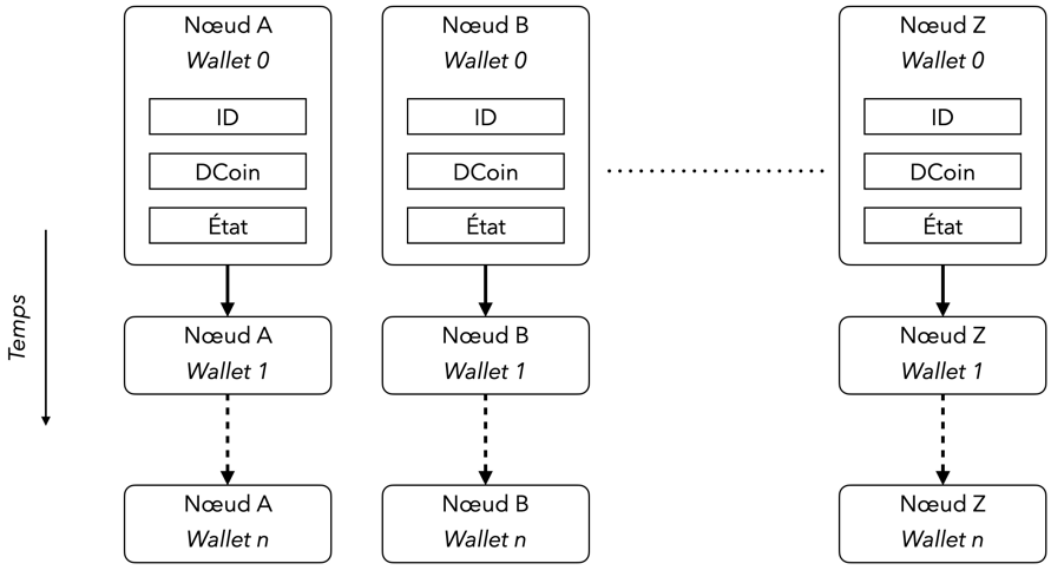


Figure 3-23 - Structure de Wallance

3.5.3. Transactions

Les nœuds gagnent le *DCoin Rate* pour chaque partage de leurs données, par le biais de transactions capteur (Figure 3-24.a). Aucune structure n'est imposée par le protocole Wallance pour ce type de transmission, fortement dépendante de l'application IdO. Toutefois, l'identifiant du nœud émetteur doit être spécifié sans quoi, le réseau ne sera pas en mesure de déterminer qui doit être récompensé. Ce partage de données représente plus de 80% de l'activité totale du réseau. Par conséquent, le processus de rémunération doit être le plus simple possible, sans intervention de l'algorithme de consensus. Puisque tous les portefeuilles sont dupliqués sur l'ensemble des nœuds du réseau, ces derniers peuvent effectuer les rémunérations adéquates, en mettant à jour les quantités de *DCoin* dans leurs copies locales.

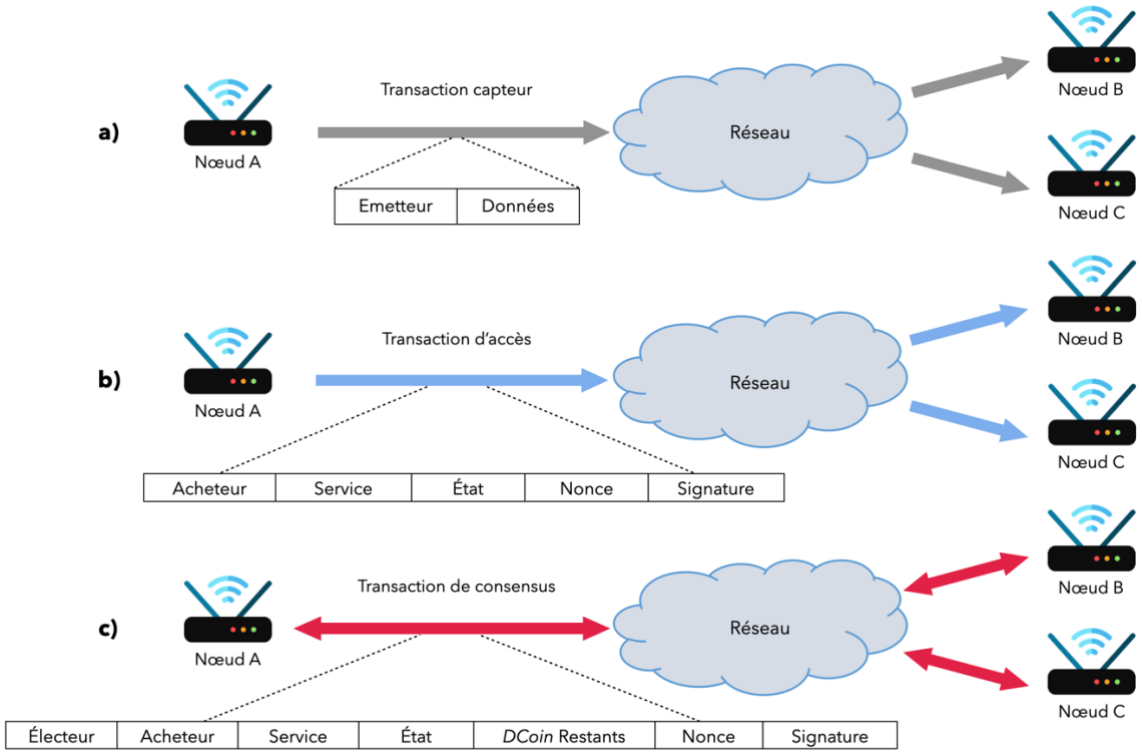


Figure 3-24 - Types de transactions du protocole Wallance

Pour accéder à d'autres données/ressources et services, les nœuds doivent transmettre leur demande à leurs homologues, au moyen d'une transaction d'accès (Figure 3-24.b). Cette dernière, étant soumise au vote via l'algorithme de consensus, doit être la plus légère possible afin d'optimiser les communications, tout en comportant suffisamment d'information pour assurer la sécurité. La transaction de consensus (Figure 3-24.c) est constituée de l'identifiant de l'acheteur et de l'achat en question (*i.e.* donnée/ressource/service). Pour éviter une attaque de double dépense, l'acheteur doit inclure l'*état* actuel de son portefeuille. Si plusieurs transactions d'accès se réfèrent au même *état*, comme sur la Figure 3-25, cela signifie que le nœud tente d'accéder à de multiples éléments en même temps, ce qui est similaire au scénario d'attaque. De ce fait, le protocole Wallance n'autorise qu'une seule demande d'accès par *état* du portefeuille. Il est cependant possible d'effectuer plusieurs demandes à la chaine, en calculant successivement les futurs *états*. Les nœuds doivent également inclure un *nonce*, solution du PoW léger nommé LWPoW (*Lightweight Proof-of-Work*) caractérisé par sa faible difficulté. Comme précédemment, son objectif est d'obliger toute entité à dépenser une certaine puissance de calcul, de temps et d'énergie pour dissuader les comportements malveillants. Toutes les transactions ayant un LWPoW incorrect ne sont pas prises en compte par les nœuds. Enfin, une signature numérique est apposée à la transaction, dans le but de certifier l'authenticité de l'émetteur, seul en mesure de générer cette preuve.

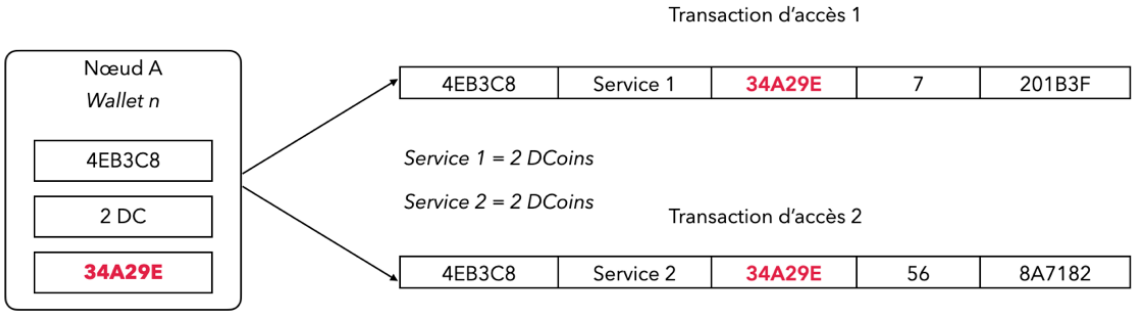


Figure 3-25 - Scénario de double dépense

3.5.4. Consensus

Conformément au principe de Wallance (*cf.* section 3.5.1), l'algorithme de consensus vise à contrôler les rémunérations et les demandes d'accès aux données/ressources/services, de manière unanime. Afin de respecter les contraintes de puissance de calcul, d'énergie et de réactivité de l'IdO, un système de votes est utilisé, où une rémunération/demande d'accès n'est validée que si la majorité du réseau donne son accord. Une fois atteinte, l'élément en question est définitivement validé, sans délai supplémentaire. Le seuil de majorité est fixé à 2/3 du réseau, soit le plus petit nombre théorique de participants honnêtes qu'un système de tolérance aux fautes byzantines (BFT) peut gérer [245]. Comme mentionné dans la section 3.2.5.h), la principale contrainte d'un tel modèle de consensus est sa scalabilité en termes de nombre de participants. En effet, plus le réseau est grand, plus le nombre de votes requis pour atteindre la majorité est élevé. Pour limiter cet impact et améliorer les performances du consensus, Wallance prévoit de n'exécuter ce dernier uniquement que pour des demandes d'accès, ce qui offre l'avantage de réduire au strict minimum les sollicitations du réseau dans l'atteinte d'un consensus. Évidemment, toutes les opérations de rémunération antérieures doivent également être certifiées. Pour se faire, une structure de vote légère (en termes d'octets) est mise en place, composée de toutes les informations nécessaires. La Figure 3-24.c illustre la composition d'un vote, *i.e.* transaction de consensus, et l'Algorithme 3-1 résume son processus de génération.

Entrée: TX_{Acces}

Sortie: TX_{Cons}

Si $TX_{Acces}.Signature$ **non valide alors**
  Suppression(TX_{Acces})

Si $TX_{Acces}.Nonce$ **non valide alors**
  Suppression(TX_{Acces})

Si $TX_{Acces}.Etat \neq Wallet_{Acheteur}.Etat$ **alors**
  Suppression(TX_{Acces})

Si $Wallet_{Acheteur}.DCoin \geq TX_{Acces}.Service_{Prix}$ **alors**
  $TX_{Cons}.Electeur = Noeud_{ID}$
  $TX_{Cons}.Acheteur = TX_{Acces}$.Acheteur
  $TX_{Cons}.Service = TX_{Acces}$.Service
  $TX_{Cons}.Etat = TX_{Acces}$.Etat
  $TX_{Cons}.DCoinRestants = Wallet_{Acheteur}.DCoin - TX_{Acces}.Service_{Prix}$
  $TX_{Cons}.Nonce$ = LWPoW(TX_{Cons})
  $TX_{Cons}.Signature$ = Sign(TX_{Cons})
  Suppression(TX_{Acces})

Algorithme 3-1 - Génération d'une transaction de consensus (vote)

Toute transaction de demande d'accès est en attente tant qu'elle ne reçoit pas la validation de la majorité. Pour qu'elle découle sur une acceptation et un vote favorable, il faut dans un premier temps que la signature et la solution du LWPoW soient valides. Vient ensuite la vérification entre l'état actuel du portefeuille de l'acheteur et celui inscrit dans la transaction. Enfin, la quantité de *DCoins* qu'il possède est évaluée pour déterminer sa capacité à acheter le service (ou les données/ressources). À noter que le prix du service fait partie intégrante de l'identifiant de ce dernier, afin d'éviter toute manipulation. Si toutes ces vérifications sont validées, alors le nœud génère son approbation, la transaction de consensus. Tout d'abord, l'identifiant de l'électeur doit être inscrit. Cela évite les votes multiples, comme par exemple en envoyant plusieurs fois le même pour d'altérer la majorité. S'ensuivent toutes les informations relatives à la demande d'accès faisant l'objet du consensus, à savoir l'identifiant de l'acheteur, du service (ou des données/ressources), ainsi que de l'*état* de son portefeuille. Outre ces éléments, la quantité de *DCoin* restante de l'acheteur après achat est également indiquée dans la transaction de consensus. Grâce à cette information supplémentaire, le réseau est en mesure d'atteindre un double consensus en un, à la fois sur les rémunérations et sur les contrôles d'accès. Un LWPoW est également exigé, dont la solution (*nonce*) doit figurer dans le vote. Cela permet notamment de dissuader les votes malicieux et d'éviter les attaques par rejeu (*cf.* 3.2.5.g). Enfin, l'électeur signe son vote pour assurer son authenticité, le transmet sur le réseau et supprime la transaction de demande d'accès traitée.

Le consensus de Wallance, représenté par l'Algorithme 3-2, s'efforce de comptabiliser les votes validant chaque demande d'accès. Dans un premier temps, les signatures et solutions au LWPoW des transactions de consensus sont contrôlées. Elles sont ensuite regroupées, suivant les informations de

la demande d'accès qu'elles valident (acheteur, données/ressources/service à acheter et *état*) ainsi qu'en fonction des *DCoins* restant estimés. Seuls les votes dont l'*état* du portefeuille de l'acheteur correspond à celui connu par le nœud sont traités. Les autres sont toutefois conservés, dans le cas où il s'agirait de la demande suivante, dont la validation aura lieu lors prochain consensus. Une fois qu'un groupe atteint la majorité (*i.e.* 2/3 du réseau), le portefeuille de l'acheteur est mis à jour. Tout d'abord, le nouveau solde en *DCoin* est fixé par l'estimation indiquée dans le vote (*DCoin Restants*). Cette valeur représente la certification de toutes les opérations de rémunération précédentes, tel que le partage de données capteur. Grâce à cela, l'ensemble du réseau se resynchronise sur le même solde. De là, le nouvel *état* du portefeuille est calculé. Chaque nœud récompense localement les électeurs ayant contribué à atteindre la majorité en leur attribuant le *DCoin Reward*. La synchronisation de leurs soldes se fera de la même manière qu'expliquée précédemment, lors du processus de consensus initié par leurs demandes d'accès. Enfin, tous les votes utilisés peuvent être supprimés et l'acheteur peut ainsi accéder à son gain. À noter qu'un service se présente sous la forme d'un binaire exécutable, se rapprochant des *smart contracts* d'Ethereum [129].

Entrée: TX_{Cons}
Sortie: $Wallet_{Acheteur}$
Si $TX_{Cons}.Signature$ **non valide alors**
  Suppression(TX_{Cons})
Si $TX_{Cons}.Nonce$ **non valide alors**
  Suppression(TX_{Cons})
$Groupe$ = Classement(TX_{Cons})
Si $Groupe.Comptage \geq Majorite$ **alors**
  $Wallet_{Acheteur}.DCoin = Groupe.DCoinRestants$
  $Wallet_{Acheteur}.Etat$ = Hash($Wallet_{Acheteur}.Etat$, $Groupe$)
  AttributionRécompense($Groupe.Electeurs$, $DCoinReward$)
  Suppression($Groupe$)
  Si $Noeud == Groupe.Acheteur$ **alors**
    Execution($Groupe.Service$)

Algorithme 3-2 - Processus de consensus

L'algorithme de consensus de Wallance est conçu pour éviter d'avoir recours à plusieurs tours de vote et les dépendances temporelles de nombreux algorithmes BFT. Ici, une fois qu'une opération de demande d'accès est validée par un nœud, celui-ci génère son vote unique, sans contrainte de temps. Puisque cette procédure ne requiert aucune autre étape, une fois transmis l'attaquant est dans l'incapacité de stopper la propagation du vote, ainsi que la finalité du consensus. Comme tout système de votes, l'algorithme proposé se base sur le fait que la majorité des entités du réseau soient honnêtes. Par exemple, avec un seuil fixé à 2/3, le système supporte jusqu'à 1/3 de participants défectueux et/ou malveillants. Cela est d'autant plus vrai pour éviter les *forks*. En effet, en reprenant

le scénario de la double dépense de la Figure 3-25, lorsque plusieurs demandes d'accès font référence au même *état* du portefeuille du demandeur, le réseau doit sélectionner celle qui sera finalement acceptée. Puisque l'ordre d'arrivée de ces demandes peut différer suivant les nœuds, le protocole Wallance stipule que dès lors qu'une entité reçoit une demande valide (signature, nonce, *état*, etc.), elle doit transmettre son vote. Pour rappel, ce dernier contient entre autres l'empreinte de la transaction de demande d'accès, qui correspond au futur *état* du portefeuille du demandeur. De ce fait, une fois le vote effectué, le nœud votant considère ce futur *état* comme acquis, bien que provisoire tant que la majorité n'est pas atteinte. En conséquence, lorsque la seconde demande d'accès lui parviendra, l'*état* auquel elle fait référence ne sera plus valide. Le nœud ne génèrera donc pas de vote. Autrement dit, une entité ne peut voter qu'une seule et unique demande à la fois par demandeur et par *état*. Cette simple procédure à l'avantage de favoriser la réactivité des votes, tout en supprimant définitivement les *forks*. La seule contrainte à cela est que la majorité des participants doit être honnête et donc respecter les règles. Dans le cas contraire, les nœuds pourraient voter pour plusieurs demandes d'un même nœud à la fois, menant à autant de majorités. En raison des caractéristiques du réseau, telle que la latence, les votes arriveraient dans un ordre différent suivant les entités. Ces dernières valideraient uniquement la première majorité constatée, sans pour autant que ce soit la même pour l'ensemble du réseau. Toutefois, l'intérêt d'un tel scénario est fortement limité grâce à la structure de portefeuilles-tressés, puisqu'un *fork* ne peut être causé que par le propriétaire du portefeuille, et n'impacte en aucun cas les autres. Chaque nouveau *fork* augmente peu à peu le clivage entre les nœuds, jusqu'à arriver au point où le réseau est tellement divisé sur une version d'un portefeuille, que celui-ci ne peut plus atteindre la majorité, stoppant alors toute activité.

3.5.5. Implémentations

D'après la section 3.3.1, les passerelles (*gateways*) constituent le point clé entre les réseaux sans fil et le réseau IP et disposent de ressources limitées, mais suffisantes pour pré-calculer les données. Afin d'avoir une représentation équitable, en termes de puissance de calcul, de capacité de stockage et de consommation d'énergie, la section 3.3.2 détaille la mise en place d'une plateforme de prototypage, constituée de cartes Raspberry Pi3 Modèle B+ [189] en guise de passerelles. Plusieurs implémentations sont proposées, toutes *open sources*, et disponibles en ligne sur le lien suivant (*https://gite.lirmm.fr/ldalmasso/WallanceProject.git*).

WallanceDDS : implémentation d'origine du protocole Wallance, WallanceDDS est codée en C/C++ et se base sur le modèle réseau *Publish-Subscribe*, plus approprié à un contexte d'IdO que l'approche client-serveur. En effet, le principe consiste à s'abonner à une donnée pour recevoir de manière automatique toutes mises à jour de celle-ci. Le client doit simplement attendre, à l'inverse du modèle client-serveur où il doit régulièrement solliciter le réseau pour s'informer d'une nouvelle

donnée disponible. Plusieurs protocoles sont disponibles tels que MQTT (*Message Queuing Telemetry Transport*) [255] et DDS (*Data Distribution Service*) [256]. L'étude [257] souligne toutefois que DDS offre une plus faible latence que MQTT, ce qui est souhaitable pour l'aspect temps réel de l'IdO. Plus en détail, le modèle DDS est composé de plusieurs fils d'actualité (*topics*), pour chaque type de données, spécifié suivant l'application (*e.g.* température d'une salle, détection de mouvement, etc.), auxquels s'inscrivent les entités du réseau. WallanceDDS utilise OpenDDS [258], une implémentation *open source* du modèle.

CWallance : cette version codée en C/C++ reprend la précédente, mais cette fois en utilisant la bibliothèque standard des *sockets*, plus légère, pour l'implémentation du réseau. Elle se possède des similarités avec OpenDDS, en exploitant les communications multicast.

pyWallance : dans le but d'élargir l'accès au protocole à un plus large panel d'acteurs de la communauté, pyWallance est l'homologue de CWallance codée en python, un des langages les plus utilisés.

Toutes les versions sont fournies avec un guide complet d'installation et d'utilisation. La suite de cette section se focalise sur la version CWallance, plus légère et efficace que pyWallance et WallanceDDS.

a) Architecture logicielle

Les cartes Raspberry embarquent une distribution Linux pour supporter l'architecture logicielle de Wallance, comme le montre la Figure 3-26. Par mesure de simplicité de mise en œuvre, un capteur virtuel est instancié afin de simuler la génération de données provenant d'objets. Ces dernières sont analysées par l'unité de traitement, avant d'être propagées sur le réseau par le transmetteur. Comme son nom l'indique, le récepteur est en charge d'écouter le réseau pour récupérer toutes les transmissions provenant d'autres entités. Afin de stocker les données de manière générique, compacte et flexible, une base de données est implémentée par la bibliothèque SQLite [259]. C'est elle qui regroupe à la fois la structure de portefeuilles-tressés, les transactions de demande d'accès en cours, et les transactions de consensus (vote). Côté cryptographique, le LWPoW est implémenté avec la fonction de hachage SHA-256 [225]. Afin d'être identifiable, un nœud doit générer une paire de clés publique/privée. Cette génération est effectuée suivant la même configuration de courbe elliptique que dans Bitcoin [114], à savoir les paramètres *secp256k1* du NIST [260]. La clé publique est utilisée en guise d'identifiant du nœud, alors que sa clé privée lui permet de signer numériquement ses transactions. Afin de tenir compte de la complexité de calcul des courbes elliptiques dans la génération des clés et de la signature ECDSA (*Elliptic Curve Digital Signature Algorithm*) [261], une implémentation légère de *secp256k1* est utilisée, nommée « micro-ecc » [262].

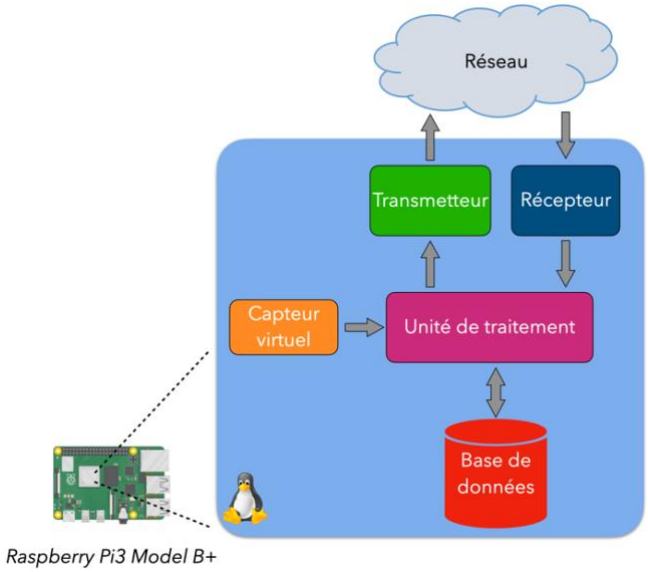


Figure 3-26 - Architecture logicielle de Wallance

b) Configurations de Wallance

Afin de concevoir un protocole flexible, l'utilisateur est en mesure de customiser différents paramètres de Wallance, listés dans le Tableau 3-4.

Tableau 3-4 - Liste des paramètres configurables de Wallance

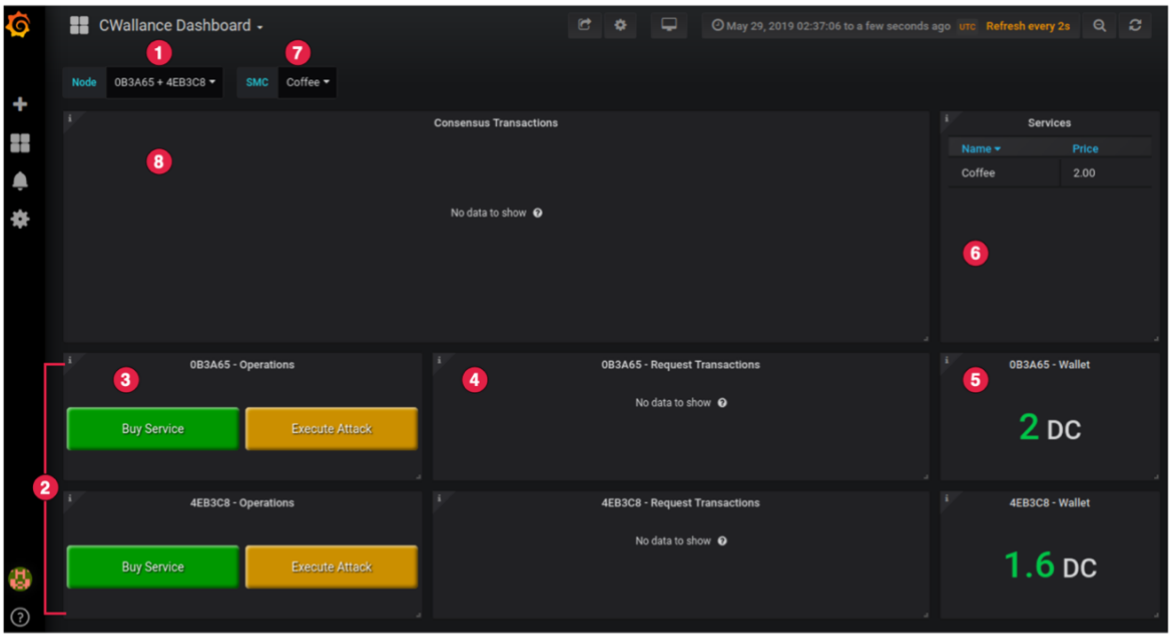
Paramètres	Descriptions	Impact sur le protocole
Seuil de majorité	Seuil minimum de vote requis pour atteindre un consensus	Niveau de sécurité (contrôle de la majorité) Latence de consensus (temps de récupération de tous les votes)
Difficulté	Nombre de '0' au début du hash du LWPoW	Surcharge de calcul Consommation d'énergie Latence de génération de transactions
DCoin Rate	Quantité de données, ressources, services à partager pour gagner 1 *DCoin*	Incitation à partager des données/ressources/services
DCoin Reward	Quantité de *DCoin* obtenue pour une participation correcte au consensus	Incitation à participer dans les règles au consensus et par la même, à la sécurité du protocole

L'implémentation proposée fixe un seuil de majorité à 2/3 du réseau, en référence aux protocoles BFT classiques. La difficulté du LWPoW est définie à 1, afin de limiter la charge de calcul et la dépense énergétique (*cf.* 3.5.6). Cela signifie que le hash doit commencer par un seul *nibble* à '0', soit les 4 bits de poids fort. Le *DCoin Rate* est fixé à 5, ce qui fait qu'un seul partage de

donnée/ressource/service équivaut à 0,2 *DCoins*. Pour promouvoir l'incitation à participer dans les règles au consensus, le *DCoin Reward* est élevé à 2 *DCoins*, soit l'équivalent de 10 partages.

c) *Interface utilisateur*

Dans le but de rendre l'utilisation du protocole Wallance plus intuitive et visuelle, une interface graphique est mise en place, basée sur Grafana [263], une plateforme d'hébergement web spécialisée dans la visualisation de données. De la même manière que les *blockchain explorers* [192], [193] et [194], cette interface permet d'afficher l'ensemble des portefeuilles du réseau, mais aussi d'interagir avec en temps réel, notamment pour déclencher des achats sur le compte d'un nœud, menant à la génération d'une transaction de demande d'accès. Enfin, il est possible d'injecter des attaques sur le réseau, afin d'évaluer la robustesse du protocole. La Figure 3-27 représente l'interface utilisateur de Wallance. Dans le cadre du démonstrateur proposé dans cette thèse, et dans l'optique de le rendre plus interactif, un service nommé « *Coffee* » est créé, que les Raspberry peuvent acheter pour préparer différents types de café.



(1): Sélection des nœuds à afficher
(2): Information des nœuds (1 nœud/ligne)
(3): Bouton d'achat d'un service / Générateur d'attaque
(4): Transactions de demande d'accès du nœud en cours
(5): Portefeuille du nœud (*DCoin*)
(6): Liste des services disponibles sur le réseau
(7): Selection du service à acheter
(8): Transactions de consensus reçues

Figure 3-27 - Interface utilisateur de Wallance

3.5.6. Résultats

a) Protocole d'évaluation

Avant de détailler les résultats de Wallance, il convient d'apporter quelques explications sur la procédure d'évaluation de ce dernier, et plus particulièrement sur les aspects énergétiques.

Chaque opération du protocole Wallance (initialisation, génération de transactions, consensus, etc.) fait l'objet de deux programmes de test. Le premier a pour objectif de mesurer le temps d'exécution de l'opération, de manière logicielle. La procédure est exécutée 1 000 fois, afin d'extraire avec fiabilité le temps minimal, moyen et maximal.

La puissance électrique induite par Wallance est mesurée à l'aide de la pince ampèremétrique PICOTECH TA018 [264], connectée sur l'alimentation d'une carte Raspberry, et reliée à un oscilloscope, pour visualiser le signal. Il est important de rappeler que Wallance fonctionne sur un OS Linux, imposant sa propre consommation. Cette dernière sera désignée par la suite sous le terme de « IDLE » c'est-à-dire sans charge de travail. Une simple remise à zéro de la pince ampèremétrique permet de s'absoudre de cette dernière, pour ne mesurer que la puissance requise par Wallance. Le rôle du second programme de test est d'assurer la synchronisation entre l'opération à évaluer, et la mesure de puissance. Là encore, la procédure est effectuée une dizaine de fois pour toutes les opérations du protocole, afin d'obtenir une puissance moyenne fiable.

À partir de ces résultats, il est possible d'en déduire l'empreinte énergétique de Wallance. Pour conclure, il est important de noter que tous les résultats détaillés ci-après correspondent à l'évaluation sur une carte Raspberry Pi 3B+, et non pas sur la totalité de la plateforme.

b) Construction des transactions

Grâce à la plateforme de prototypage, il est possible d'évaluer les performances de Wallance dans des conditions réelles. Tout d'abord, chaque interaction dans l'écosystème s'exécute au travers des transactions (*cf.* Figure 3-24). Comme vu dans la section 3.2.5.d), plus une donnée est imposante (en termes d'octets), plus elle met de temps à se propager sur le réseau, impactant la latence du système. De ce fait, les transactions et principalement celles de consensus (votes) doivent être les plus petites possibles afin d'éviter une surcharge de la bande passante réseau, tout en assurant la sécurité du protocole. En reprenant la composition des transactions de consensus de la Figure 3-24.c, les identifiants de l'électeur et de l'acheteur sont représentés par leur clé publique, soit 33 octets chacun (courbe elliptique *secp256k1* [260]). Il en est de même pour le service avec une subtilité, qui est l'ajout de 2 octets supplémentaires pour indiquer son prix. L'*état* est calculé à partir d'un SHA-256 [225],

soit 32 octets. S'ajoute ensuite la quantité de *DCoin* restante et le nonce, tous deux codés sur 2 octets. Enfin, la signature numérique en exige 64 supplémentaires [261]. Ainsi, la taille d'une transaction de consensus complète s'élève à 201 octets. Le détail est repris par le Tableau 3-5.

Tableau 3-5 - Taille d'une transaction de consensus

Éléments	Taille (octets)
Électeur	33
Acheteur	33
Service	33 + 2 (prix)
État	32
DCoin Restants	2
Nonce	2
Signature	64
Total	**201**

c) Opérations cryptographiques : LWPoW et ECDSA

Pour être considérée comme valide, chaque transaction fait l'objet d'un LWPoW, en guise de méthode anti-spam. Pour rappel, l'objectif est de trouver un nonce, permettant la création d'un hash commençant par un certain nombre de *nibbles* à '0', en fonction de la difficulté. Cette procédure représente l'opération la plus sollicitée par le protocole, qu'il convient de judicieusement calibrer. Le Tableau 3-6 détaille les résultats de l'évaluation du LWPoW de Wallance sur carte Raspberry Pi3B+, en indiquant le nombre d'itérations moyen et les temps de calcul associés pour résoudre le LWPoW, en fonction de la difficulté. Par ailleurs, le Tableau 3-7 présente les temps nécessaires à une Raspberry pour générer un seul bloc de chacune des blockchains de l'état de l'art, suivant leur difficulté actuelle.

Tableau 3-6 - Évaluation de la difficulté du LWPoW

Difficulté (*nibbles* à '0')	Nombre moyen d'itérations	Temps de calcul (Minimum)	Temps de calcul (Moyen)	Temps de calcul (Maximum)
0	1	37 µs	40 µs	95 µs
1	16	37 µs	652 µs	4,1 ms
2	263	37 µs	11 ms	99,6 ms
3	4 258	37 µs	183 ms	1,1 s

Tableau 3-7 - Évaluation du temps de génération d'un bloc des blockchains de l'état de l'art sur Raspberry Pi 3B+

Blockchain	Temps de calcul moyen
Bitcoin [114]	~7 milliards d'années[1]
Ethereum [117]	Impossible[2] [200]
Nano [119]	> 4 heures[1]
IOTA [118]	82,9 s [201]

1 : Évaluation effectuée dans cette thèse

2 : Sans disque dur externe hautes performances

Une difficulté de 0 correspond à une seule exécution du SHA-256 [225], d'une durée moyenne de 40 µs. Pour chaque niveau de difficulté supplémentaire, le temps de calcul requis est environ multiplié par 16 tous *nibbles* à '0' supplémentaires, ayant pour conséquence d'augmenter la latence de génération des transactions. En comparant ces résultats avec les principales blockchains publiques actuelles, il apparait clairement que les évolutions de ces dernières ont mené à de trop grandes difficultés, exigeant une forte puissance de calcul, incompatible avec l'IdO (*cf.* Tableau 3-7). À noter que comme Nano, la sécurité du protocole Wallance ne dépend pas directement de la difficulté du PoW/LWPoW comme Bitcoin, Ethereum ou IOTA. Cela permet de fixer un seuil de complexité moins important, réduisant le temps de résolution. Pour correctement déterminer la difficulté, il est primordial de comparer le temps moyen de résolution du LWPoW avec la latence des transmissions sur le réseau. La caractérisation de cette dernière est réalisée de la manière suivante : une carte Raspberry émet une transaction de consensus (201 octets), et mesure le temps entre l'envoi et la réception de cette dernière. Ce test est effectué 10 000 fois, de sorte d'obtenir une estimation fiable. Grâce sa petite taille, la transaction met en moyenne 33 µs pour faire l'aller-retour avec une connexion Ethernet, contre 52 µs en Wi-Fi. À noter que cette valeur dépend également des capacités des routeurs (TP-Link Archer C50 [252]). En se référant aux résultats du Tableau 3-6, les difficultés 1 et 2 imposent respectivement un temps moyen de génération d'une transaction 20 et plus de 300 fois supérieur à celui d'une transmission par Ethernet, contre 12 et 200 fois plus via Wi-Fi. Ces résultats se montrent intéressants pour ne pas mettre en péril la réactivité du protocole Wallance. De plus, l'impact énergétique qui en découle doit être pris en compte. La Figure 3-28 illustre l'évolution de la consommation en fonction de la difficulté.

	LWPoW (Diff 0)	LWPoW (Diff 1)	LWPoW (Diff 2)	LWPoW (Diff 3)	Vérification LWPoW	Signature	Vérification Signature
Énergie RPi IDLE (mJ)	0,07	1,30	22,2	366	0,07	21,4	24,2
Énergie LWPoW (mJ)	0,006	0,1	1,7	27,4	0,006	1,8	2,0

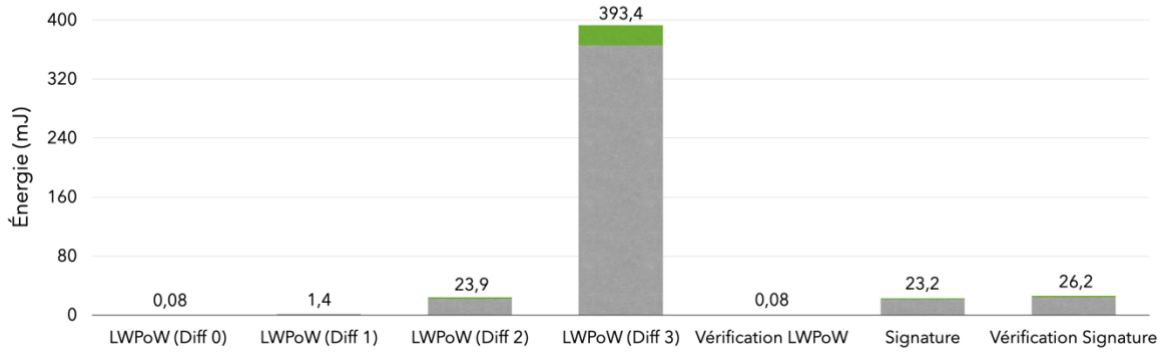


Figure 3-28 - Évaluation de la consommation énergétique du LWPoW et ECDSA (*secp256k1*)

Comme attendu, l'énergie mise en jeu par le LWPoW suit la même tendance que précédemment, et représente environ 8% de la consommation totale de la carte Raspberry. Quant à la procédure de vérification du LWPoW, sa consommation est constante, puisqu'elle ne requiert qu'une seule itération du SHA-256. Ainsi, une difficulté de 1 est dans un premier temps envisagée dans cette thèse, pouvant nécessiter un temps de résolution jusqu'à environ 4 ms, soit jusqu'à plus d'un facteur 100 sur la latence de transmission, associée avec une consommation énergétique relativement faible (1,4 mJ) par rapport aux autres configurations étudiées.

Outre le LWPoW, les transactions de demande d'accès et de consensus intègrent également une signature numérique, assurant leur authenticité. De ce fait, cette dernière doit être générée puis vérifiée lors de chaque transmission. Pour cela, Wallance utilise l'algorithme ECDSA avec la courbe elliptique *secp256k1* (code source provenant du projet « micro-ecc » [262]). Le temps moyen de génération de la signature et de sa vérification est de l'ordre de la dizaine de millisecondes, équivalent à celui du LWPoW de difficulté 2. Il en est de même pour la consommation énergétique reportée sur la Figure 3-28, validant ainsi l'adéquation de l'algorithme de signature dans le cadre de Wallance.

À des fins de comparaisons, il est intéressant d'opposer énergétiquement les configurations du LWPoW avec les algorithmes des blockchains de l'état de l'art, notamment IOTA [118] et Nano [119], étant les projets les plus légers en termes de ressources. Pour que cette comparaison soit pertinente, il convient de fixer un même seuil de difficulté pour chacun d'entre eux :

Wallance LWPoW : le principe est de trouver un hash de la transaction, via la fonction SHA-256 [225], commençant avec un certain nombre de *nibbles* (4 bits) à '0', suivant le niveau de difficulté.

Nano : le PoW de Nano consiste à trouver un hash de la transaction à l'aide de la fonction Blake2b [204], convertir le résultat sur un entier sur 64 bits et de le comparer au seuil de difficulté, lui-même un entier de 64 bits (actuellement 0xFFFFFFF800000000). Pour que l'opération soit valide, le résultat doit être supérieur ou égal au seuil. Le protocole Nano utilisait précédemment une difficulté de 0xFFFFFFC000000000 [265]. Il est alors possible d'extrapoler la valeur minimale à 0xF000000000000000.

IOTA : l'encodage des transactions d'IOTA est basé sur la logique ternaire (0,1,-1). Le principe de son PoW est similaire aux précédents, c'est-à-dire, trouver un hash de la transaction se terminant avec un certain nombre de *trits* à '0', en utilisant la fonction Curl-P [266], développée par la fondation IOTA elle-même. Avec une difficulté actuelle de 14, il est trivial de définir le seuil minimal à 4 pour concorder avec les configurations minimales de Nano et du LWPoW ci-dessus.

À défaut d'évaluation disponible dans l'état de l'art, le PoW du protocole Nano a fait l'objet d'une étude concrète sur carte Raspberry Pi 3B+ dans cette thèse. Quant au PoW d'IOTA, les résultats de l'étude [201] ont été utilisés. Toutefois, ces derniers ont été obtenus sur le modèle Pi 3B, exploitant 3 cœurs en parallèle. Pour que la comparaison soit des plus justes, plusieurs ajustements sont effectués pour approximer les résultats sur le modèle 3B+. En premier lieu, la résolution du PoW est adaptée sur un seul cœur, ce qui, en extrapolant le pire cas, multiplie par 3 le temps de calcul, soit 248,7 s au total (82,9 s × 3). D'après le benchmark officiel [267], le modèle Pi 3B+ offre un gain de rapidité d'environ 0,85 (*SysBench CPU Single-Threaded*) donnant lieu à un temps de résolution total de 213 s. Selon [268], chaque réduction de la difficulté d'IOTA (*Minimum Weight Magnitude*, MWM), diminue d'un facteur 3 la complexité, et par conséquent le temps calcul. La Figure 3-29 illustre les différents résultats.

	Nano	IOTA	LWPoW (Diff 1)
Énergie RPi IDLE (mJ)	1,1	7,2	1,3
Énergie PoW/LWPoW (mJ)	0,1	0,5	0,1
Temps de résolution	541 µs	3,6 ms	652 µs
Nombre moyen d'itérations	16	-	16

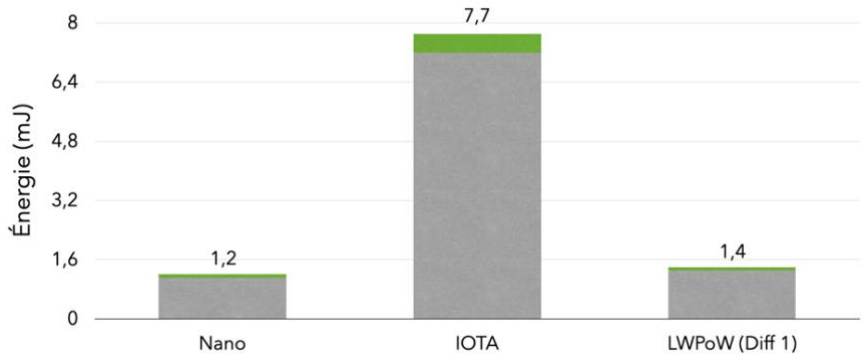


Figure 3-29 - Comparaison énergétique du LWPoW avec les PoW de Nano et IOTA à difficulté équivalente

Alors que le PoW d'IOTA se montre entre 5 et 6 fois plus énergivore, les résultats soulignent l'efficacité du protocole Nano, légèrement supérieure au LWPoW proposé. Cette différence s'explique principalement par la rapidité de la fonction de hachage de Blake2b [204] (Nano) par rapport au SHA-256 [225] du LWPoW. Bien que la puissance consommée soit plus faible pour ce dernier, le gain en temps de calcul de Blake2b est suffisant pour être plus efficace énergétiquement. Le choix du SHA-256 dans le protocole Wallance est guidé par la recommandation du NIST [225] ce qui a démocratisé son utilisation, comme c'est le cas dans le protocole Bitcoin. Par ailleurs, l'algorithme Blake2b fait partie des finalistes, mais non gagnants, de la compétition SHA-3 [269] organisée par le NIST pour succéder à la famille SHA-2. Toutefois, le concepteur de Nano n'apporte pas de précision sur son choix, stipulant uniquement dans le document officiel [119], qu'étant donné l'utilisation anti-spam du PoW, la sélection de la fonction de hachage n'a que peu d'importance, par rapport aux protocoles où l'algorithme assure l'inviolabilité de la blockchain (*e.g.* Bitcoin et Ethereum). Ainsi, étant donné les performances par rapport à l'état de l'art, le LWPoW avec une difficulté de 1 apparait comme pertinent pour limiter les spams tout en assurant l'intégrité du protocole Wallance.

d) Opérations protocolaires

Après avoir détaillé la composition des transactions et évalué les opérations cryptographiques LWPoW et ECDSA, cette sous-section s'efforce de poursuivre la cartographie énergétique de chaque

procédure du protocole Wallance, fonctionnant sur une carte Raspberry Pi 3B+. Les résultats sont présentés sur la Figure 3-30.

	Initialisation	Attribution DCoin	Génération TX Accès	Ajout TX Accès	Génération Vote	Ajout Vote	Consensus (Dépouillement) Réseau de 8 RPis	Consensus (Dépouillement) Réseau de 48 RPis
Énergie RPi IDLE (mJ)	348	64,6	26	68	84	60	130	150
Énergie Wallance Op. (mJ)	46,1	8,5	2	8,5	10,1	6,3	16,2	18,8
Temps	174 ms	32,3 ms	13 ms	34 ms	42 ms	30 ms	65 ms	75 ms

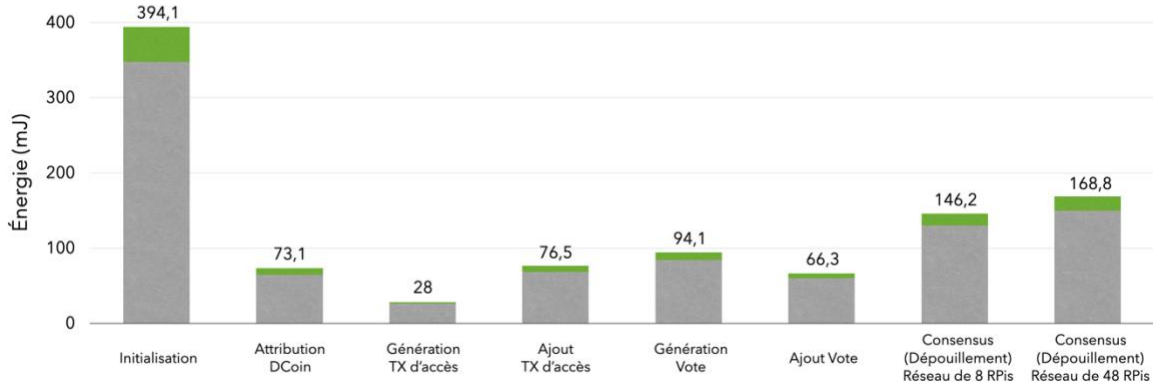


Figure 3-30 - Évaluation de la consommation énergétique des opérations du protocole Wallance

L'initialisation, comprenant la création de la base de données et la génération de la paire de clés publique/privée, constitue l'opération la plus énergivore. Toutefois, comme son nom l'indique, elle n'est exécutée qu'une seule fois, lors du premier démarrage du nœud. Bien que leur rôle soit similaire, les générations de transactions (demandes d'accès et votes) ainsi que leur ajout dans la base de données ne requièrent pas toutes les mêmes besoins énergétiques. Cela s'explique par les différentes étapes qui les composent. Par exemple, alors qu'il ne suffit uniquement d'extraire l'*état* du portefeuille à inclure dans la transaction de demande d'accès, la génération d'un vote quant à elle, requiert plusieurs accès à la base de données (*cf.* Algorithme 3-1). Il en est de même pour les procédures d'ajouts. Toutefois, elles ont en commun le calcul du LWPoW (difficulté 1), la signature numérique ainsi que leur vérification. Ces opérations cryptographiques sont responsables d'environ 87% et 26% de la consommation totale de la génération de transaction de demande d'accès et des votes respectivement. Puisque les vérifications sont moins complexes, notamment celle du LWPoW, ces parts diminuent entre 34% et 39% lors des procédures d'ajout des transactions dans la base de données. Ceci met une nouvelle fois en évidence l'intérêt d'une configuration judicieuse du LWPoW.

La seconde opération la plus consommatrice en énergie est le processus de consensus. En effet, cette procédure demande de parcourir la base de données afin de comptabiliser les votes, de mettre à jour le portefeuille de l'acheteur et ceux des nœuds ayant correctement participé, et enfin, supprimer toutes les transactions utilisées. Par ailleurs, il apparait une légère augmentation de la consommation

d'énergie, lorsque le nombre de nœuds actifs est plus important. En effet, cela provient principalement de l'attribution de la récompense (*DCoin Reward*) à chaque électeur pour leur participation, qui demande plus de temps en fonction de leur nombre. La Figure 3-31 donne une estimation de l'empreinte énergétique d'un tour complet du protocole Wallance fonctionnant sur une carte Raspberry Pi 3B+ : chaque nœud partage une donnée capteur pour lequel il est rémunéré, donnant lieu à la création de son portefeuille. Une transaction de demande d'accès est ensuite émise, déclenchant ainsi le processus de consensus (émissions puis traitement des votes). Plusieurs tailles de réseau sont étudiées, notamment celles permises par la plateforme de prototypage (de 8 à 48 nœuds).

	Taille du réseau Wallance					
	8 RPis	**16 RPis**	**24 RPis**	**32 RPis**	**40 RPis**	**48 RPis**
Énergie RPi IDLE (J)	1,8	2,6	3,4	4,3	5,1	5,9
Énergie Wallance (J)	0,2	0,3	0,4	0,5	0,6	0,7
Temps	0,9 s	1,30 s	1,71 s	2,15 s	2,56 s	2,98 s

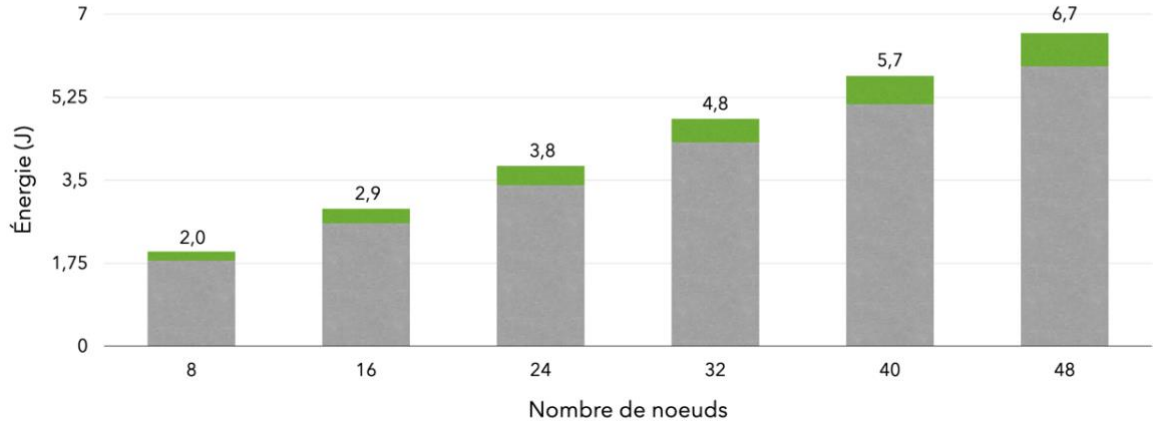


Figure 3-31 - Estimation de la consommation énergétique totale d'un nœud Wallance

Comme attendu, l'empreinte énergétique d'un tour complet de Wallance augmente en fonction du nombre de nœuds, en raison du traitement de plus nombreuses transactions (capteurs, votes). Avec un réseau maximum de 48 participants, la consommation d'un nœud s'élève à moins de 7 J, soit un total d'environ 322 J (6,7 × 48) pour l'ensemble de la plateforme. Afin de donner une dimension plus concrète à cette évaluation, la Figure 3-32 oppose la consommation énergétique requise pour la validation d'une seule transaction de demande d'accès (réception, stockage, génération de vote et consensus), avec d'autres systèmes existants. La Figure 3-33 quant à elle, oppose la consommation de Wallance avec d'autres systèmes électroniques sur une période d'une heure de fonctionnement continu.

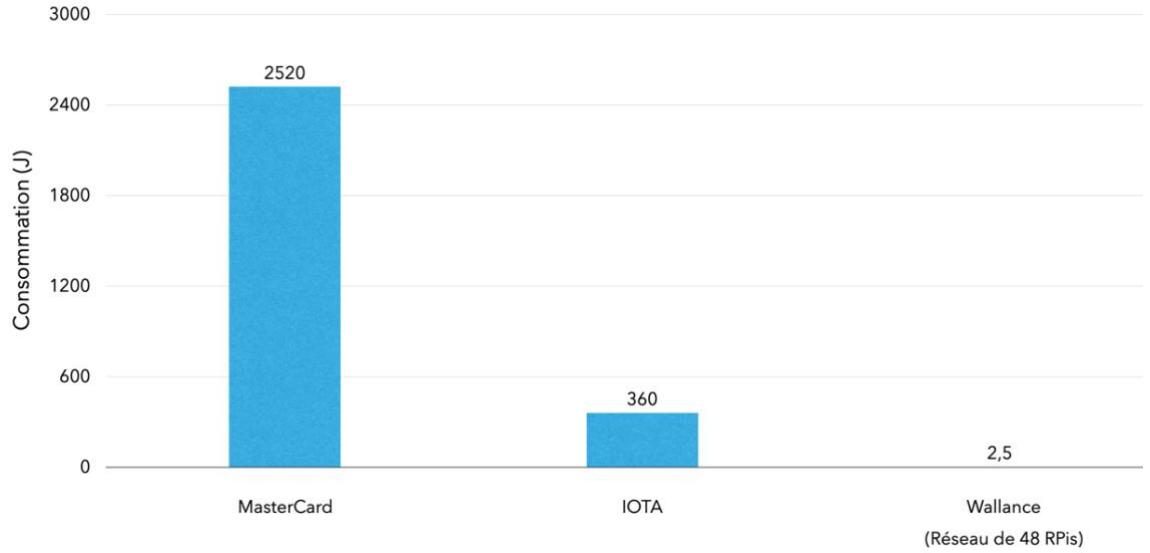


Figure 3-32 - Comparaison de l'énergie requise pour la vérification d'une transaction suivant différents services existants [198]

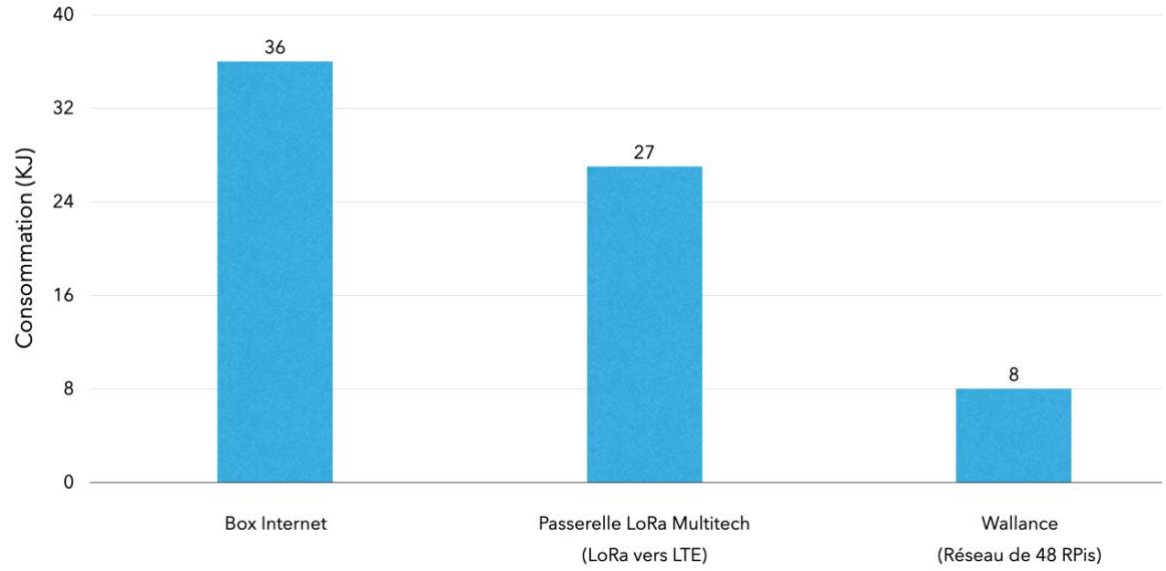


Figure 3-33 - Comparaison de l'empreinte énergétique de Wallance avec d'autres systèmes sur 1 heure de fonctionnement [270], [271]

Les résultats soulignent que l'empreinte énergétique nécessaire à la vérification d'une seule transaction du protocole Wallance est plus de 100 à 1 000 fois inférieure à celle des systèmes IOTA, et MasterCard. De même, la Figure 3-33 montre que l'utilisation de Wallance durant une heure de fonctionnement, pouvant théoriquement traiter et atteindre un consensus pour près de 1 300

demandes d'accès, requiert environ 3 fois moins d'énergie qu'une passerelle LoRa disponible sur le marché [271]. Ainsi, l'évaluation énergétique de l'implémentation proposée de Wallance sur carte Raspberry se montre pertinente pour le contexte de l'IdO.

e) Latence du consensus

Outre les aspects de puissance de calcul et de consommation énergétique, la difficulté du LWPoW impacte grandement la latence du protocole. En effet, puisque l'algorithme de consensus de Wallance est un système de votes, sa vitesse de confirmation dépend d'une part de la rapidité de génération des transactions, mais aussi du nombre de nœuds participant au consensus. Plus ils sont nombreux, plus il faut de votes pour atteindre la majorité, impliquant une latence plus importante. Pour quantifier cette dernière, la plateforme est configurée suivant différentes tailles de réseau, afin de mesurer l'impact réel de l'augmentation du nombre de participants connectés. Les résultats sont illustrés par la Figure 3-34. Pour rappel, la difficulté du LWPoW est fixée à 1.

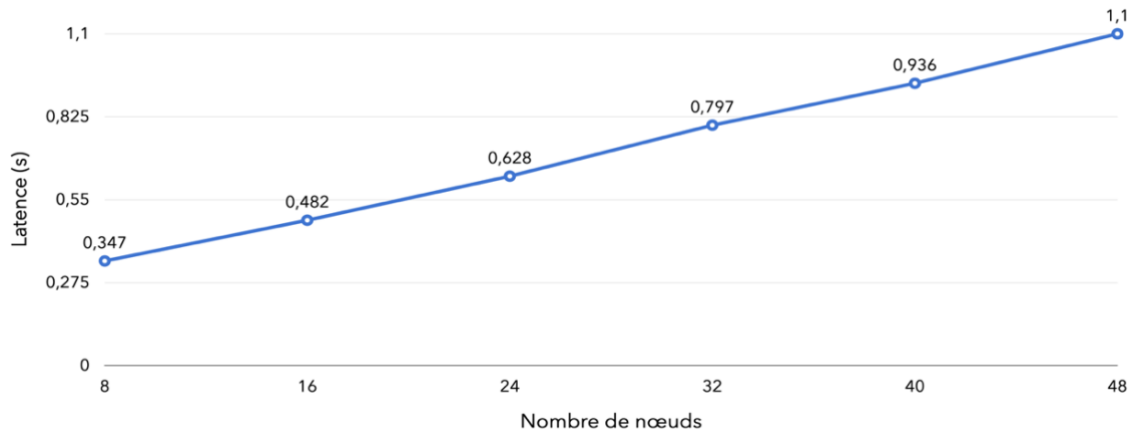


Figure 3-34 - Évolution de la latence du consensus de Wallance en fonction du nombre de nœuds

Avec un total de 48 cartes Raspberry, le protocole Wallance atteint un consensus de l'ordre d'une seconde, et ce, de manière définitive (*i.e.* sans *fork*). À noter que la latence dépend en partie des capacités réseau des routeurs (*e.g.* bande passante, connexion filaire ou sans fil, etc.). En effet, si le nombre de nœuds ou la quantité de données sont trop importants par rapport aux capacités de l'infrastructure réseau, il en découlera une latence, voire une perte de données, impactant alors la finalité de consensus. Bien sûr, une comparaison directe avec Bitcoin, Ethereum, Nano et IOTA ne peut être qu'indicative en raison des divergences de puissance des nœuds qui les composent (*full nodes* avec ASIC/GPU vs. Raspberry Pi 3B+). Malgré cela, il est intéressant d'évaluer la latence du consensus de Wallance suivant les tailles réseau des différentes blockchains. À partir des résultats précédents, il est possible d'en déduire un modèle d'évolution de la latence, dont la formule est décrite

par l'équation (3-1). Le Tableau 3-8 résume le nombre de nœuds présents pour chaque blockchain et donne les différents temps de consensus.

$$y = 0{,}0189 \times Nb_{Noeuds} + 0{,}1854 \quad \textbf{(3-1)}$$

Tableau 3-8 - Comparaison de la latence du consensus de Wallance avec l'état de l'art

Protocoles	Nombre de nœuds[1]	Temps de confirmation d'un bloc	Temps de confirmation de Wallance
Bitcoin	10 400 [220]	60 min [199]	~ 3 min
Ethereum	8 542 [221]	6 min [199]	< 3 min
Nano	90 [205]	< 2 s [272]	1,9 s
IOTA	291 [273]	10 s [274][2]	5,7 s

1 : Seulement les nœuds publics pouvant être détectés

2 : Géré par le Coordinateur, pouvant valider plusieurs transactions à la fois

Il est important de préciser que ces comparaisons avec les blockchains de l'état de l'art sont réalisées suivant les configurations actuelles de ces dernières (niveau de difficulté, capacités des nœuds bien supérieures à celles des cartes Raspberry Pi 3B+). Aussi, les données temporelles correspondent au temps de confirmation d'un bloc, c'est-à-dire lorsqu'il est définitivement inscrit dans la blockchain. En raison de leur approche probabiliste, Bitcoin et Ethereum requièrent l'attente de plusieurs blocs pour s'assurer de la validité des précédents, ce qui leur vaut une forte latence de confirmation. À l'inverse, Wallance se base sur un système de votes, dont la fin de la procédure découle sur une acceptation définitive. Grâce à cela, le protocole proposé atteint une latence environ 18 fois plus faible que celle de Bitcoin et 2 fois plus petite que celle d'Ethereum. Le cas d'IOTA est plus particulier, en raison de l'utilisation d'un nœud spécial, le coordinateur, dont le rôle est de régulièrement valider les transactions passées (toutes les 10 s en moyenne). Associé à la structure en DAG interconnectant une multitude de transactions, le coordinateur peut en valider plusieurs à la fois. Wallance atteint le consensus presque 2 fois plus rapidement, à la différence que chaque vote ne peut valider qu'une demande d'accès à la fois. En revanche la procédure s'effectue de manière totalement décentralisée contrairement à IOTA. Enfin, Nano se base lui aussi sur un système de votes, dont la finalité est similaire à celle de Wallance. Ainsi, ces résultats soulignent la pertinence des performances du protocole proposé, et laissent entrevoir une utilisation à plus grande échelle.

f) Espace de stockage

Un dernier élément primordial qu'il reste à évaluer concerne l'espace de stockage, une des principales contraintes de l'adaptation de la blockchain à l'IdO. Avant tout, il est important de souligner que le niveau de sécurité d'un protocole dépend évidemment de la sensibilité des données

qu'il manipule, et doit être adapté en conséquence. Par exemple, les blockchains Bitcoin, Ethereum, Nano et IOTA ont pour application première la cryptomonnaie, c'est-à-dire d'effectuer des transferts d'argent/de valeur. Il est trivial d'estimer qu'un tel système se doit d'être à minima autant sécurisé qu'un système bancaire classique. Une des techniques employées par ces blockchains est le stockage intégral de toutes les transactions. Comme détaillé dans le Tableau 3-3, Bitcoin et Ethereum requièrent respectivement 343 Go et 409 Go, contre environ 29 Go pour Nano et 1,6 Go pour IOTA. Alors que chaque protocole fixe théoriquement une taille maximale de bloc/transaction (*e.g.* 1Mo pour Bitcoin [275] et entre 20 Ko et 30 Ko pour Ethereum [276]), il est intéressant d'évaluer expérimentalement les tailles moyennes, détaillées par le Tableau 3-9. Avec une taille moyenne de 532 Ko, Bitcoin possède les plus gros blocs en termes d'octets, suivi par Ethereum avec 38 Ko. IOTA, qui utilise uniquement des transactions, poursuit cette réduction jusqu'à 1,6 Ko. Enfin, Nano se montre être la blockchain la plus compacte des quatre, avec 520 octets par bloc.

Tableau 3-9 - Comparaison des tailles de blocs/transactions des blockchains de l'état de l'art

Blockchains	Taille de stockage	Nombre de blocs/transactions	Taille moyenne d'un bloc/transaction (octets)
Bitcoin	343 Go [192]	644 552 blocs [192]	532 Ko
Ethereum	409 Go [192]	10 697 176 blocs [192]	38 Ko
Nano	29,61 Go [193]	56 953 665 blocs [193]	520
IOTA	1,6 Go [194]	1 000 000 TXs [194]	1 600

Bien que la sécurité soit de mise, le contexte de l'IdO visé par Wallance peut se permettre un degré inférieur à celui de ses homologues, afin de se soustraire aux forts besoins de ressources, notamment l'utilisation de l'espace de stockage, ici étudiée. À partir des résultats précédents, la Figure 3-35 compare l'espace de stockage nécessaire pour chaque protocole, en fonction du nombre de blocs (Bitcoin, Ethereum et Nano), de transactions (IOTA) ou de portefeuilles (Wallance).

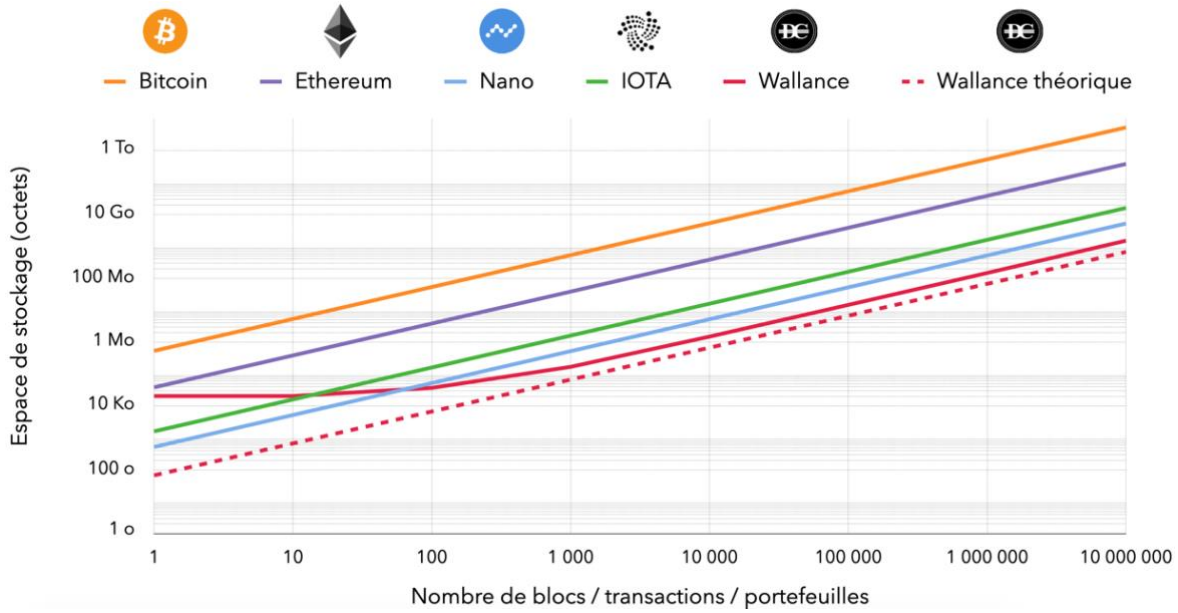


Figure 3-35 - Comparaison de l'espace de stockage de Wallance avec les blockchains de l'état de l'art

Dans sa forme la plus simple, la taille d'un portefeuille est de 67 octets : 33 pour l'identifiant du nœud, 2 pour son solde de DCoin et enfin 32 pour l'*état* (*cf.* Tableau 3-5). Toutefois, l'implémentation actuelle utilise une base de données SQLite [259] qui demande la création de « tables » permettant de classer les différentes informations (portefeuilles, transactions de demande d'accès, votes). Ces dernières requièrent environ 20 Ko alors qu'aucune donnée n'y est encore inscrite. Cela, associé au faible taux de compression de SQLite sur de petites bases de données, explique une utilisation de stockage plus importante de Wallance par rapport à Nano et IOTA pour un faible nombre de blocs/portefeuilles (inférieur à 100). Un autre point à éclaircir concerne le formatage des données. Pour des raisons de simplicité d'interopérabilité entre Wallance et SQLite (*e.g.* parcage, ajout, mise à jour, etc.), chaque donnée fait l'objet d'une conversion en chaines de caractère hexadécimales, avant d'être inscrite dans la base de données. La nouvelle taille s'élève alors à 134 octets. Toutefois, l'évaluation empirique illustrée par la Figure 3-35, révèle une taille réelle moyenne de 148 octets par portefeuille, soulignant une réduction drastique de l'espace de stockage requit par Wallance, d'un facteur supérieur à 3 000 et 258 par rapport à Bitcoin et Ethereum respectivement. De même, le protocole proposé permet une diminution du stockage par 10 en comparaison avec IOTA et par 3,5 par rapport à Nano. Mais plus important encore, la force de Wallance réside dans sa structure en portefeuilles-tressés, lui permettant de ne pas dépendre du nombre de transactions comme les blockchains classiques telles que Bitcoin, Ethereum ou IOTA, mais uniquement sur le nombre de portefeuilles, c'est-à-dire de participants sur le réseau. À noter que Nano adopte lui aussi cette structure, et propose plusieurs politiques de stockage sur ses nœuds (conservation de la totalité des blocs, d'une partie, ou comme Wallance, uniquement les dernières

mises à jour de chaque compte). Toutefois, comme le montre l'explorateur officiel de Nano [193], l'intégralité de la blockchain est conservée par les *full nodes*, garants de la sécurité du système. Cela signifie que pour l'heure, Nano n'exploite pas cette propriété, contrairement à Wallance. Par ailleurs, il est possible d'estimer la taille de Wallance suivant le nombre d'adresses (ou comptes) actuellement présents sur les différentes blockchains étudiées. Le Tableau 3-10 présente les résultats.

Tableau 3-10 - Taille de Wallance en fonction du nombre de comptes des blockchains de l'état de l'art

Blockchains	Nombre de comptes	Équivalence de taille de Wallance
Bitcoin	728 490 000[1] [277] / 31 826 511[2] [278]	107,81 Go / 4,7 Go
Ethereum	116 297 887 [279]	17,21 Go
Nano	461 170 [280]	68,25 Mo
IOTA	350 276 [281]	51,84 Mo

1 : nombre d'adresses totales, y compris celles ayant un solde nul

2 : nombre d'adresses ayant un solde non nul

Avec quelques dizaines de gigaoctets, Wallance est capable de stocker plusieurs centaines de millions de comptes. Bien que ces chiffres puissent être éloignés des prévisions stipulant 38,6 milliards d'objets connectés, équivalent à 5,7 To de stockage sur Wallance, ce dernier est avant tout conçu pour fonctionner au niveau des passerelles réseau (*gateways*). En supposant que chacune d'entre elles soit en charge d'interconnecter au réseau global une trentaine d'objets en moyenne, cela représenterait environ 1,28 milliard de passerelles, soit autant de portefeuilles Wallance, dont la taille totale s'élève à 190 Go. Une carte SD de Raspberry serait en capacité de stocker l'intégralité de la base de données. Cela équivaut à 40% et 53% de moins que la taille actuelle de Bitcoin et Ethereum respectivement, et il faudrait plus de 2 To à IOTA pour ne stocker qu'une seule transaction de chaque passerelle. Quant à Nano, pas moins de 669 Go seraient nécessaires pour stocker l'intégralité des comptes. Ce scénario souligne l'intérêt de l'approche par compte de Wallance, et témoigne de la capacité de scalabilité de ce dernier en termes de ressources de stockage.

Une autre particularité de Wallance concerne sa grande flexibilité en termes de stockage. Alors que la configuration classique du protocole stipule que chaque nœud possède une copie complète de tous les portefeuilles, la structure de portefeuilles-tressés implique une indépendance de ces derniers les uns par rapport aux autres. Cette propriété ouvre notamment la voie au *sharding* [212], [213], [214] qui consiste à découper une base de données en plusieurs parties, en créant plusieurs sous-réseaux Wallance. De plus, cela permet d'élargir les possibilités du protocole, en facilitant l'intégration d'entités ayant de faibles capacités de stockage leur permettant de prendre part à l'écosystème. Afin d'illustrer cette capacité, la Figure 3-36 détermine le nombre de portefeuilles qu'il est possible de stocker en fonction de l'espace de stockage disponible et/ou attribué.

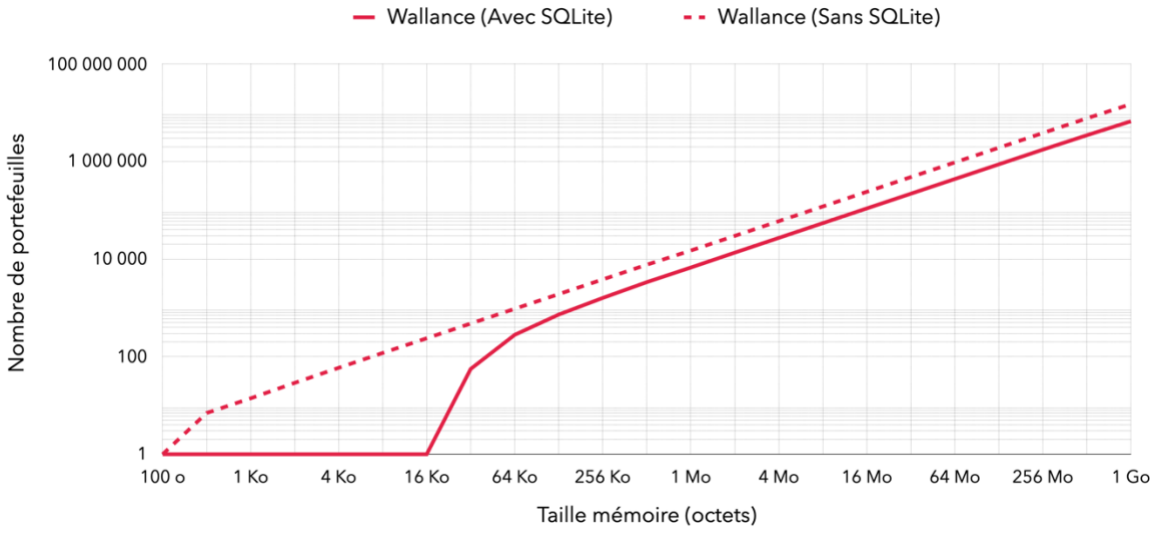


Figure 3-36 - Nombre de portefeuilles pouvant être stockés en fonction de la taille mémoire disponible

Deux évaluations sont présentées : l'une avec utilisation de SQLite [259], privilégiée pour la gestion d'un grand nombre de portefeuilles (opérations sur la base de données). La seconde correspond à une implémentation sans SQLite, plus adaptée pour ne stocker qu'une faible quantité de comptes, en raison du surplus de stockage que requiert la base de données. D'après les résultats, et en se basant sur de vraies architectures de systèmes embarqués tels que les STM32 Nucleo [282] (mémoire Flash de 16 Ko à 256 Ko), le JN5189 de NXP [208] (640 Ko de Flash) ou encore les séries ESP32 d'Espressif [283] (4 Mo à 16 Mo), ces derniers seraient en mesure de stocker d'une dizaine à plusieurs milliers (voire dizaines de milliers) de portefeuilles, en prenant en compte l'espace mémoire utilisé pour embarquer le code binaire. Toutefois, au-delà d'une certaine quantité, qui pourrait faire l'objet d'une future évaluation, la version SQLite se prête mieux au protocole, pour des questions d'efficacité de gestion des données. C'est notamment le cas pour les entités disposant de plus de ressources comme les passerelles réseau (ici les cartes Raspberry [189]), pouvant facilement gérer plusieurs millions de portefeuilles avec seulement quelques centaines de mégaoctets. Ainsi, cette analyse met en avant le fort potentiel de scalabilité, mais aussi de flexibilité de Wallance en termes d'espace de stockage.

3.5.7. Synthèse

Après avoir mis en avant les différents verrous d'une adaptation de la blockchain pour l'IdO, cette section 3.5 détaille le protocole Wallance, mis en place durant cette thèse, se positionnant comme une alternative légère à la blockchain, spécialement conçue pour le contexte de l'IdO. L'objectif est de promouvoir le partage et l'utilisation de données/ressources et services, tout en instaurant la confiance dans le réseau, en sécurisant les interactions de manière décentralisée, et en limitant le

besoin de ressources (puissance de calcul, stockage et énergie). Pour mener à bien sa mission, Wallance se focalise sur les interactions entre les passerelles réseau, cœur d'un écosystème IdO, en monétisant la quantité de partage (données/ressources/services) au travers du DCoin (Data Coin).

Pour contrer le problème lié à l'espace de stockage imposé par la blockchain, Wallance adopte une structure de portefeuilles-tressés. Chacun d'eux représente le compte d'une entité contenant son identifiant, sa quantité de DCoin, ainsi qu'un *état* assurant l'intégrité des informations. Étant tous indépendants, ils peuvent être mis à jour de manière totalement asynchrone, sans aucun impact sur les autres, réduisant considérablement la présence de *fork* et par là même, la latence de confirmation du système. De plus, cette structure permet de ne conserver uniquement la dernière version de chaque compte, résultant sur un espace de stockage qui dépend uniquement du nombre de participants, contrairement aux blockchains classiques. L'accès aux données/ressources et services s'effectue au moyen d'une demande, que la communauté doit valider au moyen d'une procédure de consensus reposant sur un système de votes. Si la majorité donne son accord, l'accès est autorisé et le demandeur se voit débiter les DCoins en conséquence. Une récompense monétaire est attribuée à tous les nœuds ayant correctement participé au consensus, incitant ainsi la sécurisation du système.

Afin d'évaluer expérimentalement le protocole Wallance, ce dernier est implémenté sur la plateforme de prototypage (*cf.* 3.3.2) constituée de cartes Raspberry Pi 3B+ [189], pouvant être assimilées aux passerelles réseau, en termes de ressources disponibles. Plusieurs implémentations sont proposées (CWallance, pyWallance et WallanceDDS), ainsi qu'une interface graphique permettant de visualiser et d'interagir en temps réel avec l'écosystème. L'évaluation souligne qu'une difficulté de 1 du LWPoW, mécanisme anti-spam de Wallance, se montre intéressante pour limiter ces derniers sans trop impacter la latence (un maximum de 4 ms pour générer une transaction) du protocole. De plus, l'étude énergétique montre qu'un tour complet de Wallance (initialisation, partage de données capteur, puis émission et acceptation d'une demande d'accès) ne dépasse par les 7 J par nœud, et se montre 3 fois moins énergivore qu'une passerelle LoRa du commerce [271], sur une période d'une heure d'utilisation (soit environ 1 300 traitements de demande d'accès sur le protocole Wallance). L'évaluation porte également sur le temps de finalité du consensus, c'est-à-dire le temps que met le protocole pour valider unanimement et définitivement une demande d'accès à une donnée/ressource ou service. Avec le maximum de cartes Raspberry connectées sur la plateforme (48), Wallance atteint le consensus de l'ordre de la seconde, ce qui est équivalent à Nano [119], 18 fois moins que Bitcoin [114] et 2 fois moins qu'Ethereum [117] et IOTA [118]. Enfin, le dernier point abordé concerne l'espace de stockage requis par le protocole proposé. Chaque portefeuille Wallance ne requiert que 67 octets dans son implémentation classique, et s'élève à 148 octets avec la base de données SQLite [259], propice à la gestion d'un grand nombre de portefeuilles. Cela offre à Wallance une réduction drastique de l'espace de stockage d'un facteur 3 000 par rapport à Bitcoin, et 3,5 par rapport à Nano,

plus petite blockchain de l'état de l'art. Avec quelques dizaines de gigaoctets, Wallance est capable de gérer plusieurs centaines de millions de comptes. Une autre particularité du protocole proposé concerne sa grande flexibilité en termes de stockage. Grâce à l'indépendance de ses portefeuilles, Wallance supporte intrinsèquement le *sharding*, mais permet également d'intégrer les entités ayant de faibles capacités de stockage, leur permettant de prendre part à l'écosystème. L'évaluation met en avant les possibilités de vraies architectures d'objets connectés (STM32 Nucleo [282], JN5189 [208], séries ESP32 [283]) pouvant stocker jusqu'à plusieurs milliers de portefeuilles. Ainsi, les résultats expérimentaux de cette section soulignent la pertinence des performances de Wallance en termes de complexité de calcul, de latence, d'énergie, mais aussi d'espace de stockage, laissant entrevoir une utilisation à plus grande échelle.

3.6. TrustLib, un protocole de confiance léger pour les objets

3.6.1. Motivations

Un environnement de l'IdO est composé d'une pléiade d'entités de différentes natures, allant de simples nœuds capteurs (*e.g.* température, humidité), à des actionneurs (*e.g.* vanne de contrôle, luminaire), en passant par des systèmes plus complexes comme les montres connectées, jusqu'aux passerelles réseaux et enfin les serveurs Cloud. Grâce aux ressources de calcul et de mémoire qu'elles embarquent, ainsi que leur capacité à communiquer avec leurs homologues de manière directe ou indirecte, chaque entité est en mesure de s'échanger des informations et de les traiter. Il est alors question d'objets intelligents, ayant la capacité de s'adapter et de prendre des décisions en autonomie. Il est important de rappeler que dans ce contexte, les serveurs/Cloud sont des éléments à part entière du réseau, offrant des services plus complexes (*e.g.* analyses par intelligence artificielle, stockage, etc.) mais ne constituent en rien une autorité centrale. Face à cette décentralisation, le principal défi est d'instaurer la confiance entre les objets, sans tierce partie.

Comme évoqué en début de ce chapitre, les systèmes blockchains ont été fortement étudiés par la communauté. Cet engouement est principalement nourri par les propriétés de données infalsifiables, l'absence d'un point de défaillance unique (*single point of failure*), la capacité à fonctionner sans tiers de confiance, ainsi que l'aspect d'auto-vérification (*i.e.* par les participants) et la force de dissémination des informations à travers le réseau qu'offrent ces systèmes. Grâce à cela, la blockchain permet la mise en place d'un écosystème distribué et sécurisé, où chaque opération est transparente. Toutefois, malgré tous ses avantages, elle ne permet pas d'assurer la viabilité des données circulant sur le réseau. Par exemple, elle est dans l'incapacité de déterminer le bon ou le mauvais comportement d'un appareil, mais seulement stocker de manière sûre des informations à son sujet, pour les analyser

grâce à un modèle de confiance. Cela souligne l'intérêt de l'association des systèmes de confiance et des blockchains (*Blockchain-based Trust Management*).

La majorité des travaux [103], [113], [284]–[286] utilise la blockchain pour stocker les niveaux de confiance de chaque objet. Afin d'améliorer leurs modèles, les études [103] et [113] utilisent une association de la blockchain avec une table de hachage distribuée (*distribued hash table*, DHT). Cette dernière étant plus rapide pour les opérations de recherche d'information, les auteurs font le choix de stocker les niveaux de confiance sur la blockchain, pour sa propriété de non-falsification, et utilisent la DHT pour les retrouver [113]. À l'inverse, l'étude [103] se sert de cette dernière uniquement pour maintenir une liste des services fournis par chaque objet du réseau. La blockchain peut également être utilisée pour stocker les informations relatives au calcul de la confiance, telles que les propriétés de l'objet, ses capacités ainsi que sa réputation aux vues des autres [287]. Enfin, l'étude [288] propose d'inscrire dans la blockchain uniquement les informations d'authentification et les niveaux de confiance de chaque entité du réseau.

Malgré de nombreuses contributions intéressantes, les travaux examinés au cours de cette thèse sont basés sur des blockchains existantes ou privées, qui ne sont pas adaptées au contexte de l'IdO. En effet, ces dernières sont aujourd'hui largement employées dans les applications de cryptomonnaie comme Bitcoin [114] et Ethereum [117], mais en raison de leur forte utilisation de ressources (*i.e.* puissance de calcul, espace de stockage et énergie), de nombreuses avancées sont encore requises pour l'adaptation dans un environnement de l'IdO [116], [122], [123], [191]. D'autre part, parmi les contributions proposées dans la littérature, peu d'entre elles prévoient de récompenser ou de punir les entités, ce qui est primordial pour inciter à un bon comportement. Enfin, les approches sont axées au niveau des passerelles réseaux, là où l'efficacité de la blockchain prend tout son sens. Par conséquent, il persiste toujours un manque de solution permettant d'assurer la confiance directement entre les objets connectés, mais aussi avec leur(s) passerelle(s). L'objectif de cette section vise à proposer un second protocole nommé TrustLib, en complément de Wallance, et plus léger que ce dernier, afin d'instaurer la confiance dans l'écosystème au plus près des objets (objets à objets et objets à passerelles).

3.6.2. Portée et principe du protocole TrustLib

Pour correctement appréhender l'approche du protocole TrustLib, il est important de comprendre sa portée et son association avec le protocole précédent Wallance (*cf.* section 3.5). Pour cela, la Figure 3-37 illustre une vue d'ensemble d'un réseau de l'IdO, où sont référencés les champs d'action des deux protocoles proposés.

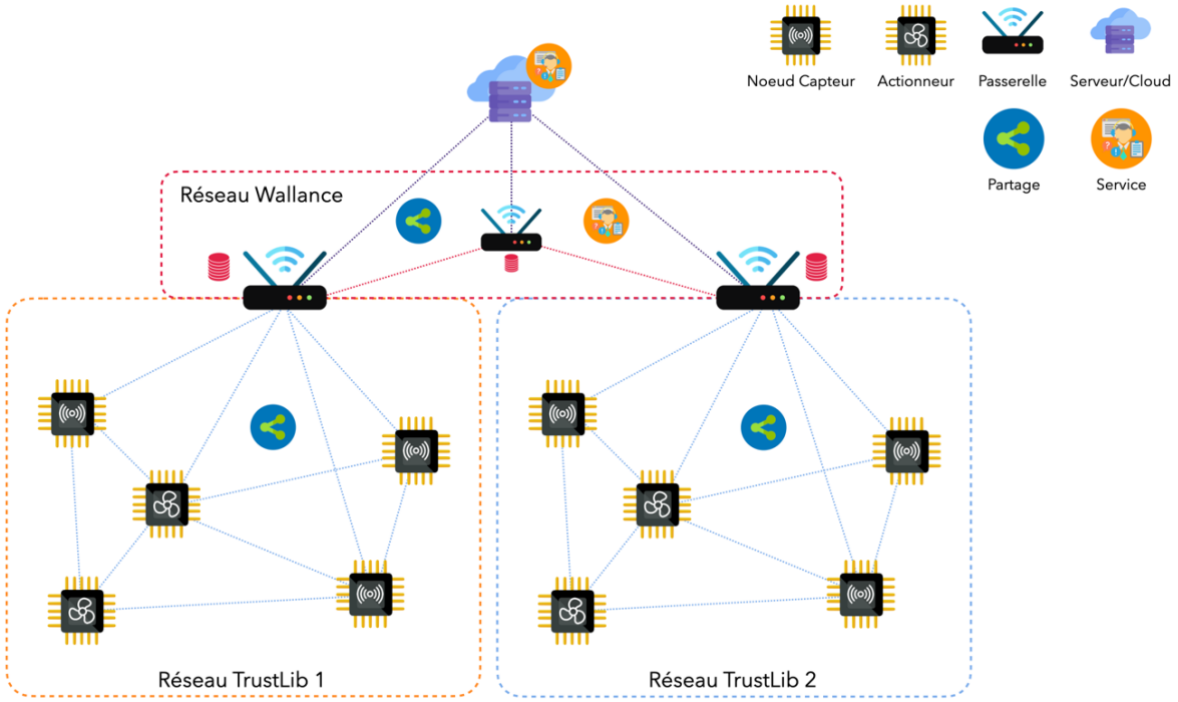


Figure 3-37 - Réseau IdO avec les protocoles Wallance & TrustLib

Bien que le protocole Wallance jouisse d'une large configurabilité, il est particulièrement adapté au niveau des passerelles réseau, dans le but de sécuriser leurs interactions. De plus, un environnement de l'IdO favorise les communications directes entre les appareils, sans impliquer les passerelles. Ainsi, pour compléter le champ d'action de Wallance et pallier au manque de solution sécurisant les interactions objets à objets/objets à passerelles, cette section propose le protocole, TrustLib, pensé comme un système de confiance pouvant être embarqué dans les objets. Comme le souligne l'étude [289], il existe de nombreuses contraintes pour mettre en place un tel protocole dans le contexte de l'IdO. En effet, la grande quantité et diversité d'objets connectés, associées à leur perpétuelle mise en en veille, rend l'environnement hautement hétérogène et dynamique. TrustLib doit ainsi être en mesure de supporter ces aléas, en tenant compte de l'énergie, de la capacité de stockage et de la puissance de calcul limitées des objets.

En outre, l'inclusion d'un modèle d'incitation, garant de la cohérence d'un écosystème, est un élément primordial dans la conception du protocole. L'objectif est d'introduire une motivation, de sorte que chaque entité adopte un comportement coopératif et respecte les règles communes, afin d'assurer la pérennité du système. Cette approche constitue un élément clé dans les blockchains telles que Bitcoin [114], Ethereum [117], et le protocole Wallance. Les participants sont incités à sécuriser les interactions en prenant part au consensus en contrepartie d'une rémunération. Concernant TrustLib, l'idée est qu'il intègre intrinsèquement cette stratégie, au travers de l'amélioration de

l'efficacité et de la fiabilité qu'il vise à mettre en place, aussi bien du point de vue des entités que systémique (*i.e.* qualité de service/d'expérience). Par ailleurs, le niveau de confiance attribué aux entités, image de leur fiabilité, doit réguler l'activation/désactivation d'opérations et de services. Par exemple, si la source d'information n'est pas fiable, l'opération/le service n'est pas effectué. La pérennité de l'écosystème dépend alors de ses propres membres, ayant ainsi intérêt à fonctionner de pair.

Alors que la pertinence d'un protocole de confiance s'exprime au travers de son efficacité et de sa pertinence, sa facilité de mise en œuvre aussi bien dans les nouveaux systèmes que dans les existants n'est pas à ignorer. En effet, l'attrait des objets connectés est aujourd'hui acté et laisse présager de larges déploiements. À tel point que de grands groupes comme Amazon et Microsoft ont mis en place plusieurs plateformes, facilitant le développement et le déploiement d'objets connectés [290]. Inspiré par cette approche, le protocole TrustLib est pensé pour être facilement mis en place. Il se présente sous la forme d'une bibliothèque logicielle (d'où son nom), libre d'accès (*open source*), à inclure dans le code source de l'objet, à l'aide d'une interface de programmation (*Application Programming Interface*, API). Grâce à cela, le protocole TrustLib peut par exemple, aisément être ajouté à Wallance en guise de module complémentaire (*i.e. plug-in*) afin d'améliorer la confiance des interactions entre passerelles.

3.6.3. Hypothèse de départ

L'incroyable pluralité d'applications que laisse entrevoir l'IdO rend la mise en place d'un protocole universel particulièrement délicate, comme en témoigne la multitude de réseaux actuellement disponibles (LoRa, SigFox, ZigBee, etc.). Comme une première tentative vers l'universalité, les travaux réalisés dans cette section s'efforcent d'être adaptables sur un large spectre d'applications. Aussi, il convient de préciser que l'utilisation d'un protocole de confiance prend tout son sens, lorsqu'un certain degré de contrôle et de sécurité des interactions est exigé. Un réseau, de quelque nature qu'il soit, n'est pas infaillible et peut entrainer des retards, voire des pertes d'informations, ce qui peut s'avérer problématique suivant l'application ciblée. Par exemple, dans le cadre d'un simple suivi de température des salles d'un bâtiment, dont la publication des valeurs s'effectue toutes les heures, la perte d'une de ces dernières n'impacte que faiblement le système. Celui-ci reprendra automatiquement son fonctionnement normal, dès la prochaine publication. À l'inverse, un détecteur d'intrusion dans une salle doit transmettre un signal d'alarme à son contrôleur, qui à son tour, déclenche des voyants lumineux, une sirène, ainsi qu'une notification au poste de sécurité. Dans un tel scénario, la viabilité du système repose sur le fonctionnement conjoint de chaque entité, et sur leur capacité à s'assurer de cette coordination :

- Détecteur d'intrusion : est-ce que le contrôleur a bien reçu et traité le signal d'alarme ? Est-ce que le signal doit être réémis ?
- Contrôleur : est-ce que le détecteur d'intrusion est fiable ? Est-ce que les éléments d'alerte (*i.e.* voyant lumineux, sirène, système de notification) ont reçu l'ordre d'activation ? Sont-ils fonctionnels ?
- Voyant lumineux, sirène, système de notification : est-ce que le contrôleur est fiable ? Est-ce que ses ordres sont légitimes ?

Ce scénario souligne la non-universalité et la non-trivialité du concept de la confiance, telles qu'évoquées en introduction de ce chapitre. L'objectif d'un protocole dédié se doit de répondre aux questions ci-dessus. Cela implique la prise en compte du rôle de chaque entité dans cet écosystème. De manière générale, trois catégories sont identifiables :

Nœud Capteur (*e.g.* détecteur d'intrusion) : son rôle est de numériser son environnement physique, au moyen d'un ou plusieurs capteurs. Les données sont ensuite traitées par le nœud afin d'en extraire les informations, avant de transmettre ces dernières sur le réseau.

Actionneur (*e.g.* voyant lumineux, sirène) : son rôle consiste à agir sur son environnement, en fonction des données provenant des capteurs et/ou des commandes des contrôleurs.

Passerelle (*e.g.* contrôleur) : son objectif premier est de transmettre les informations (*e.g.* données capteurs, ordres de commande, etc.) vers/depuis d'autres entités du réseau local et global (*e.g.* LAN, WAN). Pour cela, la passerelle est capable d'interconnecter le monde radio avec le protocole standard TCP/IP. Sa position sur le réseau lui confère un fort potentiel d'agrégation de données, permettant la mise en place de services, comme les exemples de la station météo et de l'application d'aide à la navigation, évoqués en introduction.

La suite de ce travail s'applique dans le cadre de scénarios et/ou d'interactions justifiant un besoin de confiance entre les entités, comme dans l'exemple susmentionné.

3.6.4. État de l'art sur les métriques de confiance dans un contexte d'IdO

Afin de démystifier le concept de la confiance, cette section recense les propriétés les plus pertinentes utilisées dans la littérature, dans le contexte de l'IdO. Le Tableau 3-11 extrait ces dernières des études [103], [104], [106]–[111], [291], [292] et les classe en trois catégories, en fonction de leur champ d'action : objet, service et réputation.

Tableau 3-11 - État de l'art sur les propriétés de confiance

Catégories	Propriétés	Descriptions
Objet	Similarités	Évaluer les similarités entre objets : - Propriétés communes (relations, intérêts, localisation, types, ...) - Objectifs communs (adaptation continue du réseau)
	Participation	Évaluer la participation d'un objet sur le réseau : - Partage de données/ressources/services - Rapport nombre de tâches effectuées sur nombre de tâches totales - Niveau d'énergie restant (basée sur l'activité de l'objet)
	Intégrité	Évaluer l'utilisation des niveaux de confiances corrects d'un objet
	Communication	Évaluer la communication entre objets : - Qualité du signal (*RSSI*) - Stabilité (taux de transmission, latence) - Utilisation (taille des paquets, surcharge, bande passante) - Sécurité (confidentialité, authenticité et intégrité)
Service	Test	Évaluer les fonctionnalités d'un service : - Fonctionnel (Oui/Non) - Performances (temps de réponse, énergie requise)
	Intérêt	Évaluer l'activité d'un service (nombre d'utilisation)
	Notation	Évaluer la satisfaction d'un objet par rapport à un service : - Score de satisfaction - Disponibilité (capacité à fournir le service au cours du temps) - Taux de réussite (*cf.* Communication)
Réputation	Recommandation	Niveau de confiance accordé à un objet/service par un autre objet, ayant un lien direct

En ce qui concerne la catégorie « Objets », l'évaluation de la confiance dépend en premier lieu des similarités entre les entités. Si des objets partagent les mêmes caractéristiques et les mêmes objectifs, il est plus facile d'établir un lien de confiance entre eux. La participation et l'intégrité mettent en évidence le comportement d'un objet sur le réseau. Enfin, sa capacité à assurer des communications stables peut être utilisée pour souligner sa fiabilité. Les services prennent eux aussi place dans l'estimation de la confiance d'un écosystème. Principalement inspirées par l'étude [113], trois propriétés permettent d'évaluer un service. La première, la plus statique, concerne le test. Pour cela, chaque service doit contenir les informations nécessaires à la vérification de ses fonctionnalités et des performances attendues. Quant aux propriétés « Intérêt » et « Notation », elles représentent respectivement l'évaluation de l'activité et de la satisfaction d'un service, par les objets ou d'autres services. Enfin, la réputation correspond à l'agrégation des catégories « Objet » et/ou « Service », en

une valeur unique de confiance, que les objets diffusent sur le réseau pour ceux qui n'ont pas de connexion directe avec l'entité à évaluer.

3.6.5. Confiance directe et/ou indirecte

Pour rappel, la confiance directe fait référence à l'évaluation d'une entité, en fonction de ses propres expériences. À l'opposé, la confiance indirecte se base sur les recommandations recueillies sur le réseau. D'après l'étude [293], il est impossible d'avoir une vision de la confiance complète et fiable avec seulement des interactions directes, surtout lorsque celles-ci sont peu nombreuses. De plus [110] met en avant que la confiance indirecte est un moyen important de réduire le risque d'attaques malveillantes. Néanmoins, il est important d'analyser l'impact du mécanisme de confiance indirecte sur les performances et la sécurité. À partir de la littérature ([103], [105], [110], [289] et [294]), le Tableau 3-12 résume les principales menaces qui pèsent sur les modèles de confiance.

Tableau 3-12 - Attaques sur les modèles de confiance

Attaques	Descriptions	Type de confiance
Autopromotion	L'entité malveillante se promeut elle-même comme étant honnête, mais fournit des données/services non fonctionnels ou erronés	Indirecte
Blanchiment	L'entité malveillante se connecte/déconnecte pour effacer son niveau de confiance lorsque celui-ci est trop faible	Directe / Indirecte
Discrimination	Tentative de tromper une nouvelle entité qui n'a pas encore suffisamment évalué le niveau de confiance des autres	Indirecte
Diffamation	Tentative de dégrader le niveau de confiance d'une entité en répandant de fausses et mauvaises réputations	Indirecte
Remplissage des urnes / Louange	Tentative d'augmenter le niveau de confiance d'une entité en répandant de fausses mais bonnes réputations	Indirecte
Opportuniste	L'entité malveillante fournit des données/services fiables uniquement lorsque son niveau de confiance vis-à-vis des autres est trop bas afin d'augmenter ce dernier	Directe / Indirecte

Il est intéressant de noter que la majorité des attaques sont orientées sur la confiance indirecte, plutôt que directe. En effet, alors que cette dernière ne dépend que de la propre vision d'un objet, la confiance indirecte exige une évaluation de la part des autres. Cela conduit à de potentielles manipulations et augmente le risque de mauvaises évaluations. Pour surmonter ce problème, des mécanismes supplémentaires doivent être utilisés pour garantir la validité, mais aussi l'authenticité et l'intégrité des recommandations. En outre, l'entité qui les reçoit doit également tenir compte du

niveau de confiance actuel du recommandeur. Enfin, un modèle basé sur la confiance indirecte nécessite une communication intensive entre les objets afin de transmettre les valeurs de recommandation. Tout cela augmente les couts de calcul et d'énergie, conduisant à un protocole moins évolutif, moins performant, voire moins sécurisé. Par conséquent, le compromis entre une vision fiable de la confiance et la sécurité n'est pas nécessairement significatif.

3.6.6. Sélection des métriques et calcul du niveau de confiance

Au cours de leurs interactions, le niveau de confiance qu'une entité accorde à une autre évolue, donnant lieu à des comportements différents et des activations/désactivations de services. La section 3.6.3 souligne l'intérêt de prendre en considération le rôle de chaque entité dans son écosystème, afin de mettre en place un protocole de confiance pertinent. En raison des faibles capacités des appareils, le protocole doit être aussi simple que possible. C'est pourquoi les métriques quantifiables sont privilégiées. En accord avec la section 3.6.5, TrustLib s'oriente uniquement vers une évaluation directe de la confiance, afin d'éviter la complexité induite par l'indirecte. Néanmoins, cette dernière peut être effectuée au travers de l'association avec le protocole Wallance, par exemple entre plusieurs réseaux TrustLib (*cf.* Figure 3-37). Cela permet d'alléger le protocole au niveau des objets, tout en assurant la sécurité inter-passerelles.

a) Nœud capteur

Le rôle d'un nœud capteur est principalement de produire et transmettre de l'information. Sa vision de la confiance envers son environnement sera alors étroitement liée à l'efficacité de ses homologues à traiter les informations qu'il leur transmet. Pour rappel, l'hypothèse de départ stipule que le protocole TrustLib se place dans un contexte ayant un besoin de contrôle et de confiance entre les entités, tel que le scénario d'intrusion (*e.g.* est-ce que le contrôleur a bien reçu et traité le signal d'alarme ? *cf.* section 3.6.3). Pour répondre à cette problématique, une technique permettant de s'assurer de la bonne transmission d'une information critique sur le réseau, est l'envoi par le destinataire, d'un signal de confirmation (*acknowledege*). Grâce à ce dernier, le nœud capteur est en mesure d'évaluer plusieurs métriques de communication, tels que le taux de réussite, la latence, et la qualité du signal. Il peut ainsi privilégier l'émission de ses informations aux entités avec qui il possède une meilleure stabilité réseau, alias le niveau de confiance.

En guise de preuve de concept, cette thèse propose plusieurs formules pour évaluer la confiance d'une entité vue par un nœud capteur, à partir des métriques suivantes : la réussite de transmission (R_T), la latence (NL) et la qualité du signal de réception (NS_Q) toutes deux normalisées. Puisque la source d'énergie d'un nœud capteur est souvent très limitée, son propre niveau de batterie (Bat),

normalisé entre 0,1 et 1, est également pris en compte. Tous ces éléments sont ensuite agrégés, en guise d'évaluation de la confiance. Les formules (3-2) à (3-5) détaillent les calculs.

$$R_T = \begin{cases} -1 \text{ en cas d'échec} \\ 1 \text{ en cas de succès} \end{cases} \tag{3-2}$$

$$NL = \frac{L_i - L_{MIN}}{L_{MAX} - L_{MIN}} \qquad NL \in [0\,;1] \tag{3-3}$$

$$NS_Q = \frac{S_i - S_{QMIN}}{S_{QMAX} - S_{QMIN}} \qquad NS_Q \in [0\,;1] \tag{3-4}$$

$$AG_{Capteur} = R_T \times \left(\frac{1}{\Big((1 + NL) + (2 - NS_Q)\Big)^{R_T}} \right) \times \frac{1}{Bat} \qquad Bat \in [0{,}1\,;1] \tag{3-5}$$

La réussite d'une transmission (R_T) dépend entièrement du signal de confirmation envoyé par le destinataire. Si ce dernier n'accuse pas la réception, le message émis par le nœud est considéré comme perdu. De même, le délai (L_i) entre l'émission du message, et la réception du signal de confirmation est pris en considération. Cette latence est ensuite normalisée (NL) entre 0 (meilleur cas) et 1 (pire cas), en fonction des temps définis par l'application (L_{MIN} et L_{MAX}, pour la latence minimale et maximale respectivement). La force du signal de réception (S_i) est évaluée et normalisée (NS_Q) en fonction de sa valeur maximale S_{QMAX} et minimal S_{QMIN}, par exemple, à partir du RSSI. En cas d'échec de transmission (R_T = -1), la valeur minimale (NS_Q = 0) est prise en compte puisque le signal de confirmation est inexistant.

Le calcul proposé pour l'agrégation $AG_{Capteur}$, a pour objectif de déduire le niveau de confiance qu'un nœud capteur attribue à une entité, à partir des précédentes métriques. C'est la réussite de la transmission (R_T) qui détermine l'augmentation ou la diminution du niveau de confiance. En effet, dès lors que l'opération échoue, la fiabilité de l'entité en question s'amenuise. Les autres métriques définissent le degré d'augmentation ou de diminution de la confiance. La partie centrale de la formule d'agrégation permet d'établir un score de la transmission à partir de latence et de qualité du signal de confirmation. Afin, de simuler une caractéristique sociale de la confiance, à savoir « qu'il est plus difficile de la perdre que de la gagner », ce score est élevé à la puissance R_T (-1 ou 1). Cela signifie qu'en cas d'échec de transmission, la valeur de l'agrégation sera décuplée, pénalisant ainsi fortement la confiance. Le niveau de batterie quant à lui, permet de calibrer la sensibilité de l'agrégation : lorsque le niveau d'énergie du nœud capteur diminue, l'impact de chaque interaction sur la confiance est amplifié.

Afin de justifier ces choix, le scénario suivant est proposé : lorsque le niveau de batterie du nœud capteur descend en dessous d'un certain seuil, il transmet un signal d'alerte à un contrôleur (*e.g.* passerelle). Après quoi, il passera dans un mode d'économie d'énergie en attendant une maintenance (*e.g.* changement de sa batterie). Le nœud doit donc s'assurer que son alerte a bien été reçue et traitée par la passerelle (*e.g.* via un *acknowledge*). Dans le cas contraire, le nœud devra réémettre son alerte, ce qui lui coutera de l'énergie, d'autant plus si la passerelle ne donne toujours pas de confirmation. Plusieurs solutions sont alors possibles. La première consiste à réémettre indéfiniment le message d'alerte, jusqu'à épuisement total de l'énergie du nœud capteur. La seconde méthode est de n'émettre qu'un certain nombre de fois l'alerte, au bout duquel, le nœud passe automatiquement en mode économie d'énergie. Enfin, une troisième option serait de transmettre le signal d'alerte à un autre contrôleur (ou tout autre intermédiaire). Dans ce cas, comment s'assurer que le scénario avec la première passerelle ne va pas se reproduire avec la seconde ? Cette réponse peut être apportée par le protocole TrustLib. L'analyse du niveau de confiance à l'aide de métriques, telles que la qualité de signal et la latence des communications, permet d'identifier l'entité qui est la plus à même de transmettre le message d'alerte. L'efficacité et la qualité de service du système s'en trouvent ainsi améliorées (*i.e.* réduction du coup énergétique, détection et dépannage plus rapide).

b) Actionneur

À l'inverse du nœud capteur, un actionneur exploite les données qu'il reçoit pour sa prise de décision. Cette dernière dépend alors de la fiabilité des informations reçues (*e.g.* précision, intégrité, authenticité, etc.). Il accordera plus de crédibilité à celles provenant d'entités ayant un haut niveau de confiance, améliorant ainsi sa propre fiabilité. Il en est de même pour la qualité du signal reçu, pouvant nécessiter des retransmissions d'informations et de la latence. De plus, puisque l'actionneur agit physiquement sur son environnement, les déclenchements des opérations doivent s'effectuer de manière sûre. Bien que des mécanismes de sûreté puissent être embarqués suivant le type d'actionneur (*e.g.* capteur de fin de course, régulation d'un moteur, disjoncteur, etc.), l'utilisation d'un protocole de confiance constitue un moyen intéressant de filtrage de fausses informations, pour empêcher des actes malveillants et améliorer l'efficacité de l'actionneur. Après cette analyse, cette section propose d'utiliser la qualité du signal NS_Q et la latence NL, définies précédemment, ainsi que la réussite d'une action, R_A. Cette dernière, définie par la formule (3-6), reprend le même principe que la réussite de transmission R_T du nœud capteur, à la différence qu'il s'agit ici de déterminer si l'actionneur a pu effectuer l'opération qui lui a été demandée, sans erreur. Enfin, l'agrégation $AG_{Actionneur}$ est définie par la formule (3-7).

$$R_A = \begin{cases} -1 \text{ en cas d'échec} \\ 1 \text{ en cas de succès} \end{cases} \quad (3\text{-}6)$$

$$AG_{Actionneur} = R_A \times \left(\frac{1}{\left((1 + NL) + (2 - NS_Q) \right)^{R_A}} \right) \quad (3\text{-}7)$$

L'agrégation faite par l'actionneur est sensiblement identique à celle du nœud capteur, en raison des métriques qu'ils partagent. Ainsi, la réussite d'une action R_A détermine l'augmentation ou la diminution du niveau de confiance vu par l'actionneur. De même, les échecs sont plus fortement sanctionnés que les réussites, ainsi que les faibles signaux et les latences importantes. À noter qu'ici, l'aspect énergétique n'est pas pris en compte. Comme indiqué par l'auteur [295], l'action physique sur l'environnement requiert souvent une énergie importante. Les actionneurs jouissent ainsi d'une infrastructure adaptée, rendant cette ressource moins critique en comparaison avec les nœuds capteurs.

Comme précédemment, un scénario est proposé pour justifier le choix des métriques et calculs. Une approche volontairement simplifiée est adoptée, afin de ne pas complexifier le scénario. Le programme informatique d'un actionneur peut se découper en deux étapes. La première consiste à recueillir les données de nœuds capteurs ou provenant des passerelles. La seconde concerne la prise de décision (*i.e.* marche, arrêt). De ce fait, la réactivité de l'actionneur dépend de la stabilité de ses communications entrantes (*e.g.* qualité de signal pouvant entrainer des erreurs de transmission et de la latence). Aussi, la fiabilité des informations est à prendre en compte, afin d'assurer la sûreté des opérations de l'actionneur (*e.g.* présence de personnes à proximité). Dans ce scénario, le protocole TrustLib permettrait un auto-apprentissage continu des entités avec qui il est en contact. Au travers de l'évaluation de la confiance, il serait alors possible d'adapter le comportement de l'actionneur, afin d'optimiser son efficacité et sa qualité de service. Par exemple, privilégier la prise de décision sur les capteurs ayant une meilleure stabilité de communication (*i.e.* qualité de signal, latence), et menant le plus souvent à la réussite de l'opération de l'actionneur.

c) Passerelle

Les passerelles interagissent de multiples manières. D'un point de vue réseau local (passerelles à objets), la vision de la confiance reprend les éléments précédents, à savoir l'efficacité des communications et la fiabilité des informations. De plus, les transmissions d'un actionneur étant principalement dédiées à des fins de suivi/surveillance durant l'opération en cours (*i.e. monitoring*), il est possible d'en extraire les performances de ce dernier. Grâce aux ressources plus conséquentes que les nœuds capteurs et actionneurs, notamment en termes de capacité de calcul et de mémoire, il est possible d'élargir le modèle en procédant à une analyse continue des performances de chaque

entité, permettant un classement de ces dernières en termes de fiabilité. Cela se montre intéressant pour des questions de qualité de service/d'expérience aussi bien sur le réseau local que global. Par exemple, en favorisant la propagation des données les plus pertinentes, en incluant un indice de satisfaction, ou encore en prenant en compte l'offre et de la demande de données/ressources/services, dans un but de diversifier l'environnement.

De même que pour le nœud capteur et l'actionneur, cette section propose un modèle de confiance pour les passerelles. Cette première preuve de concept se veut volontairement simple, afin de valider la faisabilité d'un tel système. L'évaluation du niveau de confiance du point de vue des passerelles se base une nouvelle fois sur la qualité du signal, défini par la formule (3-4). Un score de fiabilité F_{Score} est également pris en compte, image de la pertinence des données provenant des nœuds capteurs et des performances des actionneurs issues de leur suivi (*monitoring*). En d'autres termes, il s'agit d'évaluer la qualité de service. Sa valeur normalisée est notée NF_{Score} définie par la formule (3-8). Un second score est attribué, pour indiquer cette fois l'utilité des interactions vis-à-vis du réseau, en guise de qualité d'expérience, notée U_{Score} et définie par la formule (3-9). Enfin, le calcul de l'agrégation $AG_{Passerelle}$ est présenté par la formule (3-10).

$$NF_{Score} = \frac{F_{Score} - F_{SMIN}}{F_{SMAX} - F_{SMIN}} \qquad NF_{Score} \in [0\,;1] \tag{3-8}$$

$$U_{Score} = \begin{cases} -1 \text{ interaction inutile} \\ 1 \text{ interaction utile} \end{cases} \tag{3-9}$$

$$AG_{Passerelle} = U_{Score} \times \left(\frac{1}{\left((2 - NF_{Score}) + (2 - NS_Q) \right)^{U_{Score}}} \right) \tag{3-10}$$

D'après la définition même de l'IdO, l'objectif est de tout interconnecter, dans le but d'échanger divers types de données. Un premier modèle d'attribution du U_{Score} est envisagé : en accord avec les principes de l'IdO, le partage d'une donnée sur le réseau est défini comme une interaction utile, favorisant l'enrichissement de l'écosystème. Le U_{Score} est de 1. Toute autre interaction (*e.g.* demande de reconfiguration, demande d'accès à une donnée, etc.) est considérée comme non utile en termes de valeur ajoutée sur le réseau. Le U_{Score} est de -1. L'intérêt d'un tel modèle est particulièrement intéressant pour favoriser la dissémination d'information sur le réseau, tout en dissuadant les échanges non essentiels.

Pour illustrer une nouvelle fois le choix des métriques, il peut être intéressant de faire le lien avec les scénarios précédents. Dans le premier, la passerelle est en charge de remonter le signal d'alerte du nœud capteur afin de déclencher la procédure de remplacement de batterie, ce qui implique le

déplacement d'un technicien. Une passerelle classique se chargerait de transmettre l'alerte. Dans un tel système, en cas de dysfonctionnement ou acte malveillant du nœud capteur, celui-ci peut continuellement transmettre le signal d'alerte, déclenchant ainsi une intervention à humaine à chaque fois. Ignorer l'alerte n'étant pas une solution viable (*e.g.* cas où l'alerte est bien réelle), un mécanisme de confiance se montre approprié. En effet, au travers du protocole TrustLib, la passerelle peut évaluer la fiabilité du nœud capteur, afin de s'assurer de la légitimité de l'alerte. En accord avec son modèle d'interaction utile/inutile (U_{Score}), le signal d'alerte est considéré comme inutile d'un point de vue enrichissement de l'écosystème. Cela signifie qu'un nœud transmettant continuellement de tels signaux verra son niveau de confiance diminuer. La passerelle est alors en mesure d'éviter d'inutiles interventions humaines.

Dans le scénario d'un actionneur, la passerelle a un double rôle. Elle peut être utilisée pour lui communiquer des données provenant d'autres capteurs, mais aussi de suivre et contrôler les opérations (*i.e. monitoring*). À l'aide du protocole TrustLib, et notamment au travers du score de satisfaction (NF_{Score}), la passerelle peut analyser et procéder automatiquement à des reconfigurations des paramètres de l'actionneur. Par exemple, en ajustant temporairement la fréquence des transmissions actionneur-passerelle, il est possible d'obtenir un meilleur suivi des opérations, permettant après analyse, d'améliorer les performances. Ainsi, au travers de ces différents scénarios, le protocole TrustLib se montre intéressant pour améliorer l'efficacité et la qualité de service d'un écosystème de l'IdO.

d) Niveau de confiance

En accord avec la définition de la confiance, donnée en début de ce chapitre (*cf.* 3.2.1.a), la confiance dépend de l'expérience, c'est-à-dire qu'elle varie en fonction des interactions passées et présentes. En conséquence, le niveau de confiance qu'une entité accorde à une autre à un instant donné est représenté par l'équation (3-11). À noter que la variable AG correspond à une des formules d'agrégation précédentes : $AG_{Capteur}$, $AG_{Actionneur}$ ou $AG_{Passerelle}$ suivant la nature de l'entité qui embarque le protocole TrustLib.

$$Confiance_n = Confiance_{n-1} + AG_n \qquad \text{avec } n \in \mathbb{N}^* \text{ et } Confiance_0 = 0 \tag{3-11}$$

3.6.7. Structure

Le protocole TrustLib est pensé pour être embarqué dans les objets connectés et doit prendre en compte leurs ressources souvent très limitées. Étant donné que le protocole se concentre sur l'environnement proche d'un objet, le nombre d'entités locales connectées sur une période spécifique

reste faible par rapport au réseau globale ou inter-passerelles. Par conséquent, il est possible de mettre en place une liste dynamique d'entités avec leur niveau de confiance associé, comme illustrée par la Figure 3-38. Par ailleurs, comme précisé en début de chapitre (*cf.* 3.2.1.a), la confiance est dynamique et éphémère. Afin d'inclure cet aspect dans le protocole TrustLib, chaque niveau attribué possède une date de validité, au bout de laquelle il diminue. Ce délai, ainsi que le coefficient de diminution est paramétrable suivant l'application. Il en est de même pour la taille de la liste, en fonction de la capacité de stockage de l'objet.

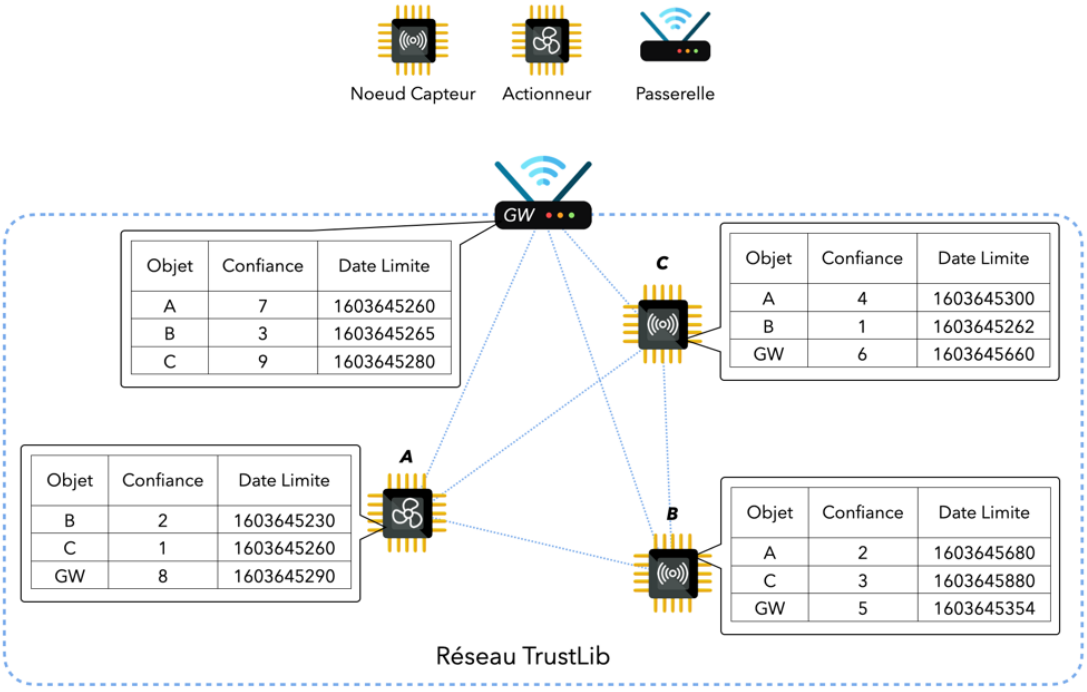


Figure 3-38 - Structure de données du protocole TrustLib

3.6.8. Gestion de la liste de confiance

Avant de poursuivre, il est important de noter que la confiance est nominative, c'est-à-dire qu'elle est attribuée à une entité spécifique. Un mécanisme assurant l'authenticité des interactions est donc nécessaire. Cela peut être intégré au niveau réseau et/ou applicatif. Par exemple, le LoRaWAN utilise plusieurs modes de l'AES-128, notamment le CMAC, à l'aide d'une clé spécifique de 128 bits (*NwkSKey*, *Network Session Key*) pour assurer l'intégrité et l'authenticité des paquets réseau [24]. Le problème de l'authentification dans le contexte de l'IdO est un domaine de recherche à part entière, et ne fait pas partie des problématiques adressées par le protocole TrustLib. La suite de ces travaux considère que chaque entité possède un mécanisme permettant d'authentifier de manière sûre ses interactions.

L'algorithme illustré par la Figure 3-39, représente l'intégralité du protocole TrustLib. La première phase consiste en l'évaluation des interactions avec une entité, en reprenant les différents calculs évoqués en section 3.6.6. Les configurations de ces derniers (*e.g.* valeurs pour les normalisations) dépendent des besoins de l'application. Un examen de la liste est ensuite effectué, afin de récupérer le précédent niveau de confiance de l'entité à évaluer, pour le mettre à jour comme défini par l'équation (3-11). Dans le cas où elle ne figure pas dans la liste, le niveau de confiance correspondra à la valeur de l'agrégation ($Confiance_{n-1} = 0$). Dans le même temps, le parcours de la liste est utilisé pour assurer la caractéristique éphémère de la confiance. Chaque entité dont la date limite est dépassée se voit attribuer une pénalité (valeur paramétrable suivant les spécificités de l'application). De nouvelles dates sont ensuite définies pour perpétuer la procédure. Afin d'éviter une utilisation inutile de l'espace de stockage, seuls les niveaux de confiance supérieurs à 0 sont conservés. Cela ouvre l'opportunité à de nouvelles entités d'être incluses dans la liste, en supprimant celles dont le niveau de confiance est nul. Le choix de ne pas utiliser de niveau négatif se justifie par la méthode d'identification des entités, basée sur des clés cryptographiques (*e.g. NwkSKey*, pour LoRaWAN). Cela signifie que le simple changement de ces dernières suffirait à outrepasser un niveau de confiance négatif et repartir sur le niveau par défaut 0.

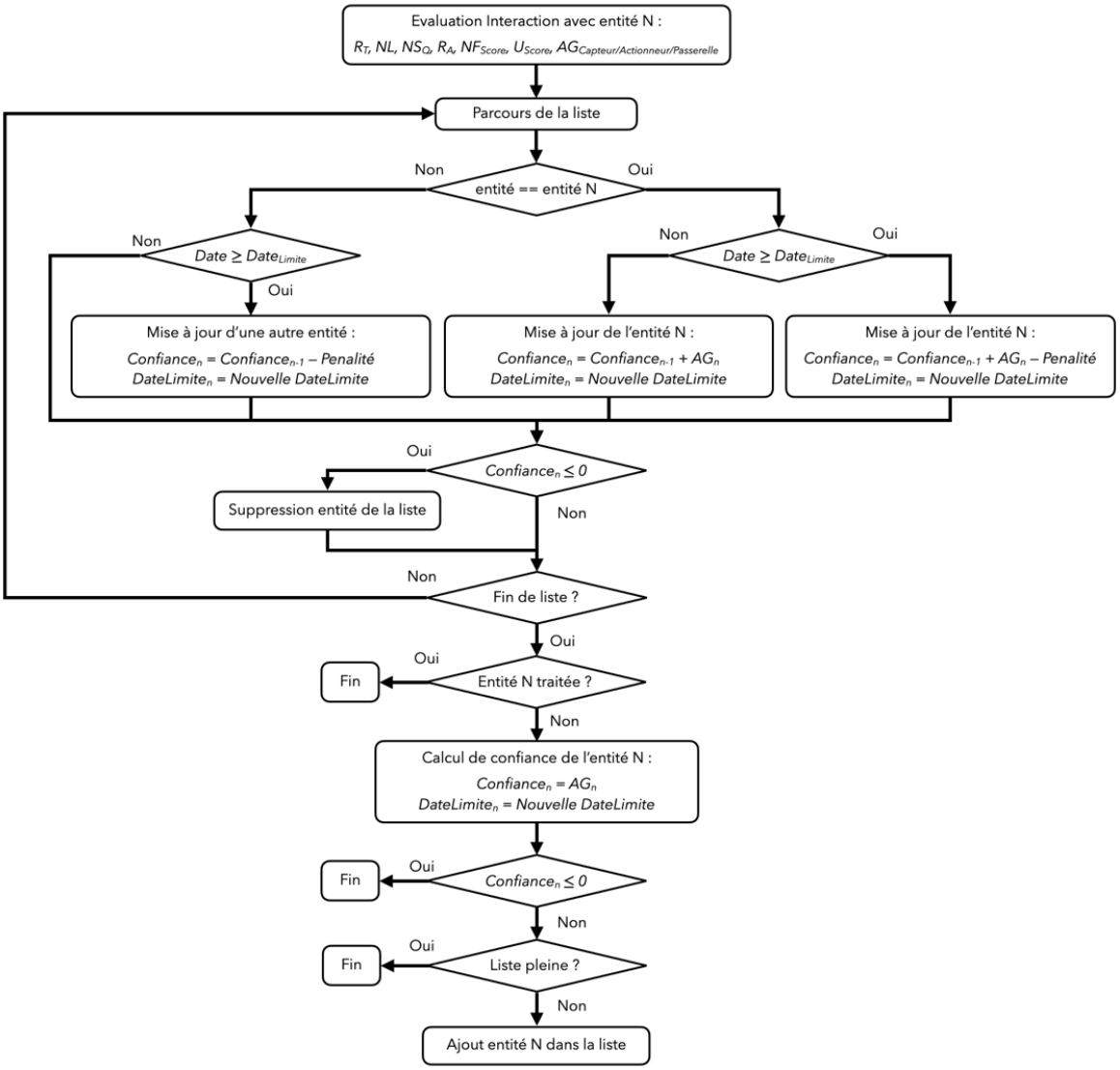


Figure 3-39 – Algorithme du protocole TrustLib embarqué dans chaque entité (nœud capteur, actionneur et passerelle)

3.6.9. Analyse de sécurité du protocole

Pour rappel, TrustLib repose uniquement sur les évaluations de confiance directe, le rendant insensible à toute attaque exploitant les mauvaises ou élogieuses réputations. Comme le montre le Tableau 3-12, les attaques « opportuniste » et « blanchiment » sont les plus pertinentes concernant TrustLib. La première, qui consiste à ne fournir des données/services fiables que lorsque son niveau de confiance est trop faible, peut être atténuée par la sévérité du protocole en cas d'interactions non fiables. Cela oblige l'entité malveillante à satisfaire son écosystème plus régulièrement, sans quoi elle sera isolée du système. Il en est de même pour les demandes d'accès ou des requêtes de configurations. Bien qu'elles puissent découler sur une meilleure efficacité du réseau à postériori, ce qui ne manquera

pas d'être récompensé, elles sont perçues comme sans valeur ajoutée d'un point de vue système à un instant donné. Cela permet d'inciter chaque membre du réseau à participer à l'enrichissement de son environnement, et de ne pas être un simple utilisateur.

La seconde attaque est dite de « blanchiment », dont le principe réside dans la déconnexion et reconnexion au réseau, dans le but d'effacer un mauvais niveau de confiance. Une autre approche consiste à créer une nouvelle identité. Le protocole TrustLib est construit de sorte que tant qu'un objet n'est pas enregistré dans la liste, son score est de 0, le plus faible qu'il soit (pas de valeur négative comme vue précédemment). De ce fait, une telle attaque ne montre aucun avantage et qui plus est, la variante utilisant la génération d'une nouvelle identité exige un minimum de ressources de calcul, pouvant même pénaliser l'attaquant lui-même.

3.6.10. Simulations

Cette section a pour but d'illustrer l'impact de chaque métrique sur l'évolution du niveau de confiance, suivant le modèle de chaque entité. La Figure 3-40 présente les résultats pour le nœud capteur, l'actionneur et la passerelle.

Courbes	Configurations des métriques		
	Noeud capteur	Actionneur	Passerelle
	(R_T = 1 ; NL = 0; NS_Q = 1; Bat = 1)	(R_A = 1 ; NL = 0; NS_Q = 1)	(U_{Score} = 1 ; NF_{Score} = 1; NS_Q = 1)
	(R_T = -1 ; NL = 0; NS_Q = 1; Bat = 1)	(R_A =-1 ; NL = 0; NS_Q = 1)	(U_{Score} = -1 ; NF_{Score} = 1; NS_Q = 1)
	(R_T = -1 ; NL = 1; NS_Q = 1; Bat = 1)	(R_A = -1 ; NL = 1; NS_Q = 1)	(U_{Score} = -1 ; NF_{Score} = 0; NS_Q = 1)
	(R_T = -1 ; NL = 1; NS_Q = 0; Bat = 1)	(R_A = -1 ; NL = 1; NS_Q = 0)	(U_{Score} = -1 ; NF_{Score} = 0; NS_Q = 0)
	(R_T = -1 ; NL = 1; NS_Q = 0; Bat = 0,1)	-	-

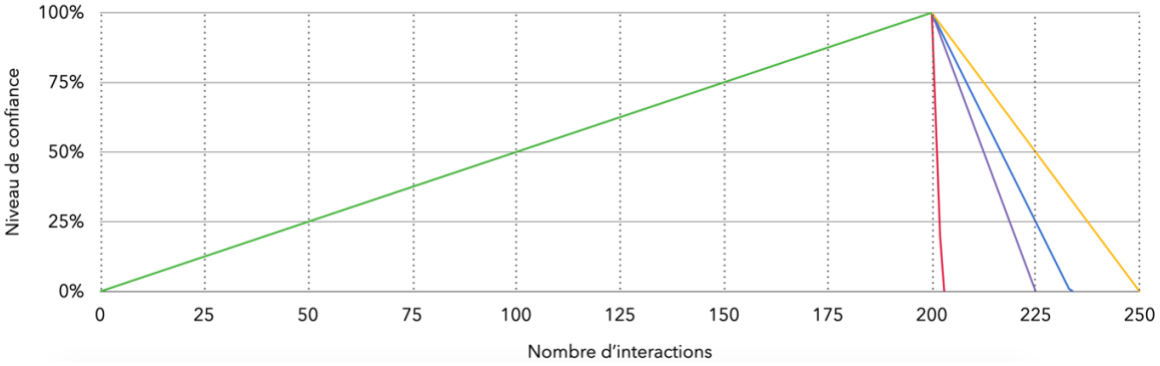


Figure 3-40 - Évolution du niveau de confiance vu par un nœud capteur/actionneur/passerelle

Les simulations ci-dessus illustrent la difficulté de gagner la confiance. Pas moins de 200 interactions dans des conditions optimales (*i.e.* réussite des transmissions, latence, qualité de signal et score de satisfaction), sont nécessaires entre les objets pour atteindre le niveau de confiance maximal. À l'inverse, entre 50 et 25 interactions, suffisent pour perdre toute confiance, suivant les métriques.

La diminution est la plus forte dans le cas d'un nœud capteur ayant son niveau de batterie au plus bas. Au bout de seulement 3 interactions infructueuses, ayant une qualité de signal minimale et une latence maximale, la confiance qu'un nœud capteur attribue à une entité est réduite à 0%. En reprenant le scénario du message d'alerte d'énergie du capteur, ce dernier sera en mesure de rapidement identifier le destinataire le plus apte à transmettre son signal. De même, les actionneurs sont en capacité d'isoler les capteurs les moins fiables, améliorant l'efficacité de leur prise de décision. Enfin, les passerelles peuvent suivre en temps réel les performances de chaque entité au travers de leur niveau de confiance, et détecter rapidement celles dont la configuration n'est pas optimale.

Outre l'évaluation du modèle de confiance, il est primordial de quantifier l'utilisation du stockage de TrustLib. Pour rappel, la liste de confiance est composée de l'ID des périphériques (*e.g.* clé cryptographique ou un *Extended Unique Identifier*, EUI), du niveau de confiance ainsi que de la date limite. À des fins d'évaluation, il a été choisi la cryptographie à courbe elliptique *secp256k1* du NIST, telle qu'utilisée dans Bitcoin [114] et dans Wallance, définissant une clé publique sur 33 octets [260]. Pour conserver la précision du niveau de confiance, ce dernier peut être stocké sous sa forme décimale (*float*, sur 4 octets). Toutefois, pour réduire au maximum l'espace de stockage nécessaire, il peut être envisagé de les convertir en entier sur seulement un octet. Cela offre un panel de 256 niveaux différents, ce qui peut être suffisant pour la plupart des applications. Enfin, la date est mémorisée sous la forme d'un *timestamp* de 32 bits (4 octets). Par conséquent, jusqu'à 41 octets sont nécessaires pour chaque entité stockée. La Figure 3-41 montre l'évolution de l'utilisation du stockage requit en fonction de la taille de la liste. Ainsi, pour un objet très limité, avec moins de 400 octets disponibles, TrustLib peut évaluer une dizaine d'entités. Avec un peu plus de capacités de stockage, environ 1 Ko sont nécessaires pour en suivre près d'une trentaine. Étant donné que le protocole se focalise sur un environnement de proximité, cela peut être suffisant pour de nombreuses applications. À titre de comparaison, la Figure 3-41 montre que TrustLib permet de réduire de près d'un facteur 2 la taille de l'espace de stockage par rapport à Wallance (version sans SQLite).

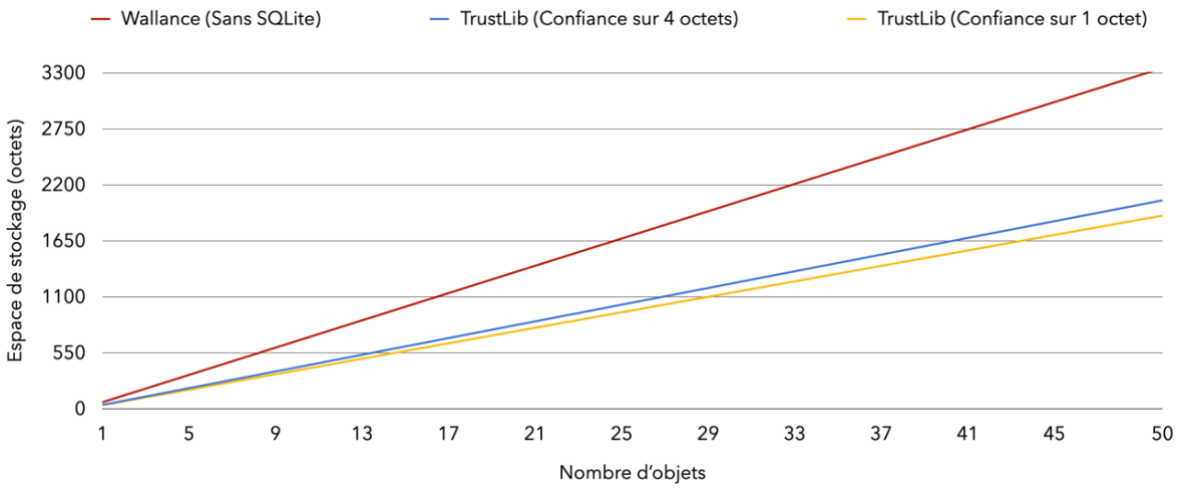


Figure 3-41 - Évolution de l'espace de stockage requis par TrustLib en fonction du nombre d'objets

3.6.11. Synthèse

L'expansion de l'IdO et son omniprésence engendre de fortes menaces de cybersécurité. Dans son approche la plus globale, c'est la confiance sur le réseau qui est mis à mal et qui se doit d'être perpétuellement assurée, et ce, de manière de plus en plus décentralisée. Malgré de nombreux efforts à tous les niveaux, de la standardisation des définitions et propriétés de la confiance, à des techniques de gestion efficaces, en passant par l'association avec les blockchains, il persiste toujours un manque de solution adaptée et facilement intégrable à l'IdO et plus particulièrement au niveau des objets eux-mêmes. Dans ce contexte, cette section présente une preuve de concept d'un nouveau protocole de confiance léger, nommé TrustLib. Son objectif est de sécuriser les objets dans un environnement proche, c'est-à-dire objets à objets et objets à passerelles, tout en favorisant la qualité de service et en minimisant la consommation d'énergie.

Étant donné les faibles ressources présentes dans les objets, la conception de TrustLib se doit d'associer simplicité et efficacité, afin qu'il puisse être embarqué dans ces derniers. Pensée dès le départ comme une extension à Wallance, l'association de ces deux protocoles permet de tirer pleinement profit de la diversité des entités constituant un écosystème d'IdO. En effet, alors que Wallance instaure la confiance entre les différents réseaux (passerelles à passerelles), TrustLib se focalise sur les objets (objets à objets et objets à passerelles). La force de cette synergie réside dans l'inclusion de toutes les entités du réseau, quelles que soient leurs capacités, à participer à la sécurisation de ce dernier, et ce, en prenant en compte les performances et la consommation des ressources. Les résultats soulignent la faible demande d'espace de stockage pour évaluer l'ensemble d'un environnement proche (une trentaine d'entités avec environ 1 Ko). De plus, les simulations ont permis d'illustrer l'adoption d'une propriété sociale de la confiance, à savoir le fait qu'il est plus

difficile de la gagner que de la perdre. Ceci a pour but d'inciter les entités de l'écosystème à continuellement respecter les règles, sous peine d'être isolées des autres (perte d'accès aux données/ressources et services). Enfin, grâce à l'approche de sa conception à la manière d'une bibliothèque logicielle, TrustLib peut facilement être intégré à Wallance comme un module complémentaire (*plug-in*), mais aussi à n'importe quel autre système, facilitant son utilisation et son déploiement.

3.7. Conclusion

De par l'omniprésence des objets connectés, les menaces de cybersécurité sont de plus en plus fortes. De manière générale, c'est la confiance sur le réseau qui se doit d'être assurée, et ce, dans une topologie réseau de plus en plus décentralisée. Bien qu'il s'agisse d'un domaine de recherche actif, il persiste toujours un manque de solution adaptée à l'IdO. Ce chapitre se concentre sur les deux grandes approches de la littérature, à savoir les systèmes de confiance et de réputation, et les systèmes blockchains.

Après analyse des différents verrous d'une adaptation de la blockchain pour l'IdO (puissance de calcul, stockage, scalabilité, énergie), un protocole plus léger, nommé Wallance, est proposé en guise d'alternative à la blockchain. Son objectif est de promouvoir le partage et l'utilisation de données/ressources et services, tout en instaurant la confiance dans le réseau, en sécurisant les interactions de manière décentralisée, et en limitant le besoin de ressources. Son modèle repose sur la monétisation des partages (données/ressources/services), cœur de concept de l'IdO, et de la participation des entités à assurer la sécurité de l'écosystème. Le protocole est évalué expérimentalement à l'aide d'une plateforme de prototypage, réalisée durant cette thèse, constituée de cartes Raspberry Pi 3B+ [189]. L'évaluation souligne la faible empreinte énergétique de Wallance (moins de 7 J par nœud), pour une latence 18 fois inférieure à celle de Bitcoin [114] et 2 fois moins que celles Ethereum [117] et IOTA [118]. Sa structure sous forme de blocs/portefeuilles-tressés lui offre à la fois souplesse et faible utilisation de ressources de stockage, pouvant gérer plusieurs centaines de millions de comptes, avec quelques dizaines de gigaoctets. Cela représente une réduction d'un facteur 3 000 par rapport à Bitcoin.

Bien que le protocole Wallance puisse être entièrement modulable, il est particulièrement adapté au niveau des passerelles réseau. Aussi, de par les communications directes entre les objets que permet un environnement de l'IdO, les passerelles ne sont pas toujours impliquées. C'est pourquoi un second protocole, nommé TrustLib, est proposé afin d'instaurer la confiance au plus près des objets. Pour qu'il puisse être embarqué par ces derniers, le protocole se doit d'associer simplicité et efficacité. Pour cela, les calculs de la confiance se basent sur des métriques quantifiables, tels que la qualité du

signal, la latence ou encore le niveau d'énergie disponible. Le modèle exploite une propriété sociale de la confiance, à savoir qu'il est plus difficile de la gagner que de la perdre, illustrée par les simulations. Cela permet d'inciter chaque entité à respecter les règles afin de ne pas être exclu de l'écosystème (perte d'accès aux données/ressources et services). D'autre part, l'évaluation montre le faible besoin d'espace de stockage nécessaire au suivi d'un environnement proche d'un objet (environ 1 Ko pour une trentaine d'entités). Enfin, l'approche de conception de TrustLib à la manière d'une bibliothèque logicielle lui offre une facilité d'utilisation et de déploiement dans n'importe quel système, comme par exemple, dans le protocole Wallance.

4. Conclusion et perspectives

4.1. Synthèse des travaux et résultats

Cette thèse s'intéresse à la sécurité d'un environnement Internet des Objets, et s'articule suivant deux axes : la sécurisation des interactions entre les objets, et la mise en place d'un mécanisme de confiance dans un écosystème décentralisé.

L'essence même de l'IdO repose sur la diversité et le partage d'innombrables données. La richesse de cet environnement repose presque entièrement sur les informations qui y circulent. De par leur démocratisation, les objets connectés deviennent omniprésents, avec pour conséquence la manipulation de données de plus en plus liée aux personnes (*e.g.* position, santé, compte bancaire, etc.). La question de la sécurité, et plus particulièrement de la confidentialité, devient un enjeu majeur. Pour répondre à cette problématique, des outils comme la cryptographie symétrique (*e.g.* l'AES) sont disponibles, mais présentent de fortes contraintes d'implémentation (*i.e.* taille, consommation énergétique, etc.). Pour remédier à cela, une alternative réside dans la cryptographie légère, dont font partie PRESENT et GIFT, partageant des propriétés avec l'AES. Afin de déterminer quel algorithme est le plus pertinent pour être embarqué sur les objets et assurer la confidentialité de leurs données, il est primordial de les évaluer. Étant données les fortes contraintes en termes de ressources et d'énergie, l'approche matérielle se montre particulièrement intéressante. C'est pourquoi, la première partie de cette thèse s'est concentrée sur la comparaison de l'AES-128, PRESENT-80/128 et GIFT-64-128/128-128 selon leur taille d'implémentation, leur efficacité et leur empreinte énergétique. L'aspect robustesse n'est pas écarté, avec notamment l'évaluation de leur résistance face aux attaques par canaux auxiliaires, et plus précisément, via l'analyse du rayonnement électromagnétique par la méthode CEMA. Pour résumer les résultats de cette évaluation, il est intéressant d'effectuer un classement des algorithmes, suivant les différents critères utilisés. Le Tableau 4-1 illustre cela, en attribuant les rangs de 1 à 5, du plus pertinent au moins.

Tableau 4-1 - Classement des algorithmes étudiés suivant les différents critères

Algorithmes	Taille	Efficacité	Énergie/bit	Robustesse
AES-128	5	5	3	5
PRESENT-80	2	3	4	2
PRESENT-128	3	4	5	**1**
GIFT-64-128	**1**	**1**	2	4
GIFT-128-128	4	2	**1**	3

Mis à part le cas de l'AES, qui cumule de nombreux inconvénients pour une utilisation dans un contexte d'IdO, le choix d'un algorithme de chiffrement n'est pas nécessairement une tâche triviale. Les résultats de ce premier axe soulignent la nécessité de faire des compromis sur la taille, la performance ou la sécurité. Toutefois, dès lors qu'il est question de confidentialité, cette dernière se montre particulièrement importante. À ce titre, l'algorithme PRESENT-128 se trouve être la meilleure solution de cette étude, de par sa plus grande robustesse, et ses performances certes inférieures, mais relativement proches de ses homologues.

Le second axe s'oriente sur une vision systémique de l'environnement IdO. La confidentialité n'est qu'une caractéristique de ce que doit être un environnement sécurisé, et d'autres mécanismes doivent être pris en considération. De plus, l'expansion de l'IdO souligne un fort intérêt dans le développement d'architectures de réseaux décentralisées. Favorisant les interactions directes entre entités, il en découle de meilleures performances en termes de réactivité et d'énergie (*e.g.* plus d'intermédiaires, limiter la sollicitation d'un point central exigeant souvent une communication longue distance). D'un point de vue sécurité, la décentralisation apporte son lot d'avantages, comme la suppression d'un point de défaillance unique (*single point of failure*), permettant à l'infrastructure de toujours fonctionner même en cas de panne et/ou d'attaque (*e.g.* DoS/DDoS, etc.). Toutefois, puisqu'aucun organisme central ne contrôle un tel système, ce dernier doit être entièrement autonome et assurer sa propre sécurité. De plus, étant donnée l'importance du partage de données et de ressources dans un environnement IdO, l'incitation à ces derniers doit être prise en considération. En d'autres termes, la viabilité et la sécurité d'un tel écosystème reposent sur sa capacité à instaurer et maintenir la confiance. Le concept de la blockchain fait aujourd'hui partie des solutions activement étudiées pour répondre au besoin de confiance et de sécurité dans un environnement décentralisé. Avec les cryptomonnaies comme applications phares, la blockchain offre de remarquables propriétés d'infalsifiabilité, de fonctionnement sans tiers, ainsi que d'auto-vérification et de dissémination d'information sur le réseau.

Malgré ses avantages, la blockchain est aujourd'hui beaucoup trop consommatrice de ressources (*i.e.* puissance de calcul, espace mémoire, énergie) pour une utilisation à l'IdO, fortement contraint. En conséquence, après analyse des différents points critiques (*i.e.* latence, stockage, capacité de calcul, etc.), les travaux de cette thèse ont mené à l'élaboration d'un protocole de confiance nommé Wallance, se positionnant comme une alternative légère à la blockchain. Il emploie un modèle de monétisation afin de promouvoir le partage et l'utilisation de données/ressources et services, en accord avec l'IdO. Spécialement conçu pour instaurer la confiance de manière décentralisée tout en étant suffisamment léger pour être embarqué dans les passerelles réseau (*gateways*), les résultats soulignent sa faible empreinte énergétique, 3 fois inférieure à celle d'une passerelle LoRa du commerce, ainsi que sa faculté à atteindre un consensus global de l'ordre de la seconde, soit 18 fois

plus rapide que Bitcoin. Enfin, son utilisation de l'espace de stockage est telle, qu'elle permet à Wallance de gérer plusieurs centaines de millions d'objets avec seulement quelques dizaines de gigaoctets, largement atteignables par les passerelles.

Alors que Wallance possède une grande flexibilité lui permettant d'être implémenté dans une multitude de systèmes, y compris les objets connectés, son fonctionnement est optimal au niveau des passerelles réseau. Avec comme double objectif de combler un manque de solution au plus proche des objets, mais aussi d'améliorer le protocole Wallance, un second projet, nommé TrustLib a été mise en place durant cette thèse. Cette association permet alors de tirer pleinement profit de toutes les entités d'un écosystème. L'idée principale est d'assurer la viabilité des données, et par conséquent la qualité de service, en intégrant un système de notation. Plus une entité partage de l'information utile et de manière efficace (*e.g.* qualité de transmission), plus elle est considérée comme digne de confiance. En guise d'incitation, cette dernière peut d'une part continuer à faire activement partie du réseau, mais se voit également accorder des privilèges, tel que l'accès à des ressources ou à des données particulières. De plus, le protocole emploie un modèle permettant de simuler des propriétés sociales de la confiance, à savoir son aspect éphémère, ainsi que la difficulté de la gagner, mais une facilité à la perdre. Cela permet d'intensifier l'incitation au respect des règles, sous peine d'être isolé de l'écosystème. À noter que le champ d'action de TrustLib se focalise sur l'environnement proche, c'est-à-dire uniquement sur les objets aux alentours avec qui ils interagissent. Il peut être supposé que leur nombre n'excède guère les quelques dizaines, pour la majorité des applications. Les résultats stipulent qu'avec moins de 400 octets, TrustLib est capable de suivre une dizaine d'entités, et jusqu'à près d'une trentaine avec environ 1 Ko. Cela représente une diminution de près d'un facteur 2 comparé à Wallance. Enfin, l'approche choisie pour la conception de TrustLib, à la manière d'une bibliothèque logicielle, lui permet de faciliter son intégration et son déploiement aussi bien dans les nouveaux systèmes que dans ceux déjà existants, comme par exemple dans Wallance.

4.2. Perspectives

Bien que PRESENT-128 soit le plus pertinent des cinq algorithmes de chiffrement étudiés dans cette thèse, force est de constater qu'ils se montrent tous hautement vulnérables à l'attaque par canaux auxiliaires CEMA. Entre 1 100 et 3 400 traces sont nécessaires pour retrouver la clé, ce qui représente seulement quelques heures de manipulation dans le pire des cas. L'accessibilité des objets connectés et le faible cout matériel de l'attaque par observation du rayonnement électromagnétique la rendent particulièrement dangereuse. C'est pourquoi l'implémentation de contremesures est une piste à explorer et à évaluer en termes de cout sur la taille, l'efficacité, l'énergie, et bien évidemment sur la robustesse. Elles se décomposent suivant deux familles. La première correspond à la dissimulation (*hiding*), dont l'objectif est de diminuer le rapport signal sur bruit des traces

d'acquisition afin de dissimuler les fuites. Une technique employée consiste en la désynchronisation temporelle des traces, à l'aide d'une horloge instable (*clock jitter*) dans une implémentation matérielle. La seconde famille de contremesures est le masquage (*masking*) qui tente de rendre les valeurs intermédiaires de l'algorithme de chiffrement indépendantes des fuites. Le principe général est de scinder la valeur à masquer en plusieurs parts aléatoires de même taille, de sorte que l'ensemble de ces dernières soit indispensable pour remonter à la valeur initiale. Bien que théoriquement viable, l'implémentation des techniques de masquage doivent faire l'objet d'une attention toute particulière, en raison de leur forte sensibilité aux phénomènes physiques tels que les délais de propagation (*glitches*), dans le cas des implémentations matérielles. Ces derniers induisent des retards sur les entrées des opérateurs logiques, qui subissent alors des transitions jusqu'à ce que toutes leurs entrées soient disponibles. Ainsi, les valeurs intermédiaires peuvent alors fuiter au travers de ces transitions, mettant en échec la contremesure, comme démontré sur une implémentation de l'AES [296]. Pour contrebalancer les effets de ces phénomènes physiques, des techniques d'implémentation à seuil (*Threshold Implementation*, TI) [297] ont été proposées. Toutefois, leur cout notamment en termes de taille sont loin d'être négligeables, d'un facteur 5 au minimum d'après les études [76] et [80]. Ainsi, l'étude et la mise en place de contremesures efficaces et n'exigeant qu'un faible surcout en termes de ressources, apparait comme crucial pour assurer la confidentialité d'objets fortement contraints.

Le modèle actuel du protocole Wallance valorise la quantité de données partagées. Afin d'être plus en phase avec une utilisation réelle, une autre approche doit être mise en place. En effet, l'objectif final est de favoriser l'échange d'information, et ce, de manière efficace, dont l'association avec TrustLib en est un premier pas. L'efficacité énergétique et la qualité de service constituent les pistes envisagées, dans une optique d'amélioration de l'environnement IdO. Enfin, les capacités de robustesse de Wallance face aux cyberattaques doivent être investiguées. Pour cela, il convient d'implémenter et de soumettre le protocole aux plus pertinentes d'entre elles (*e.g.* attaque Sybil), afin d'apporter des contremesures efficaces. De plus, la question de l'ajout de nouveaux nœuds dynamiquement apporte son lot de problématiques en matière de sécurité. Comment sécuriser la synchronisation de ces derniers avec la blockchain actuelle (*i.e. bootstrap*) ? Comment l'évolution de la taille du réseau, utilisée lors du consensus à la majorité, est-elle gérée par les nœuds ?

Bien qu'il puisse être un modèle complémentaire à Wallance sur les passerelles, l'implémentation du protocole TrustLib sur de vraies architectures d'objets connectés (*e.g.* STM32 Nucleo [282], JN5189 [208], ESP32 [283]), avec différents protocoles de communication comme le LoRa et le ZigBee est primordiale. Il serait alors possible de quantifier l'impact du protocole en termes de taille de code, de temps d'exécution (calculs des métriques, gestion de la liste de confiance), et l'énergie que cela implique. De manière plus systémique, l'implémentation de TrustLib en conditions réelles

permettrait également d'évaluer sa capacité à améliorer à la fois la qualité de service, tout en diminuant la consommation énergétique des entités de l'écosystème.

De manière plus générale, une extension de la plateforme de prototypage est envisagée. Grâce à l'augmentation du nombre de nœuds connectés, il serait alors possible d'évaluer plus concrètement les performances du protocole Wallance, suivant des tailles réseau plus conséquentes. De plus, cela permettrait d'exploiter le concept du *sharding*, consistant à interconnecter plusieurs sous-réseaux Wallance. Une manière de réaliser cela serait de se munir de plusieurs plateformes distinctes, séparées géographiquement. Usant actuellement de cartes Raspberry Pi 3B+ [189], il serait également intéressant d'évoluer vers une architecture SoC (*System-on-Chip*), afin de décupler le potentiel des passerelles de la plateforme. Celles-ci peuvent par exemple être implémentées par des cartes Zybo [298], intégrant à la fois un processeur et une partie logique reprogrammable. Le protocole Wallance pourrait y être embarqué (logiciellement) et amélioré par des accélérateurs matériels, notamment pour le LWPoW et la signature numérique. De même, la plateforme pourrait également intégrer des prototypages de vraies architectures d'objets connectés usant de différents protocoles de communication, et embarquant le protocole TrustLib. Enfin, cela offrirait l'opportunité de lier les deux axes de cette thèse, en implémentant dans les objets les algorithmes de chiffrement légers étudiés. Ainsi, toutes ces mises en place permettraient de faire évoluer la plateforme du stade de prototypage, à un écosystème complet et opérationnel, offrant un support de recherche pertinent pour de nouvelles investigations liées à l'Internet des Objets.

Bibliographie

[1] Mark Weiser, "The Computer for the 21st Century.pdf." Scientific American, Sep. 1991.

[2] International Telecommunication Union (ITU), "Termes et définitions applicables à l'Internet des objets," Jul. 2012.

[3] M. S. Mekala and P. Viswanathan, "A Survey: Smart agriculture IoT with cloud computing," in *2017 International conference on Microelectronic Devices, Circuits and Systems (ICMDCS)*, Vellore, Aug. 2017, pp. 1–7, doi: 10.1109/ICMDCS.2017.8211551.

[4] Y. Lu, "Industry 4.0: A survey on technologies, applications and open research issues," *Journal of Industrial Information Integration*, vol. 6, pp. 1–10, Jun. 2017, doi: 10.1016/j.jii.2017.04.005.

[5] C. A. Medina, M. R. Perez, and L. C. Trujillo, "IoT Paradigm into the Smart City Vision: A Survey," in *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, Exeter, Jun. 2017, pp. 695–704, doi: 10.1109/iThings-GreenCom-CPSCom-SmartData.2017.109.

[6] "StopCovid," [Online]. Available: https://www.gouvernement.fr/info-coronavirus/stopcovid.

[7] G. A. Akpakwu, B. J. Silva, G. P. Hancke, and A. M. Abu-Mahfouz, "A Survey on 5G Networks for the Internet of Things: Communication Technologies and Challenges," *IEEE Access*, vol. 6, pp. 3619–3647, 2018, doi: 10.1109/ACCESS.2017.2779844.

[8] International Telecommunication Union (ITU), "Présentation générale de l'Internet des objets." Juin 2012.

[9] David Mercer, "Global Connected and IoT Device Forecast Update," *Strategy Analytics Consumer Electronics*, page 6, mai 2019, [Online]. Available: https://www.strategyanalytics.com/access-services/devices/connected-home/consumer-electronics/reports/report-detail/global-connected-and-iot-device-forecast-update.

[10] D. Evans, "How More Relevant and Valuable Connections Will Change the World," *Cisco Internet Business Solutions Group (IBSG)*, p. 9.

[11] X. Bonnetain *et al.*, "Le traçage anonyme, dangereux oxymore," p. 13.

[12] M. Antonakakis *et al.*, "Understanding the Mirai Botnet," *26th USENIX Security Symposium (USENIX Security 17), USENIX Association,* pp. 1093–1110.

[13] J. Daemen and V. Rijmen, "The Rijndael Block Cipher," p. 47, 1999.

[14] A. Bogdanov *et al.*, "PRESENT: An Ultra-Lightweight Block Cipher," in *Cryptographic Hardware and Embedded Systems - CHES 2007*, vol. 4727, P. Paillier and I. Verbauwhede, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 450–466.

[15] S. Banik, S. K. Pandey, T. Peyrin, Y. Sasaki, S. M. Sim, and Y. Todo, "GIFT: A Small Present," in *Cryptographic Hardware and Embedded Systems – CHES 2017*, vol. 10529, W. Fischer and N. Homma, Eds. Cham: Springer International Publishing, 2017, pp. 321–345.

[16] P. Zande, "The Day DES Died," *SANS Institute 2001*, p. 10, 2001.

[17] C. Shannon and W. Weaver, "The Mathematical Theory of Communication," *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[18] O. Dunkelman and N. Keller, "The effects of the omission of last round's MixColumns on AES," *Information Processing Letters*, vol. 110, no. 8–9, pp. 304–308, Apr. 2010, doi: 10.1016/j.ipl.2010.02.007.

[19] L. Batina, D. Jakobovic, N. Mentens, S. Picek, A. de la Piedra, and D. Sisejkovic, "S-box Pipelining Using Genetic Algorithms for High-Throughput AES Implementations: How Fast Can We Go?," in *Progress in Cryptology -- INDOCRYPT 2014*, vol. 8885, W. Meier and D. Mukhopadhyay, Eds. Cham: Springer International Publishing, 2014, pp. 322–337.

[20] S. Banik, A. Bogdanov, and F. Regazzoni, "Atomic-AES: A Compact Implementation of the AES Encryption/Decryption Core," in *Progress in Cryptology – INDOCRYPT 2016*, vol. 10095, O. Dunkelman and S. K. Sanadhya, Eds. Cham: Springer International Publishing, 2016, pp. 173–190.

[21] R. Ueno, S. Morioka, N. Homma, and T. Aoki, "A High Throughput/Gate AES Hardware Architecture by Compressing Encryption and Decryption Datapaths: — Toward Efficient CBC-Mode Implementation," in *Cryptographic Hardware and Embedded Systems – CHES 2016*, vol. 9813, B. Gierlichs and A. Y. Poschmann, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 538–558.

[22] D. A. Osvik, J. W. Bos, D. Stefan, and D. Canright, "Fast Software AES Encryption," in *Fast Software Encryption*, vol. 6147, S. Hong and T. Iwata, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 75–93.

[23] P. Schwabe and K. Stoffelen, "All the AES You Need on Cortex-M3 and M4," in *Selected Areas in Cryptography – SAC 2016*, vol. 10532, R. Avanzi and H. Heys, Eds. Cham: Springer International Publishing, 2017, pp. 180–194.

[24] A. Adomnicai, "Cryptographie Légère pour l'Internet des Objets: Implémentations et Intégrations Sécurisées," PhD Thesis, Université de Lyon, 2019.

[25] J. Guo, T. Peyrin, A. Poschmann, and M. Robshaw, "The LED Block Cipher," in *Cryptographic Hardware and Embedded Systems – CHES 2011*, vol. 6917, B. Preneel and T. Takagi, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 326–341.

[26] S. Banik *et al.*, "Midori: A Block Cipher for Low Energy (Extended Version)," *ASIACRYPT 2015*, vol. 9453, pp. 411–436, 2015, doi: 10.1007/978-3-662-48800-3_17.

[27] C. Beierle *et al.*, "The SKINNY Family of Block Ciphers and Its Low-Latency Variant MANTIS," in *Advances in Cryptology – CRYPTO 2016*, vol. 9815, M. Robshaw and J. Katz,

Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 123–153.

[28] International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC), "ISO/IEC 29192-2 Information security — Lightweight cryptography — Part 2: Block ciphers," 2019, [Online]. Available: https://www.iso.org/obp/ui/#iso:std:iso-iec:29192:-2:ed-2:v1:en.

[29] A. Bogdanov and V. Rijmen, "Linear hulls with correlation zero and linear cryptanalysis of block ciphers," *Des. Codes Cryptogr.*, vol. 70, no. 3, pp. 369–383, Mar. 2014, doi: 10.1007/s10623-012-9697-z.

[30] A. Poschmann, "Lightweight Cryptography, Cryptographic Engineering for a Pervasive World," PhD Thesis, Faculty of Electrical Engineering and Information Technology Ruhr-University Bochum, Germany, IACR Cryptology, 2009.

[31] A. Heuser, S. Picek, S. Guilley, and N. Mentens, "Side-channel Analysis of Lightweight Ciphers: Current Status and Future Directions," *Lightweight Cryptography Workshop 2016*, p. 15, 2016.

[32] P. C. Kocher, "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems," *Advances in Cryptology: Proceedings of CRYPTO' 96*, pp. 104–113, 1996.

[33] D. J. Bernstein, "Cache-timing attacks on AES," p. 37, 2005.

[34] A. Moradi, "Side-Channel Leakage through Static Power: Should We Care about in Practice?," in *Advanced Information Systems Engineering*, vol. 7908, C. Salinesi, M. C. Norrie, and Ó. Pastor, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 562–579.

[35] J.-J. Quisquater, D. Samyde, I. Attali, and T. Jensen, "ElectroMagnetic Analysis (EMA) : Measures and Counter-measures for Smart Cards," *Smart Card Programming and Security, International Conference on Research in Smart Cards*, vol. 2140, pp. 200–210, 2001.

[36] D. Dinu, A. Biryukov, J. Großschädl, D. Khovratovich, Y. L. Corre, and L. Perrin, "FELICS - Fair Evaluation of Lightweight Cryptographic Systems," *NIST Workshop on Lightweight Cryptography 2015, National Institute of Standards and Technology (NIST) (2015)*, p. 14.

[37] Microchip Technology Inc, "AVR ATmega 8 bits," [Online]. Available: https://www.microchip.com/wwwproducts/en/ATMEGA128.

[38] Texas Instruments Incorporated, "TI MSP430 16 bits," [Online]. Available: http://www.ti.com/microcontrollers/msp430-ultra-low-power-mcus/overview.html.

[39] Arm Limited, "ARM Cortex M3 32 bits," [Online]. Available: https://developer.arm.com/ip-products/processors/cortex-m/cortex-m3.

[40] T. Good and M. Benaissa, "AES on FPGA from the Fastest to the Smallest," in *Cryptographic Hardware and Embedded Systems – CHES 2005*, vol. 3659, J. R. Rao and B. Sunar, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 427–440.

[41] Xinmiao Zhang and K. K. Parhi, "High-speed VLSI architectures for the AES algorithm,"

IEEE Trans. VLSI Syst., vol. 12, no. 9, pp. 957–967, Sep. 2004, doi: 10.1109/TVLSI.2004.832943.

[42] Xilinx, "Xilinx Vivado Design Suite User Guide," [Online]. Available: https://www.xilinx.com/support/documentation/sw_manuals/xilinx2016_1/ug901-vivado-synthesis.pdf.

[43] Xilinx, "Xilinx Spartan II Families," [Online]. Available: https://www.xilinx.com/support/documentation/data_sheets/ds001.pdf.

[44] Xilinx, "Xilinx Virtex-4 FPGA User Guide," [Online]. Available: https://www.xilinx.com/support/documentation/user_guides/ug070.pdf.

[45] Xilinx, "Xilinx Spartan-3 Generation FPGA User Guide," [Online]. Available: https://www.xilinx.com/support/documentation/user_guides/ug331.pdf.

[46] Xilinx, "Xilinx Virtex-5 FPGA User Guide," [Online]. Available: https://www.xilinx.com/support/documentation/user_guides/ug190.pdf.

[47] Xilinx, "Xilinx Spartan-6 FPGA Configurable Logic Block," [Online]. Available: https://www.xilinx.com/support/documentation/user_guides/ug384.pdf.

[48] Xilinx, "Xilinx 7 Series FPGAs Configurable Logic Block User Guide (UG474)," [Online]. Available: https://www.xilinx.com/support/documentation/user_guides/ug474_7Series_CLB.pdf.

[49] International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC), "ISO/IEC 14443-2:2016 Identification cards — Contactless integrated circuit cards — Proximity cards — Part 2: Radio frequency power and signal interface," [Online]. Available: https://www.iso.org/standard/66288.html.

[50] C.-J. Chang, C.-W. Huang, H.-Y. Tai, M.-Y. Lin, and T.-K. Hu, "8-bit AES FPGA Implementation using Block RAM," in *IECON 2007 - 33rd Annual Conference of the IEEE Industrial Electronics Society*, Taipei, Taiwan, 2007, pp. 2654–2659, doi: 10.1109/IECON.2007.4460363.

[51] Y.-S. Jeon, Y.-J. Kim, and D.-H. Lee, "A Compact Memory-Free Architecture For The AES Algorithm Using Resource Sharing Methods," *J CIRCUIT SYST COMP*, vol. 19, no. 05, pp. 1109–1130, Aug. 2010, doi: 10.1142/S0218126610006633.

[52] P. Chodowiec and K. Gaj, "Very Compact FPGA Implementation of the AES Algorithm," in *Cryptographic Hardware and Embedded Systems - CHES 2003*, vol. 2779, C. D. Walter, Ç. K. Koç, and C. Paar, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 319–333.

[53] N. Hanley and M. ONeill, "Hardware Comparison of the ISO/IEC 29192-2 Block Ciphers," in *2012 IEEE Computer Society Annual Symposium on VLSI*, Amherst, MA, USA, Aug. 2012, pp. 57–62, doi: 10.1109/ISVLSI.2012.25.

[54] G. Rouvroy, F.-X. Standaert, J.-J. Quisquater, and J.-D. Legat, "Compact and efficient encryption/decryption module for FPGA implementation of the AES Rijndael very well

suited for small embedded applications," in *International Conference on Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004.*, Las Vegas, NV, USA, 2004, pp. 583-587 Vol.2, doi: 10.1109/ITCC.2004.1286716.

[55] P. Bulens, F.-X. Standaert, J.-J. Quisquater, P. Pellegrin, and G. Rouvroy, "Implementation of the AES-128 on Virtex-5 FPGAs," in *Progress in Cryptology – AFRICACRYPT 2008*, vol. 5023, S. Vaudenay, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 16–26.

[56] J. Chu and M. Benaissa, "Low area memory-free FPGA implementation of the AES algorithm," in *22nd International Conference on Field Programmable Logic and Applications (FPL)*, Oslo, Norway, Aug. 2012, pp. 623–626, doi: 10.1109/FPL.2012.6339250.

[57] Helion Technology, "AES Tiny Core," [Online]. Available: https://www.heliontech.com/aes_tiny.htm.

[58] Helion Technology, "AES Standard Core," [Online]. Available: https://www.heliontech.com/aes_std.htm.

[59] X. Guo, Z. Chen, and P. Schaumont, "Energy and Performance Evaluation of an FPGA-Based SoC Platform with AES and PRESENT Coprocessors," in *Embedded Computer Systems: Architectures, Modeling, and Simulation*, vol. 5114, M. Bereković, N. Dimopoulos, and S. Wong, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 106–115.

[60] Helion Technology, "AES Fast Core," [Online]. Available: https://www.heliontech.com/aes_fast.htm.

[61] A. Aghaie, A. Moradi, S. Rasoolzadeh, A. R. Shahmirzadi, F. Schellenberg, and T. Schneider, "Impeccable Circuits," *IEEE Trans. Comput.*, vol. 69, no. 3, pp. 361–376, Mar. 2020, doi: 10.1109/TC.2019.2948617.

[62] "Hadipourh - AES-128," [Online]. Available: https://github.com/hadipourh/AES-VHDL.

[63] W. Diehl, F. Farahmand, P. Yalla, J.-P. Kaps, and K. Gaj, "Comparison of hardware and software implementations of selected lightweight block ciphers," in *2017 27th International Conference on Field Programmable Logic and Applications (FPL)*, Ghent, Belgium, Sep. 2017, pp. 1–4, doi: 10.23919/FPL.2017.8056808.

[64] C. A. Lara-Nino, A. Diaz-Perez, and M. Morales-Sandoval, "Lightweight Hardware Architectures for the Present Cipher in FPGA," *IEEE Trans. Circuits Syst. I*, vol. 64, no. 9, pp. 2544–2555, Sep. 2017, doi: 10.1109/TCSI.2017.2686783.

[65] C. A. Lara-Nino, M. Morales-Sandoval, and A. Diaz-Perez, "Novel FPGA-Based Low-Cost Hardware Architecture for the PRESENT Block Cipher," in *2016 Euromicro Conference on Digital System Design (DSD)*, Limassol, Cyprus, Aug. 2016, pp. 646–650, doi: 10.1109/DSD.2016.46.

[66] M. Sbeiti, M. Silbermann, A. Poschmann, and C. Paar, "Design space exploration of present implementations for FPGAS," in *2009 5th Southern Conference on Programmable Logic (SPL)*, São Carlos, Brazil, Apr. 2009, pp. 141–145, doi: 10.1109/SPL.2009.4914893.

[67] H. D. Azari and D. P. V. Joshi, "An Efficient Implementation Of Present Cipher Model With 80 Bit And 128 Bit Key Over Fpga Based Hardware Architecture," *International Journal of Pure and Applied Mathematics*, vol. 119, no. 14, pp. 1825–1832, 2018.

[68] J. G. Pandey, T. Goel, and A. Karmakar, "Hardware architectures for PRESENT block cipher and their FPGA implementations," *IET Circuits, Devices & Systems*, vol. 13, no. 7, pp. 958–969, Oct. 2019, doi: 10.1049/iet-cds.2018.5273.

[69] J. G. Pandey, T. Goel, and A. Karmakar, "An Efficient VLSI Architecture for PRESENT Block Cipher and Its FPGA Implementation," in *VLSI Design and Test*, vol. 711, B. K. Kaushik, S. Dasgupta, and V. Singh, Eds. Singapore: Springer Singapore, 2017, pp. 270–278.

[70] "Huljar - PRESENT-80:128," [Online]. Available: https://github.com/huljar/present-vhdl.

[71] P. Yalla and J.-P. Kaps, "Lightweight Cryptography for FPGAs," in *2009 International Conference on Reconfigurable Computing and FPGAs*, Cancun, Mexico, Dec. 2009, pp. 225–230, doi: 10.1109/ReConFig.2009.54.

[72] E. B. Kavun and T. Yalcin, "RAM-Based Ultra-Lightweight FPGA Implementation of PRESENT," in *2011 International Conference on Reconfigurable Computing and FPGAs*, Cancun, Mexico, Nov. 2011, pp. 280–285, doi: 10.1109/ReConFig.2011.74.

[73] J. J. Tay, M. L. D. Wong, M. M. Wong, C. Zhang, and I. Hijazin, "Compact FPGA implementation of PRESENT with Boolean S-Box," in *2015 6th Asia Symposium on Quality Electronic Design (ASQED)*, Kula Lumpur, Malaysia, Aug. 2015, pp. 144–148, doi: 10.1109/ACQED.2015.7274024.

[74] C. A. Lara-Nino, A. Diaz-Perez, and M. Morales-Sandoval, "FPGA-Based Assessment of Midori and Gift Lightweight Block Ciphers," in *Information and Communications Security*, vol. 11149, D. Naccache, S. Xu, S. Qing, P. Samarati, G. Blanc, R. Lu, Z. Zhang, and A. Meddahi, Eds. Cham: Springer International Publishing, 2018, pp. 745–755.

[75] "Espresso," [Online]. Available: https://ptolemy.berkeley.edu/projects/embedded/pubs/downloads/espresso/index.htm.

[76] A. Jati, N. Gupta, A. Chattopadhyay, S. K. Sanadhya, and D. Chang, "Threshold Implementations of GIFT : A Trade-Off Analysis," *IEEE Trans.Inform.Forensic Secur.*, vol. 15, pp. 2110–2120, 2020, doi: 10.1109/TIFS.2019.2957974.

[77] J. P. Oliver, J. Curto, D. Bouvier, M. Ramos, and E. Boemo, "Clock gating and clock enable for FPGA power reduction," in *2012 VIII Southern Conference on Programmable Logic*, Bento Goncalves, Mar. 2012, pp. 1–5, doi: 10.1109/SPL.2012.6211782.

[78] Xilinx, "Digilent Nexys 4 Artix-7 FPGA Trainer Board," [Online]. Available: https://www.xilinx.com/products/boards-and-kits/1-3yznp5.html#overview.

[79] Xilinx, "Xilinx 7 Series FPGA Libraries Guide for HDL Designs," [Online]. Available: https://www.xilinx.com/support/documentation/sw_manuals/xilinx14_1/7series_hdl.pdf.

[80] V. Satheesh and D. Shanmugam, "Secure Realization of Lightweight Block Cipher: A Case

Study Using GIFT," in *Security, Privacy, and Applied Cryptography Engineering*, vol. 11348, A. Chattopadhyay, C. Rebeiro, and Y. Yarom, Eds. Cham: Springer International Publishing, 2018, pp. 85–103.

[81] F. -x. Standaert, G. Rouvroy, and J. -j. Quisquater, "FPGA Implementations of the DES and Triple-DES Masked Against Power Analysis Attacks," in *2006 International Conference on Field Programmable Logic and Applications*, Madrid, Spain, 2006, pp. 1–4, doi: 10.1109/FPL.2006.311315.

[82] J. Ge, Y. Xu, R. Liu, E. Si, N. Shang, and A. Wang, "Power Attack and Protected Implementation on Lightweight Block Cipher SKINNY," in *2018 13th Asia Joint Conference on Information Security (AsiaJCIS)*, Guilin, China, Aug. 2018, pp. 69–74, doi: 10.1109/AsiaJCIS.2018.00020.

[83] S. Patranabis, J. Breier, D. Mukhopadhyay, and S. Bhasin, "One Plus One is More than Two: A Practical Combination of Power and Fault Analysis Attacks on PRESENT and PRESENT-Like Block Ciphers," in *2017 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*, Taipei, Sep. 2017, pp. 25–32, doi: 10.1109/FDTC.2017.11.

[84] J. Breier, D. Jap, and S. Bhasin, "SCADPA: Side-channel assisted differential-plaintext attack on bit permutation based ciphers," in *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, Dresden, Germany, Mar. 2018, pp. 1129–1134, doi: 10.23919/DATE.2018.8342180.

[85] S. Patranabis, N. Datta, D. Jap, J. Breier, S. Bhasin, and D. Mukhopadhyay, "SCADFA: Combined SCA+DFA Attacks on Block Ciphers with Practical Validations," *IEEE Trans. Comput.*, vol. 68, no. 10, pp. 1498–1510, Oct. 2019, doi: 10.1109/TC.2019.2913644.

[86] J. Breier, D. Jap, X. Hou, and S. Bhasin, "On Side Channel Vulnerabilities of Bit Permutations in Cryptographic Algorithms," *IEEE Trans.Inform.Forensic Secur.*, vol. 15, pp. 1072–1085, 2020, doi: 10.1109/TIFS.2019.2932230.

[87] R. Anderson and M. Kuhn, "Tamper Resistance — a Cautionary Note," *The Second USENIX Workshop on Electronic Commerce Proceedings*, vol. 2, pp. 1–11, 1996.

[88] S. P. Skorobogatov and R. J. Anderson, "Optical Fault Induction Attacks," in *Cryptographic Hardware and Embedded Systems - CHES 2002*, vol. 2523, B. S. Kaliski, çetin K. Koç, and C. Paar, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 2–12.

[89] C. Aumüller, C. Bier, W. Fischer, P. Hofreiter, and J.-P. Seifert, "Fault Attacks on RSA with CRT, Concrete Results and Practical Countermeasures.pdf," *Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA*, vol. 2523, pp. 260–275, 2002, doi: 10.1007/3-540-36400-5_20.

[90] M. Agoyan, J.-M. Dutertre, D. Naccache, B. Robisson, and A. Tria, "When Clocks Fail: On Critical Paths and Clock Faults," in *Smart Card Research and Advanced Application*, vol. 6035, D. Gollmann, J.-L. Lanet, and J. Iguchi-Cartigny, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 182–193.

[91] Y. Kim *et al.*, "Flipping bits in memory without accessing them: An experimental study of DRAM disturbance errors," in *2014 ACM/IEEE 41st International Symposium on Computer Architecture (ISCA)*, Minneapolis, MN, USA, Jun. 2014, pp. 361–372, doi: 10.1109/ISCA.2014.6853210.

[92] M. Madau *et al.*, "The Impact of Pulsed Electromagnetic Fault Injection on True Random Number Generators," in *2018 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*, Amsterdam, Netherlands, Sep. 2018, pp. 43–48, doi: 10.1109/FDTC.2018.00015.

[93] "SECNUM," [Online]. Available: http://web-pcm.cnfm.fr/secnum/.

[94] M. Renauld and F.-X. Standaert, "Algebraic Side-Channel Attacks," in *Information Security and Cryptology*, vol. 6151, F. Bao, M. Yung, D. Lin, and J. Jing, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 393–410.

[95] N. Moro, A. Dehbaoui, K. Heydemann, B. Robisson, and E. Encrenaz, "Electromagnetic fault injection: towards a fault model on a 32-bit microcontroller," *2013 Workshop on Fault Diagnosis and Tolerance in Cryptography*, pp. 77–88, Aug. 2013, doi: 10.1109/FDTC.2013.9.

[96] P. Kocher and S. Francisco, "Differential Power Analysis," *Advances in Cryptology --- CRYPTO' 99*, pp. 388–397, 1999, doi: 10.1007/3-540-48405-1_25.

[97] C. Canovas and J. Clediere, "What do S-boxes Say in Differential Side Channel Attacks?," *Cryptology ePrint Archive*, p. 14, 2005.

[98] E. Brier, C. Clavier, and F. Olivier, "Correlation Power Analysis with a Leakage Model," in *Cryptographic Hardware and Embedded Systems - CHES 2004*, vol. 3156, M. Joye and J.-J. Quisquater, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 16–29.

[99] Langer, "Sonde EMV Langer," https://www.langer-emv.de/en/product/rf-passive-30-mhz-3-ghz/35/rf-u-2-5-2-h-field-probe-30-mhz-up-to-3-ghz/11.

[100] Femto, "Femto - Amplificateur Haute Vitesse," [Online]. Available: https://www.femto.de/images/pdf-dokumente/de-hsa-x-2-40.pdf.

[101] Teledyne LeCroy, "LECROY-WAVEPRO 735ZI-Datasheet," [Online]. Available: http://www.testequipmenthq.com/datasheets/LECROY-WAVEPRO%20735ZI-Datasheet.pdf.

[102] MathWorks Inc, "MATLAB," [Online]. Available: https://www.mathworks.com.

[103] D. E. Kouicem, A. Bouabdallah, and H. Lakhlef, "An Efficient Architecture for Trust Management in IoE Based Systems of Systems," in *2018 13th Annual Conference on System of Systems Engineering (SoSE)*, Paris, Jun. 2018, pp. 138–143, doi: 10.1109/SYSOSE.2018.8428732.

[104] H. Xu *et al.*, "Trust-Based Probabilistic Broadcast Scheme for Mobile Ad Hoc Networks," *IEEE Access*, vol. 8, pp. 21380–21392, 2020, doi: 10.1109/ACCESS.2020.2969447.

[105] International Telecommunication Union (ITU), "Overview of trust provisioning in

information and communication technology infrastructures and services." 2017.

[106] Z. Gao *et al.*, "A Credible and Lightweight Multidimensional Trust Evaluation Mechanism for Service-Oriented IoT Edge Computing Environment," in *2019 IEEE International Congress on Internet of Things (ICIOT)*, Milan, Italy, Jul. 2019, pp. 156–164, doi: 10.1109/ICIOT.2019.00035.

[107] E. K. Wang, C.-M. Chen, D. Zhao, W. H. Ip, and K. L. Yung, "A Dynamic Trust Model in Internet of Things," *Soft Comput*, vol. 24, no. 8, pp. 5773–5782, Apr. 2020, doi: 10.1007/s00500-019-04319-2.

[108] F. Bao and I.-R. Chen, "Dynamic trust management for internet of things applications," in *Proceedings of the 2012 international workshop on Self-aware internet of things - Self-IoT '12*, San Jose, California, USA, 2012, p. 1, doi: 10.1145/2378023.2378025.

[109] J. Liang, M. Zhang, and V. C. M. Leung, "A Reliable Trust Computing Mechanism Based on Multisource Feedback and Fog Computing in Social Sensor Cloud," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5481–5490, Jun. 2020, doi: 10.1109/JIOT.2020.2981005.

[110] B. Wang, M. Li, X. Jin, and C. Guo, "A Reliable IoT Edge Computing Trust Management Mechanism for Smart Cities," *IEEE Access*, vol. 8, pp. 46373–46399, 2020, doi: 10.1109/ACCESS.2020.2979022.

[111] N. Djedjig, D. Tandjaoui, F. Medjek, and I. Romdhani, "Trust-aware and cooperative routing protocol for IoT security," *Journal of Information Security and Applications*, vol. 52, p. 102467, Jun. 2020, doi: 10.1016/j.jisa.2020.102467.

[112] Y. Hussain *et al.*, "Context-Aware Trust and Reputation Model for Fog-Based IoT," *IEEE Access*, vol. 8, pp. 31622–31632, 2020, doi: 10.1109/ACCESS.2020.2972968.

[113] B. Shala, U. Trick, A. Lehmann, B. Ghita, and S. Shiaeles, "Blockchain and Trust for Secure, End-User-Based and Decentralized IoT Service Provision," *IEEE Access*, vol. 8, pp. 119961–119979, 2020, doi: 10.1109/ACCESS.2020.3005541.

[114] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," p. 9, 2008.

[115] H. Tschofenig and M. Pegourie-Gonnard, "Performance of State-of-the-Art Cryptography on ARM-based Microprocessors," *Proc. NIST Lightweight Cryptography Workshop*, p. 40, 2015.

[116] I. Makhdoom, M. Abolhasan, H. Abbas, and W. Ni, "Blockchain's adoption in IoT: The challenges, and a way forward," *Journal of Network and Computer Applications*, vol. 125, pp. 251–279, Jan. 2019, doi: 10.1016/j.jnca.2018.10.019.

[117] V. Buterin, "A Next Generation Smart Contract & Decentralized Application Platform," pp. 1–36, 2014.

[118] S. Popov, "Tangle.pdf." 2016.

[119] C. LeMahieu, "Nano: A Feeless Distributed Cryptocurrency Network," *Whitepaper*, p. 8.

[120] Hyperledger White Paper Working Group, "Hyperledger - Introduction to Hyperledger,"

2018, [Online]. Available: https://www.hyperledger.org/wp-content/uploads/2018/07/HL_Whitepaper_IntroductiontoHyperledger.pdf.

[121] D. Voell and P. Mylund Nielsen, "Quorum Whitepaper v0.2," [Online]. Available: https://github.com/ConsenSys/quorum-docs/blob/master/Quorum%20Whitepaper%20v0.2.pdf.

[122] M. A. Ferrag, M. Derdour, M. Mukherjee, A. Derhab, L. Maglaras, and H. Janicke, "Blockchain Technologies for the Internet of Things: Research Issues and Challenges," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2188–2204, Apr. 2019, doi: 10.1109/JIOT.2018.2882794.

[123] A. Reyna, C. Martín, J. Chen, E. Soler, and M. Díaz, "On blockchain and its integration with IoT. Challenges and opportunities," *Future Generation Computer Systems*, vol. 88, pp. 173–190, Nov. 2018, doi: 10.1016/j.future.2018.05.046.

[124] Vitalik Buterin, "On Public and Private Blockchains," [Online]. Available: https://blog.ethereum.org/2015/08/07/on-public-and-private-blockchains/.

[125] M. Hearn and R. G. Brown, "Corda: A distributed ledger," *Corda Technical White Paper*, p. 73, 2016.

[126] XinFin Organization, "The XDC Protocol, Technical Whitepaper," 2017, [Online]. Available: https://www.xinfin.io/pdf/whitepaper/The%20XDC%20Protocol%20-%20Tech%20Whitepaper%20by%20Xinfin%20Organization%20V1.0.pdf.

[127] Binance Academy, "What are Nodes ?" [Online]. Available: https://academy.binance.com/blockchain/what-are-nodes.

[128] BLOG X-TRIX, "UTXO & Account," [Online]. Available: https://medium.com/x-trix/comprendre-utxo-account-model-ethereum-bitcoin-5dcec393656b.

[129] F. Bobée, "Smart Contract," *Bitconseil*. [Online]. Available: https://bitconseil.fr/smart-contract-contrat-intelligent/.

[130] IOTA Foundation, "IOTA," [Online]. Available: https://www.iota.org.

[131] M. S. Ali, M. Vecchio, M. Pincheira, K. Dolui, F. Antonelli, and M. H. Rehmani, "Applications of Blockchains in the Internet of Things: A Comprehensive Survey," *IEEE Commun. Surv. Tutorials*, vol. 21, no. 2, pp. 1676–1717, 2019, doi: 10.1109/COMST.2018.2886932.

[132] M. Salimitari and M. Chatterjee, "A Survey on Consensus Protocols in Blockchain for IoT Networks," *arXiv:1809.05613 [cs]*, Jun. 2019, Accessed: Aug. 26, 2020. [Online]. Available: http://arxiv.org/abs/1809.05613.

[133] Q. Bramas, "The Stability and the Security of the Tangle," p. 17, 2018.

[134] V. Pureswaran, S. Panikkar, S. Nair, and P. Brody, "Empowering the edge," *IBM Institute for Business Value*, p. 24, 2015.

[135] S. Gan, "An IoT simulator in NS3 and a key-based authentication architecture for IoT devices

using blockchain," Master Thesis, Department Of Computer Science And Engineering Indian Institute Of Technology Kanpur, 2017.

[136] K. Biswas and V. Muthukkumarasamy, "Securing Smart Cities Using Blockchain Technology," in *2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, Sydney, Australia, Dec. 2016, pp. 1392–1393, doi: 10.1109/HPCC-SmartCity-DSS.2016.0198.

[137] D. Han, H. Kim, and J. Jang, "Blockchain based Smart Door Lock System," *2017 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 1165–1167, 2017, doi: 10.1109/ICTC.2017.8190886.

[138] M. Samaniego and R. Deters, "Internet of Smart Things - IoST: Using Blockchain and CLIPS to Make Things Autonomous," in *2017 IEEE International Conference on Cognitive Computing (ICCC)*, Honolulu, HI, USA, Jun. 2017, pp. 9–16, doi: 10.1109/IEEE.ICCC.2017.9.

[139] G. Prisco, "Slock. It to introduce smart locks linked to smart ethereum contracts, decentralize the sharing economy," https://bitcoinmagazine.com/articles/slock-it-to-introduce-smart-locks-linked-to-smart-ethereum-contracts-decentralize-the-sharing-economy-1446746719.

[140] Chronicled , "Chronicled," [Online]. Available: https://www.chronicled.com.

[141] A. Dorri, S. S. Kanhere, and R. Jurdak, "Blockchain in Internet of Things: Challenges and Solutions," p. 13, 2016.

[142] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for IoT security and privacy: The case study of a smart home," in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, Kona, HI, Mar. 2017, pp. 618–623, doi: 10.1109/PERCOMW.2017.7917634.

[143] T. Bocek, B. B. Rodrigues, T. Strasser, and B. Stiller, "Blockchains everywhere - a use-case of blockchains in the pharma supply-chain," in *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, Lisbon, Portugal, May 2017, pp. 772–777, doi: 10.23919/INM.2017.7987376.

[144] Z. Shae and J. J. P. Tsai, "On the Design of a Blockchain Platform for Clinical Trial and Precision Medicine," in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, Atlanta, GA, USA, Jun. 2017, pp. 1972–1980, doi: 10.1109/ICDCS.2017.61.

[145] M. A. Salahuddin, A. Al-Fuqaha, M. Guizani, K. Shuaib, and F. Sallabi, "Softwarization of Internet of Things Infrastructure for Secure and Smart Healthcare," *Computer*, vol. 50, no. 7, pp. 74–79, 2017, doi: 10.1109/MC.2017.195.

[146] M. Siddiqi, S. T. All, and V. Sivaraman, "Secure lightweight context-driven data logging for bodyworn sensing devices," in *2017 5th International Symposium on Digital Forensic and*

Security (ISDFS), Tirgu Mures, Romania, Apr. 2017, pp. 1–6, doi: 10.1109/ISDFS.2017.7916500.

[147] C. Esposito, A. De Santis, G. Tortora, H. Chang, and K.-K. R. Choo, "Blockchain: A Panacea for Healthcare Cloud-Based Data Security and Privacy?," *IEEE Cloud Comput.*, vol. 5, no. 1, pp. 31–37, Jan. 2018, doi: 10.1109/MCC.2018.011791712.

[148] R. Guo, H. Shi, Q. Zhao, and D. Zheng, "Secure Attribute-Based Signature Scheme With Multiple Authorities for Blockchain in Electronic Health Records Systems," *IEEE Access*, vol. 6, pp. 11676–11686, 2018, doi: 10.1109/ACCESS.2018.2801266.

[149] X. Liang, J. Zhao, S. Shetty, J. Liu, and D. Li, "Integrating blockchain for data sharing and collaboration in mobile healthcare applications," in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Montreal, QC, Oct. 2017, pp. 1–5, doi: 10.1109/PIMRC.2017.8292361.

[150] A. Shahnaz, U. Qamar, and A. Khalid, "Using Blockchain for Electronic Health Records," *IEEE Access*, vol. 7, pp. 147782–147795, 2019, doi: 10.1109/ACCESS.2019.2946373.

[151] F. Tang, S. Ma, Y. Xiang, and C. Lin, "An Efficient Authentication Scheme for Blockchain-Based Electronic Health Records," *IEEE Access*, vol. 7, pp. 41678–41689, 2019, doi: 10.1109/ACCESS.2019.2904300.

[152] T. Ahram, A. Sargolzaei, S. Sargolzaei, J. Daniels, and B. Amaba, "Blockchain technology innovations," in *2017 IEEE Technology & Engineering Management Conference (TEMSCON)*, Santa Clara, CA, USA, Jun. 2017, pp. 137–141, doi: 10.1109/TEMSCON.2017.7998367.

[153] N. Kshetri, "Can Blockchain Strengthen the Internet of Things?," *IT Prof.*, vol. 19, no. 4, pp. 68–72, 2017, doi: 10.1109/MITP.2017.3051335.

[154] Modum, "Modum," [Online]. Available: https://modum.io.

[155] J. Huang, L. Kong, G. Chen, M.-Y. Wu, X. Liu, and P. Zeng, "Towards Secure Industrial IoT: Blockchain System With Credit-Based Consensus Mechanism," *IEEE Trans. Ind. Inf.*, vol. 15, no. 6, pp. 3680–3689, Jun. 2019, doi: 10.1109/TII.2019.2903342.

[156] G. Wang, Z. Shi, M. Nixon, and S. Han, "ChainSplitter: Towards Blockchain-Based Industrial IoT Architecture for Supporting Hierarchical Storage," in *2019 IEEE International Conference on Blockchain (Blockchain)*, Atlanta, GA, USA, Jul. 2019, pp. 166–175, doi: 10.1109/Blockchain.2019.00030.

[157] Y. Xu, J. Ren, G. Wang, C. Zhang, J. Yang, and Y. Zhang, "A Blockchain-Based Nonrepudiation Network Computing Service Scheme for Industrial IoT," *IEEE Trans. Ind. Inf.*, vol. 15, no. 6, pp. 3632–3641, Jun. 2019, doi: 10.1109/TII.2019.2897133.

[158] T. Alladi, V. Chamola, R. M. Parizi, and K.-K. R. Choo, "Blockchain Applications for Industry 4.0 and Industrial IoT: A Review," *IEEE Access*, vol. 7, pp. 176935–176951, 2019, doi: 10.1109/ACCESS.2019.2956748.

[159] T. M. Fernandez-Carames and P. Fraga-Lamas, "A Review on the Application of Blockchain to the Next Generation of Cybersecure Industry 4.0 Smart Factories," *IEEE Access*, vol. 7, pp. 45201–45218, 2019, doi: 10.1109/ACCESS.2019.2908780.

[160] U. Bodkhe *et al.*, "Blockchain for Industry 4.0: A Comprehensive Review," *IEEE Access*, vol. 8, pp. 79764–79800, 2020, doi: 10.1109/ACCESS.2020.2988579.

[161] A. Lei, H. Cruickshank, Y. Cao, P. Asuquo, C. P. A. Ogah, and Z. Sun, "Blockchain-Based Dynamic Key Management for Heterogeneous Intelligent Transportation Systems," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1832–1843, Dec. 2017, doi: 10.1109/JIOT.2017.2740569.

[162] X. Huang, C. Xu, P. Wang, and H. Liu, "LNSC: A Security Model for Electric Vehicle and Charging Pile Management Based on Blockchain Ecosystem," *IEEE Access*, vol. 6, pp. 13565–13574, 2018, doi: 10.1109/ACCESS.2018.2812176.

[163] A. Dorri, M. Steger, S. S. Kanhere, and R. Jurdak, "BlockChain: A Distributed Solution to Automotive Security and Privacy," *IEEE Commun. Mag.*, vol. 55, no. 12, pp. 119–125, Dec. 2017, doi: 10.1109/MCOM.2017.1700879.

[164] J. Kang, R. Yu, X. Huang, S. Maharjan, Y. Zhang, and E. Hossain, "Enabling Localized Peer-to-Peer Electricity Trading Among Plug-in Hybrid Electric Vehicles Using Consortium Blockchains," *IEEE Trans. Ind. Inf.*, vol. 13, no. 6, pp. 3154–3164, Dec. 2017, doi: 10.1109/TII.2017.2709784.

[165] L. Li *et al.*, "CreditCoin: A Privacy-Preserving Blockchain-Based Incentive Announcement Network for Communications of Smart Vehicles," *IEEE Trans. Intell. Transport. Syst.*, vol. 19, no. 7, pp. 2204–2220, Jul. 2018, doi: 10.1109/TITS.2017.2777990.

[166] Z. Yang, K. Zheng, K. Yang, and V. C. M. Leung, "A blockchain-based reputation system for data credibility assessment in vehicular networks," in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Montreal, QC, Oct. 2017, pp. 1–5, doi: 10.1109/PIMRC.2017.8292724.

[167] B. Leiding, P. Memarmoshrefi, and D. Hogrefe, "Self-managed and blockchain-based vehicular ad-hoc networks," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, Heidelberg Germany, Sep. 2016, pp. 137–140, doi: 10.1145/2968219.2971409.

[168] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. M. Leung, "Blockchain-Based Decentralized Trust Management in Vehicular Networks," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1495–1505, Apr. 2019, doi: 10.1109/JIOT.2018.2836144.

[169] H. Tan and I. Chung, "Secure Authentication and Key Management With Blockchain in VANETs," *IEEE Access*, vol. 8, pp. 2482–2498, 2020, doi: 10.1109/ACCESS.2019.2962387.

[170] D. Zheng, C. Jing, R. Guo, S. Gao, and L. Wang, "A Traceable Blockchain-Based Access Authentication System With Privacy Preservation in VANETs," *IEEE Access*, vol. 7, pp. 117716–117726, 2019, doi: 10.1109/ACCESS.2019.2936575.

[171] Y. Zhang and J. Wen, "An IoT electric business model based on the protocol of bitcoin," in *2015 18th International Conference on Intelligence in Next Generation Networks*, Paris, France, 2015, pp. 184–191, doi: 10.1109/ICIN.2015.7073830.

[172] Y. R. Kafle, K. Mahmud, S. Morsalin, and G. E. Town, "Towards an internet of energy," in *2016 IEEE International Conference on Power System Technology (POWERCON)*, Wollongong, Australia, Sep. 2016, pp. 1–6, doi: 10.1109/POWERCON.2016.7754036.

[173] Ó. Blanco-Novoa, T. Fernández-Caramés, P. Fraga-Lamas, and L. Castedo, "An Electricity Price-Aware Open-Source Smart Socket for the Internet of Energy," *Sensors*, vol. 17, no. 3, p. 643, Mar. 2017, doi: 10.3390/s17030643.

[174] T. M. Fernández-Caramés, "An Intelligent Power Outlet System for the Smart Home of the Internet of Things," *International Journal of Distributed Sensor Networks*, vol. 11, no. 11, p. 214805, Nov. 2015, doi: 10.1155/2015/214805.

[175] J. Gao *et al.*, "GridMonitoring: Secured Sovereign Blockchain Based Monitoring on Smart Grid," *IEEE Access*, vol. 6, pp. 9917–9925, 2018, doi: 10.1109/ACCESS.2018.2806303.

[176] G. Liang, S. R. Weller, F. Luo, J. Zhao, and Z. Y. Dong, "Distributed Blockchain-Based Data Protection Framework for Modern Power Systems Against Cyber Attacks," *IEEE Trans. Smart Grid*, vol. 10, no. 3, pp. 3162–3173, May 2019, doi: 10.1109/TSG.2018.2819663.

[177] Z. Li, J. Kang, R. Yu, D. Ye, Q. Deng, and Y. Zhang, "Consortium Blockchain for Secure Energy Trading in Industrial Internet of Things," *IEEE Trans. Ind. Inf.*, pp. 1–1, 2017, doi: 10.1109/TII.2017.2786307.

[178] N. Z. Aitzhan and D. Svetinovic, "Security and Privacy in Decentralized Energy Trading Through Multi-Signatures, Blockchain and Anonymous Messaging Streams," *IEEE Trans. Dependable and Secure Comput.*, vol. 15, no. 5, pp. 840–852, Sep. 2018, doi: 10.1109/TDSC.2016.2616861.

[179] LO3Energy, "LO3Energy," [Online]. Available: https://lo3energy.com.

[180] A. A. G. Agung and R. Handayani, "Blockchain for smart grid," *Journal of King Saud University - Computer and Information Sciences*, p. S1319157819309000, Jan. 2020, doi: 10.1016/j.jksuci.2020.01.002.

[181] Feng Tian, "An agri-food supply chain traceability system for China based on RFID & blockchain technology," in *2016 13th International Conference on Service Systems and Service Management (ICSSSM)*, Kunming, China, Jun. 2016, pp. 1–6, doi: 10.1109/ICSSSM.2016.7538424.

[182] M. Shyamala Devi, R. Suguna, A. S. Joshi, and R. A. Bagate, "Design of IoT Blockchain Based Smart Agriculture for Enlightening Safety and Security," in *Emerging Technologies in Computer Engineering: Microservices in Big Data Analytics*, vol. 985, A. K. Somani, S. Ramakrishna, A. Chaudhary, C. Choudhary, and B. Agarwal, Eds. Singapore: Springer Singapore, 2019, pp. 7–19.

[183] P. Chun-Ting, L. Meng-Ju, H. Nen-Fu, L. Jhong-Ting, and S. Jia-Jung, "Agriculture Blockchain Service Platform for Farm-to-Fork Traceability with IoT Sensors," in *2020 International Conference on Information Networking (ICOIN)*, Barcelona, Spain, Jan. 2020, pp. 158–163, doi: 10.1109/ICOIN48656.2020.9016535.

[184] M. A. Ferrag, L. Shu, X. Yang, A. Derhab, and L. Maglaras, "Security and Privacy for Green IoT-Based Agriculture: Review, Blockchain Solutions, and Challenges," *IEEE Access*, vol. 8, pp. 32031–32053, 2020, doi: 10.1109/ACCESS.2020.2973178.

[185] Storj Labs, Inc., "Storj, A Decentralized Cloud Storage Network Framework," 2018, [Online]. Available: https://github.com/storj/whitepaper.

[186] G. Ateniese, M. T. Goodrich, V. Lekakis, C. Papamanthou, E. Paraskevas, and R. Tamassia, "Accountable Storage," in *Applied Cryptography and Network Security*, vol. 10355, D. Gollmann, A. Miyaji, and H. Kikuchi, Eds. Cham: Springer International Publishing, 2017, pp. 623–644.

[187] M. A. Ferrag, L. Maglaras, A. Argyriou, D. Kosmanos, and H. Janicke, "Security for 4G and 5G cellular networks: A survey of existing authentication and privacy-preserving schemes," *Journal of Network and Computer Applications*, vol. 101, pp. 55–82, Jan. 2018, doi: 10.1016/j.jnca.2017.10.017.

[188] K. Fan, Y. Ren, Y. Wang, H. Li, and Y. Yang, "Blockchain-based efficient privacy preserving and data sharing scheme of content-centric network in 5G," *IET Communications*, vol. 12, no. 5, pp. 527–532, Mar. 2018, doi: 10.1049/iet-com.2017.0619.

[189] Raspberry Pi Foundation, "Raspberry Pi3 Model B+," [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/.

[190] Arduino, "Arduino Intel Edison," [Online]. Available: https://www.arduino.cc/en/guide/IntelEdison.

[191] T. M. Fernandez-Carames and P. Fraga-Lamas, "A Review on the Use of Blockchain for the Internet of Things," *IEEE Access*, vol. 6, pp. 32979–33001, 2018, doi: 10.1109/ACCESS.2018.2842685.

[192] Bitinfocharts.com, "bitinfocharts," [Online]. Available: https://bitinfocharts.com.

[193] Nano Crawler, "Taille Nano," [Online]. Available: https://nanocrawler.cc/status.

[194] GitBook, "Taille IOTA," [Online]. Available: https://domschiener.gitbooks.io/iota-guide/content/chapter1/transactions-and-bundles.html.

[195] Digiconomist.net, "Consommation Bitcoin," [Online]. Available: https://digiconomist.net/bitcoin-energy-consumption/.

[196] Digiconomist.net, "Consommation Ethereum," [Online]. Available: https://digiconomist.net/ethereum-energy-consumption.

[197] Isnanogreenyet.com, "Consommation Nano," [Online]. Available:

https://isnanogreenyet.com.

[198] A. Abbaszadeh Sori, "Consommation IOTA," https://medium.com/@a.abbaszadeh.s/measuring-iota-pow-s-energy-consumption-and-comparing-with-other-payment-systems-413f4de50274.

[199] D. Frumkin, "Débit des blockchains," [Online]. Available: https://www.investinblockchain.com/transactions-per-second-and-consensus-mechanisms-of-the-top-50-cryptocurrencies/.

[200] H. Sun, S. Hua, E. Zhou, B. Pi, J. Sun, and K. Yamashita, "Using Ethereum Blockchain in Internet of Things: A Solution for Electric Vehicle Battery Refueling," in *Blockchain – ICBC 2018*, vol. 10974, S. Chen, H. Wang, and L.-J. Zhang, Eds. Cham: Springer International Publishing, 2018, pp. 3–17.

[201] A. Elsts, E. Mitskas, and G. Oikonomou, "Distributed Ledger Technology and the Internet of Things: A Feasibility Study," pp. 7–12, 2018, doi: 10.1145/3282278.3282280.

[202] Nano Foundation, "Nano Node," [Online]. Available: https://docs.nano.org/running-a-node/node-setup/.

[203] Raspberry Pi Foundation, "Raspberry Pi 4," [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-4-model-b/.

[204] J.-P. Aumasson, S. Neves, Z. Wilcox-O'Hearn, and C. Winnerlein, "BLAKE2: Simpler, Smaller, Fast as MD5," in *Applied Cryptography and Network Security*, vol. 7954, M. Jacobson, M. Locasto, P. Mohassel, and R. Safavi-Naini, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 119–135.

[205] Nano Crawler, "Nombre de noeuds Nano," [Online]. Available: https://nanocrawler.cc/network.

[206] IOTA Foundation, "Noeud IOTA," [Online]. Available: https://docs.iota.org/docs/getting-started/1.1/running-nodes/running-a-node.

[207] IOTA Foundation, "IOTA Guide," [Online]. Available: https://www.iota-guide.com/f-a-q/.

[208] NXP, "NXP JN5189," [Online]. Available: https://www.nxp.com/products/wireless/thread/jn5189-88-t-high-performance-and-ultra-low-power-mcus-for-zigbee-and-thread-with-built-in-nfc-option:JN5189_88_T.

[209] M. Goldstein, "Big Data for IoT: Analytics from Descriptive to Predictive to Prescriptive," *Phoenix Data Conference*, 2017, [Online]. Available: https://www.slideshare.net/markgirc/phoenix-data-conference-big-data-analytics-for-iot-11417.

[210] J. D. Bruce, "The Mini-Blockchain Scheme," p. 13, 2014.

[211] Cryptoast, "Vitesse et aspect pratique des crypto-monnaies par rapport aux systèmes de paiement traditionnels," 2020. [Online]. Available: https://cryptoast.fr/cryptomonnaies-

face-reseaux-traditionnels-vitesse/.

[212] K. Qin and A. Gervais, "An overview of blockchain scalability, interoperability and sustainability," p. 15, 2020.

[213] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxena, "A Secure Sharding Protocol For Open Blockchains," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, Vienna Austria, Oct. 2016, pp. 17–30, doi: 10.1145/2976749.2978389.

[214] E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta, and B. Ford, "OmniLedger: A Secure, Scale-Out, Decentralized Ledger via Sharding," in *2018 IEEE Symposium on Security and Privacy (SP)*, San Francisco, CA, May 2018, pp. 583–598, doi: 10.1109/SP.2018.000-5.

[215] F. Bobée, "Qu'est-ce qu'une side chain ?" *Bitconseil*, 2019. [Online]. Available: https://bitconseil.fr/side-chain/.

[216] BitcoinWiki, "Sidechain & blockchain," [Online]. Available: https://en.bitcoinwiki.org/wiki/Sidechain.

[217] A. Back *et al.*, "Enabling Blockchain Innovations with Pegged Sidechains," p. 25, 2014.

[218] J. Poon and T. Dryja, "The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments," p. 59, 2016.

[219] F. Tschorsch and B. Scheuermann, "Bitcoin and Beyond: A Technical Survey on Decentralized Digital Currencies," *IEEE Commun. Surv. Tutorials*, vol. 18, no. 3, pp. 2084–2123, 2016, doi: 10.1109/COMST.2016.2535718.

[220] Bitnodes.io, "Nombre de noeuds Bitcoin," [Online]. Available: https://bitnodes.io.

[221] Ethernodes.org, "Nombre de noeuds Ethereum," [Online]. Available: https://www.ethernodes.org/?synced=0.

[222] S. Goldwasser, S. Micali, and C. Rackoff, "The Knowledge Complexity of Interactive Proof-System," *Proceedings of the 17th Annual ACM Symposium on Theory of Computing*, pp. 291–304, 1985.

[223] M. Blum, A. De Santis, S. Micali, and G. Persiano, "Non interactive Zero-Knowkedge.pdf," 1991.

[224] E. Fujisaki and K. Suzuki, "Traceable Ring Signature," *Public Key Cryptography*, pp. 181–200, 2007.

[225] Q. H. Dang, "Secure Hash Standard," National Institute of Standards and Technology, NIST FIPS 180-4, Jul. 2015. doi: 10.6028/NIST.FIPS.180-4.

[226] Vasan Subramanian, "Why Bitcoin Payments Take 10 Minutes," [Online]. Available: https://medium.com/unblockchain/why-bitcoin-payments-take-10-minutes-c6f37f424b4f.

[227] M. Bedford Taylor, "The Evolution of Bitcoin Hardware," *Computer*, vol. 50, no. 9, pp. 58–

66, 2017, doi: 10.1109/MC.2017.3571056.

[228] D. G. Wood, "Ethereum: A Secure Decentralised Generalised Transaction Ledger," *Ethereum Project Yellow Paper*, p. 39, 2014.

[229] Colin LeMahieu, "Nano PoW v2," [Online]. Available: https://medium.com/nanocurrency/nano-pow-the-details-ba22a9092d6f.

[230] Ethereum Foundation, "Proof of Stake (Ethereum)," [Online]. Available: https://eth.wiki/concepts/proof-of-stake-faqs.

[231] PeerCoin Foundation, "PeerCoin," [Online]. Available: https://www.peercoin.net.

[232] Nxt Community, "Nxt Whitepaper," 2014, [Online]. Available: https://whitepaper.io/document/62/nxt-whitepaper

[233] BitShares Blockchain Foundation Revision, "Delegated Proof of Stake," [Online]. Available: https://how.bitshares.works/en/master/technology/dpos.html.

[234] Lisk Foundation, "Lisk," [Online]. Available: https://lisk.io/documentation/lisk-protocol/consensus-algorithm.html.

[235] ARK.io, "ARK Ecosystem Whitepaper," 2019, [Online]. Available: https://ark.io/Whitepaper.pdf

[236] Vardan Sevan, "Nano Examples of Consensus Protocols," [Online]. Available: https://medium.com/@vardan.sevan/nano-examples-of-consensus-protocols-36134f3548d8.

[237] NEM Group Ltd, "NEM - Technical Reference," 2018, [Online]. Available: https://nemplatform.com/wp-content/uploads/2020/05/NEM_techRef.pdf

[238] Linux Foundation, "Proof of Elapsed Time," [Online]. Available: https://sawtooth.hyperledger.org/docs/core/releases/latest/architecture/poet.html?highlight=poet.

[239] Intel Corporation, "Intel SGX," [Online]. Available: https://software.intel.com/content/www/us/en/develop/topics/software-guard-extensions.html.

[240] Arm Limited, "ARM TrustZone," [Online]. Available: https://developer.arm.com/ip-products/security-ip/trustzone.

[241] S.V. Schaik, M. Minkin, A. Kwong, D Genkin, Y Yarom, "Intel SGX Attacks." [Online]. Available: https://sgaxe.com.

[242] IOTA Foundation, "IOTA Coordicide," [Online]. Available: https://blog.iota.org/coordinator-part-1-the-path-to-coordicide-ee4148a8db08.

[243] S. Popov *et al.*, "IOTA Coordicide Whitepaper," 2020.

[244] M. Castro and B. Liskov, "Practical Byzantine Fault Tolerance," *3rd OSDI*, p. 14, 1999.

[245] L. Lamport, R. Shostak, and M. Pease, "The Byzantine Generals Problem," *ACM Transactions on Programming Languages and Systems*, vol. 4, no. 3, p. 20.

[246] Neo Team, "NEO, Delegated Byzantine Fault Tolerance," [Online]. Available: https://docs.neo.org/docs/en-us/basic/technology/dbft.html.

[247] J. Kwon, "Tendermint: Consensus without Mining," p. 11.

[248] Y. Gilad, R. Hemo, S. Micali, G. Vlachos, and N. Zeldovich, "Algorand: Scaling Byzantine Agreements for Cryptocurrencies," in *Proceedings of the 26th Symposium on Operating Systems Principles*, Shanghai China, Oct. 2017, pp. 51–68, doi: 10.1145/3132747.3132757.

[249] S. Micali, M. Rabin, and S. Vadhan, "Verifiable random functions," in *40th Annual Symposium on Foundations of Computer Science (Cat. No.99CB37039)*, New York City, NY, USA, 1999, pp. 120–130, doi: 10.1109/SFFCS.1999.814584.

[250] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog Computing and Its Role in the Internet of Things," *MCC '12: Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pp. 13–16, 2012, doi: doi/10.1145/2342509.2342513.

[251] P. Benoit, L. Dalmasso, G. Patrigeon, T. Gil, F. Bruguier, and L. Torres, "Edge-Computing Perspectives with Reconfigurable Hardware," in *2019 14th International Symposium on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC)*, York, United Kingdom, Jul. 2019, pp. 51–58, doi: 10.1109/ReCoSoC48741.2019.9034961.

[252] TP-Link, "TP-Link Archer C50," [Online]. Available: https://www.tp-link.com/fr/home-networking/wifi-router/archer-c50/.

[253] Cisco, "Cisco Switch Catalyst 1900," [Online]. Available: https://www.cnet.com/products/cisco-catalyst-1900-switch-24-ports-managed/.

[254] K. Skiadopoulos, "Synchronization of data measurements in wireless sensor networks for IoT applications," *Ad Hoc Networks*, vol. 89, pp. 47–57, 2019, doi: doi.org/10.1016/j.adhoc.2019.03.002.

[255] MQTT.org, "MQTT," [Online]. Available: https://mqtt.org.

[256] Object Management Group (OMG), "Data Distribution Service for Real-time Systems," 2007.

[257] J. Laurent, P. Benoit, L. Dalmasso, and T. Gil, "Computing in the Fog with Reconfigurable Gateways," in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, Florence, 2018, pp. 1–4, doi: 10.1109/ISCAS.2018.8351774.

[258] OpenDDS.org, "OpenDDS," [Online]. Available: https://opendds.org.

[259] SQLite.org, "SQLite," [Online]. Available: https://www.sqlite.org/index.html.

[260] Svetlin Nakov, "Elliptic Curve Cryptography (ECC)," [Online]. Available: https://cryptobook.nakov.com/asymmetric-key-ciphers/elliptic-curve-cryptography-ecc.

[261] Sveltin Nakov, "ECDSA," [Online]. Available: https://cryptobook.nakov.com/digital-signatures/ecdsa-sign-verify-messages.

[262] Ken MacKay, "micro-ecc," [Online]. Available: http://kmackay.ca/micro-ecc/.

[263] Grafana Labs, "Grafana," [Online]. Available: https://grafana.com.

[264] Pico Technology, "PicoTech 60 A AC/DC current probe," [Online]. Available: https://www.picotech.com/accessories/bnc-current-probes/60-a-current-probe-bnc.

[265] Nano Foundation, "Nano Difficulty," [Online]. Available: https://docs.nano.org/integration-guides/work-generation/#difficulty-thresholds.

[266] E. Heilman, N. Narula, G. Tanzer, J. Lovejoy, M. Virza, and T. Dryja, "Cryptanalysis of Curl-P and Other Attacks on the IOTA Cryptocurrency," *Proc. IACR Cryptol. ePrint Arch*, p. 344, 2019.

[267] Lucy Hattersely, " Raspberry Pi 3B+ Specs and Benchmarks," [Online]. Available: https://magpi.raspberrypi.org/articles/raspberry-pi-3bplus-specs-benchmarks.

[268] C. Fan, "Performance Analysis and Design of an IoT-Friendly DAG-based Distributed Ledger System," Master Thesis, University of Alberta, 2019.

[269] S. Chang *et al.*, "Third-Round Report of the SHA-3 Cryptographic Hash Algorithm Competition," National Institute of Standards and Technology, Gaithersburg, MD, NIST IR 7896, Nov. 2012. doi: 10.6028/NIST.IR.7896.

[270] Energuide.be, "Consommation appareils électroménagers," [Online]. Available: https://www.energuide.be/fr/questions-reponses/combien-les-appareils-electromenagers-consomment-ils/71/.

[271] H. H. R. Sherazi, G. Piro, L. A. Grieco, and G. Boggia, "When Renewable Energy Meets LoRa: A Feasibility Analysis on Cable-Less Deployments," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 5097–5108, Dec. 2018, doi: 10.1109/JIOT.2018.2839359.

[272] Nano Foundation, "Temps de confirmation Nano," [Online]. Available: https://docs.nano.org/what-is-nano/overview/.

[273] TheTangle.org, "Nombre de noeuds IOTA," [Online]. Available: https://thetangle.org/nodes.

[274] IOTA Foundation, "IOTA Coordinator Monitor," [Online]. Available: http://coordinator.iotawatch.it.

[275] BitcoinWiki, "Bitcoin Block Size," [Online]. Available: https://en.bitcoin.it/wiki/Block_size_limit_controversy.

[276] Ethereum Gas Station, "Ethereum Block Size," [Online]. Available: https://ethgasstation.info/blog/ethereum-block-size/.

[277] Blockchain.com, "Nombre d'adresses Bitcoin (Totale)," [Online]. Available:

https://www.blockchain.com/explorer?utm_campaign=dcomnav_explorer.

[278] Bitinfocharts.com, "Nombre d'adresses Bitcoin (avec Solde)," [Online]. Available: https://bitinfocharts.com/top-100-richest-bitcoin-addresses.html.

[279] Etherscan.io, "Nombre d'adresses Ethereum," [Online]. Available: https://etherscan.io/chart/address.

[280] Nanocrawler.cc, "Nombre d'adresses Nano," [Online]. Available: https://nanocrawler.cc/explorer/accounts/1.

[281] TheTangle.org, "Nombre d'adresses IOTA," [Online]. Available: https://thetangle.org/statistics/tokens-distribution.

[282] STMicroelectronics, "STM32 Nucleo," [Online]. Available: https://www.st.com/en/evaluation-tools/stm32-nucleo-boards.html.

[283] Espressif, "ESP32 Series," [Online]. Available: https://www.espressif.com/en/products/modules/esp32.

[284] S. Malik, V. Dedeoglu, S. S. Kanhere, and R. Jurdak, "TrustChain: Trust Management in Blockchain and IoT Supported Supply Chains," in *2019 IEEE International Conference on Blockchain (Blockchain)*, Atlanta, GA, USA, Jul. 2019, pp. 184–193, doi: 10.1109/Blockchain.2019.00032.

[285] A. Lahbib, K. Toumi, A. Laouiti, A. Laube, and S. Martin, "Blockchain based trust management mechanism for IoT," in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, Marrakesh, Morocco, Apr. 2019, pp. 1–8, doi: 10.1109/WCNC.2019.8885994.

[286] S. Asiri, "A Blockchain-Based IoT Trust Model," Master Thesis, University of Ryerson, 2013.

[287] M. Boussard, S. Papillon, P. Peloso, M. Signorini, and E. Waisbard, "STewARD:SDN and blockchain-based Trust evaluation for Automated Risk management on IoT Devices," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Paris, France, Apr. 2019, pp. 841–846, doi: 10.1109/INFCOMW.2019.8845126.

[288] A. Moinet, B. Darties, and J.-L. Baril, "Blockchain based trust & authentication for decentralized sensor networks," *arXiv:1706.01730 [cs]*, Jun. 2017, Accessed: Jul. 17, 2020. [Online]. Available: http://arxiv.org/abs/1706.01730.

[289] W. Abdelghani, C. A. Zayani, I. Amous, and F. Sèdes, "Trust Management in Social Internet of Things: A Survey," in *Social Media: The Good, the Bad, and the Ugly*, vol. 9844, Y. K. Dwivedi, M. Mäntymäki, M. N. Ravishankar, M. Janssen, M. Clement, E. L. Slade, N. P. Rana, S. Al-Sharhan, and A. C. Simintiras, Eds. Cham: Springer International Publishing, 2016, pp. 430–441.

[290] J. Marouani, PAC (CXP Group), "Acteurs mondiaux les plus performants de l'IoT," *Electroniques*. [Online]. Available: http://www.electroniques.biz/index.php/economie/vie-

de-la-profession/item/63037-un-cabinet-d-etudes-revele-une-liste-des-12-acteurs-mondiaux-les-plus-performants-de-l-iot.

[291] N. B. Truong, H. Lee, B. Askwith, and G. M. Lee, "Toward a Trust Evaluation Mechanism in the Social Internet of Things," *Sensors*, vol. 17, no. 6, p. 1346, Jun. 2017, doi: 10.3390/s17061346.

[292] R. Lacuesta, G. Palacios-Navarro, C. Cetina, L. Peñalver, and J. Lloret, "Internet of Things: Where to be is to trust," *J Wireless Com Network*, vol. 2012, no. 1, p. 203, Dec. 2012, doi: 10.1186/1687-1499-2012-203.

[293] T. Wang, L. Qiu, A. K. Sangaiah, A. Liu, M. Z. A. Bhuiyan, and Y. Ma, "Edge-Computing-Based Trustworthy Data Collection Model in the Internet of Things," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4218–4227, May 2020, doi: 10.1109/JIOT.2020.2966870.

[294] E. Bellini, Y. Iraqi, and E. Damiani, "Blockchain-Based Distributed Trust and Reputation Management Systems: A Survey," *IEEE Access*, vol. 8, pp. 21127–21151, 2020, doi: 10.1109/ACCESS.2020.2969820.

[295] B. Romdhani, "Exploitation de l'hétérogénéité des réseaux de capteurs et d'actionneurs dans la conception des protocoles d'auto-organisation et de routage," PhD Thesis, INSA de Lyon, 2012.

[296] S. Mangard, N. Pramstaller, and E. Oswald, "Successfully Attacking Masked AES Hardware Implementations," in *Cryptographic Hardware and Embedded Systems – CHES 2005*, vol. 3659, J. R. Rao and B. Sunar, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 157–171.

[297] S. Nikova, C. Rechberger, and V. Rijmen, "Threshold Implementations Against Side-Channel Attacks and Glitches," in *Information and Communications Security*, vol. 4307, P. Ning, S. Qing, and N. Li, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 529–545.

[298] Digilent, "Digilent Zybo," [Online]. Available: https://www.xilinx.com/support/university/boards-portfolio/xup-boards/DigilentZYBO.html.

Liste des publications

Loïc Dalmasso, Florent Bruguier, Pascal Benoit : « TrustLib, a Lightweight Trust Procotol for IoT Devices », dans 58th Design Automation Conference (DAC), San Francisco, 2021 *(en cours de soumission)*

Loïc Dalmasso, Florent Bruguier, Achraf Lamlih, Pascal Benoit : « Wallance, an Alternative to Blockchain for IoT », dans IEEE World Forum on Internet of Things 2020 (WF-IOT), Jun 2020, New Orleans, United States. ⟨hal-02893953v2⟩

Loïc Dalmasso, Florent Bruguier, Pascal Benoit, Lionel Torres : « Evaluation of SPN-Based Lightweight Crypto-Ciphers », dans IEEE Access vol. 7, pp. 10559-10567, 2019, doi : 10.1109/ACCESS.2018.2889790. ⟨lirmm-02081085⟩

Florent Bruguier, **Loïc Dalmasso**, Pascal Benoit, Béatrice Pradarelli, Lionel Torres : « AMUSE : un escape game pour la sécurité numériques » dans Les IDEFI : expérimenter, former, pour transformer, Dec 2019, Paris, France. ⟨lirmm-02444068⟩

Florent Bruguier, Emmanuelle Lecointre, Béatrice Pradarelli, **Loïc Dalmasso**, Pascal Benoit, Lionel Torres : « Teaching Hardware Security : Earnings of an Introduction proposed as an Escape Game », dans International Conference on Interactive Collaborative and Blended Learning (ICBL), Nov 2019, Santiago de Cuba, Cuba. ⟨lirmm-02392461⟩

Pascal Benoit, **Loïc Dalmasso**, Guillaume Patrigeon, Thierry Gil, Florent Bruguier, Lionel Torres : « Edge-Computing Perspectives with Reconfigurable Hardware », dans Proceedings of the 14th International Symposium on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC), York, United Kingdom, 2019, pages 51-58, doi : 10.1109/ReCoSoC48741.2019.9034961. ⟨lirmm-02499157⟩

Florent Bruguier, Pascal Benoit, **Loïc Dalmasso**, Béatrice Pradarelli, Lionel Torres : « AMUSE : un escape game pour l'enseignement de la sécurité numérique », dans Journées Pédagogiques du CNFM, Nov 2018, Saint Malo, France. ⟨lirmm-02090700⟩

Johan Laurent, Pascal Benoit, **Loïc Dalmasso**, Thierry Gil : « Computing in the Fog with Reconfigurable Gateways », dans 2018 IEEE International Symposium on Circuits and Systems (ISCAS), Florence, 2018, pp. 1-4, doi : 10.1109/ISCAS.2018.8351774

Communications

Loïc Dalmasso, Florent Bruguier, Pascal Benoit, Lionel Torres : « Fog Computing and Blockchain Technologies for IoT Applications », dans GDR SoC2 Colloque 2019, Paris

Loïc Dalmasso, Florent Bruguier, Pascal Benoit, Lionel Torres : « Investigation of Lightweight Block Ciphers on FPGA », dans GDR SoC2 Colloque 2018, Paris

Valorisations

Loïc Dalmasso, Florent Bruguier, Achraf Lamlih, Pascal Benoit : « Prix Coup de Cœur » attribué à « Wallance, an Alternative to Blockchain for IoT », aux Rencontres Cybersécurité d'Occitanie, 2020

Loïc Dalmasso, Florent Bruguier, Achraf Lamlih, Pascal Benoit : « Wallance, an Alternative to Blockchain for IoT », dans University Booth Demo DATE2020
Vidéo de présentation disponible sur ce lien : *Présentation-Plateforme-DATE2020*

Florent Bruguier, **Loïc Dalmasso** : « Prix Coup de Cœur » attribué à « AMUSE : Un Escape Game pour la Sécurité Numérique », aux Rencontres Cybersécurité d'Occitanie, 2019

Abstract

The Internet of Things (IoT) is a global infrastructure, aiming to connect any systems of various kinds, to generate new services. While this hyperconnectivity offers many opportunities in many areas (*e.g.* health, industry, energy, etc.), its deployment represents a major challenge in terms of security and privacy. The use of cryptography becomes essential, particularly to ensure the confidentiality of data. Nevertheless, classical algorithms such as AES are not designed to be embedded in systems with limited resources, such as connected devices. To address this issue, the scientific community is moving towards so-called "lightweight" cryptography. Despite this term, to what extent is it relevant and in line with the resources of the connected objects ? What level of security can it achieve ?

Through common metrics, the objective of the first axis of this thesis is to determine the gains of lightweight cryptography compared to the classical one, in terms of size, performance and robustness against side channel analysis, recognized as a major attack vector. This evaluation focuses on the AES-128 standard and its close lightweight derivatives PRESENT-80/128 and more recent GIFT-64-128/GIFT-128-128, whose bit-level oriented structure favours hardware implementation. In terms of performances, GIFT-64-128 is positioned as the best alternative, with a reduction in size by almost a factor of 10, for an efficiency gain of up to a factor of 10 compared to the AES-128. In terms of resistance against side channel analysis, CEMA in this thesis, lightweight cryptography is once again proving its worth, with PRESENT-128 achieving an increased robustness by a factor of 3 compared to AES-128.

Although indispensable, confidentiality alone is not enough to create a secure environment. Facing the emergence of cyber attacks, an IoT ecosystem has to integrate security mechanisms to deal with various threats. In addition, its network topology is becoming conducive to decentralization for efficiency reasons. Without central organization, how can the reliability and security of a highly heterogeneous environment be ensured ?

The second part of this thesis proposes a trust protocol, specifically designed for IoT. Inspired by the blockchain concept, several optimizations have been implemented, allowing to reduce the use of resources to a minimum. In an approach centered on network gateways, the results show a reduction in storage space by a factor of 3 000 compared to Bitcoin, as well as a latency divided by 18, for a energy consumption divided by 3 comparared to a LoRa gateway. Finally, an extension of the protocol is proposed to be embedded in objects. The model focuses on the trust assessment in a close environment, that means on entities where direct communication is possible. With less than 400 bytes, the protocol is capable of evaluating the trust of about ten objects, and about thirty with only 1 KB.

Keywords : *Internet of Things, Lightweight cryptography, Side channel analysis, Blockchain, Decentralized network, Trust*

Résumé

L'Internet des Objets (IdO) est une infrastructure mondiale, visant à connecter une multitude de systèmes de diverses natures, dans le but de générer de nouveaux services. Alors que cette hyperconnectivité laisse entrevoir de nombreuses perspectives dans tous les secteurs d'activité (e.g. la santé, l'industrie, l'énergie, etc.), son déploiement constitue un défi majeur en termes de sécurité et de vie privée. L'usage de la cryptographie devient alors primordial, notamment pour assurer la confidentialité des données. Néanmoins, les algorithmes classiques comme l'AES ne sont pas conçus pour être embarqués dans des systèmes à ressources limitées que sont les objets connectés. Pour répondre à cette problématique, la communauté scientifique s'oriente vers la cryptographie dite « légère ». Malgré ce qualificatif, jusqu'à quel point cette dernière est-elle pertinente et en adéquation avec les ressources des objets connectés ? Quel niveau de sécurité permet-elle d'atteindre ?

Au travers de métriques communes, l'objectif du premier axe de cette thèse est de déterminer les gains qu'apporte la cryptographie légère par rapport à son homologue classique, en termes de taille, de performance et de robustesse face aux attaques par canaux auxiliaires, reconnues comme un vecteur d'attaque majeur. Cette évaluation se focalise sur le standard AES-128 et ses proches dérivés légers PRESENT-80/128 et plus récents GIFT-64-128/GIFT-128-128, dont leur structure orientée au niveau bit favorise l'implémentation matérielle. En termes de performances, GIFT-64-128 se positionne comme la meilleure alternative, avec une réduction de taille de près d'un facteur 10, pour un gain en efficacité pouvant atteindre un facteur 10 par rapport à l'AES-128. Concernant la résistance aux attaques par canaux auxiliaires, ici la CEMA, la cryptographie légère se montre une nouvelle fois plus pertinente, avec PRESENT-128 atteignant une robustesse accrue d'un facteur 3 par rapport à l'AES-128.

Bien qu'indispensable, la confidentialité seule n'est pas suffisante pour instaurer un environnement sécurisé. Face à l'émergence des cyberattaques, un écosystème de l'IdO se doit d'intégrer des mécanismes de sécurité pour faire face à diverses menaces. En outre, sa topologie réseau devient propice à la décentralisation pour des questions d'efficacités. Sans organisme central, comment assurer à la fois la fiabilité et la sécurité d'un environnement fortement hétérogène ?

La seconde partie de cette thèse propose un protocole de confiance, spécialement conçu pour l'IdO. Inspiré du concept de la blockchain, plusieurs optimisations ont été mises en place, permettant de réduire l'utilisation des ressources au minimum. Dans une approche centrée sur les passerelles réseau, les résultats démontrent une réduction de l'espace de stockage d'un facteur 3 000 par rapport à Bitcoin, ainsi qu'une latence divisée par 18, pour une consommation inférieure à 3 fois celle d'une passerelle LoRa. Enfin, une extension du protocole est proposée pour être embarquée dans les objets. Le modèle se concentre sur l'évaluation de la confiance dans un environnement proche, se focalisant sur les entités où une communication directe est possible. Avec moins de 400 octets, le protocole est capable d'évaluer la confiance d'une dizaine d'objets, et environ une trentaine avec seulement 1 Ko.

Mots-clés : *Internet des Objets, Cryptographie légère, Attaques par canaux auxiliaires, Blockchain, Réseau décentralisé, Confiance*