République Algérienne Démocratique et Populaire
Ministre de L'Enseignement Supérieur et de la recherche scientifique

Université A/MIRA-Bejaia
Faculté des Sciences Exactes
Département d'Informatique

Mémoire de fin de cycle

En vue de l'obtntion du diplome Master en infomatique

Option : Administration et Sécurité des Réseaux

Théme :

Gestion de clés basé sur puf pour une sécurité meilleur dans l'internet des objets

---

Encadré par : Dr. SADI Mustapha

Réalisé par :
MERABET Yanis
ZEGANE Sara

Soutenu le 01/07/2024 Devant le jury composé de :

Président : Dr. KHENOUS Lachemi
Examinateur : Dr. BENNAI Yani-Athmane

2023/2024

Remerciement

Nous remercions en premier lieu DIEU tout puissant de nous avoir donné la patience, la santé et la volonté pour achever ce travail.

Nos sincères remerciements vont à nos parents, ainsi qu'à nos frères et sœurs, pour leur soutien moral et leurs encouragements constants.

Nous souhaitons exprimer notre profonde gratitude et nos sincères remerciements à notre encadrant, **M. SADI Mustapha**, pour la qualité exceptionnelle de son encadrement, son suivi attentif, sa disponibilité et ses précieux conseils. Sans votre aide, la réalisation de ce mémoire n'aurait pas été possible.
Merci infiniment.

Nous tenons à remercier les membres du jury d'avoir accepté d'évaluer, examiner et apporter leurs précieuses contributions à notre modeste travail.

Nous adressons nos sincères remerciements à tous les professeurs du département informatique, ainsi qu'à toutes les personnes qui, par leurs conseils et leurs critiques, ont guidé nos réflexions et ont accepté de nous rencontrer et de répondre à nos questions durant nos recherches.

Enfin nous remercions tous ceux qui ont contribué de loin ou de près à la réalisation de ce travail.

Dédicace

À mon père

Dans le silence laissé par ton départ, chaque succès que j'atteins est empreint de tristesse, car je ne peux le partager avec toi. Aujourd'hui, plus que jamais, je sens ta présence à mes côtés. Ce projet, qui marque la fin de mes études, est un hommage à toi, pour toutes les leçons de courage et de persévérance que tu m'as inculquées. Chaque battement de cœur me rappelle tes enseignements, que j'applique avec détermination. Je dédie cette réussite à ta mémoire éternelle, espérant que, où que tu sois, tu ressentes toute la fierté et l'amour que je porte pour toi.

À ma mère

Chère maman, symbole de courage, de patience et de tendresse. Je ne saurais te rendre le centième de ce que tu m'as donné. Que Dieu te garde de tout malheur et te donne une longue vie pour te voir fière de ta fille.

À mes sœurs

Chahrazade, Karima, Ouassila, Souhila, Ouarda, Lynda, Sabrina, Je ne saurais jamais assez-vous remercier, mes chères sœurs, pour la joie que vous me procurez, pour votre énergie et votre humour. Merci d'être toujours présentes pour moi.

À la mémoire de mes grands-parents que dieu leurs offre le paradis éternel
À mes tantes et mes oncles et Toute la famille
À mes chers amis

Sara

Dédicace

Nous exprimons notre reconnaissance et nos louanges à Dieu pour nous avoir donné la capacité de mener à bien ce travail simple.

Je dédie ce modeste travail en premier lieu à mes parents, dont le soutien constant et les encouragements ont été essentiels tout au long de mes études.

À toutes les personnes qui ont apporté leur contribution, directe ou indirecte, à la réalisation de ce projet, je vous adresse mes sincères remerciements.

Yanis

Table des matières

Table des figures

Listes des abréviations

AVISS	Automated Validation of Internet Security Protocols and Services
AVISPA	Automated Validation of Internet Security Protocols and Applications
CAS	Challenging-Response Authentication Scheme
CERP-IdO	Cluster des Projets Européens de Recherche sur l'Internet des Objets
CI	Circuit Intégré
CRP	Challenge-Response Pair
DBD	Database Design
EEPROM	Electrically Erasable Programmable Read-Only Memory
FPGA	Field-Programmable Gate Array
HLSPL	High-Level Protocol Specification Language
IBSG	Internet Business Solutions Group
IEEE	Institute of Electrical and Electronics Engineers
IF	Intermediate Format
Intel	Integrated Electronics
IoT	Internet of Things
KDF	Key Derivation Function
LFSR	Linear Feedback Shift Register
LoRa	Long Range
LPWAN	Low-Power Wide-Area Network
LTL	Linear Temporal Logic
MiTM	Man-in-the-Middle
MQTT	Message Queuing Telemetry Transport
M2M	Machine à Machine
NXP	Next eXPerience

OC	Objet Connecté
OFMC	On-the-fly Model-Checker
PUF	Physically Unclonable Function
RFID	Radio-Frequency Identification
ROM	Read-Only Memory
SATMC	The SAT-based Model-Checker
SIEM	Security Information and Event Management
SIM	Simple Interactive Model
SoC	System on a Chip
SPAN	Software for Practical Analysis of Networks
SRAM	Static Random Access Memory
TIC	Technologies de l'Information et de la Communication
T2T	Things to things
Wi Fi	Wireless Fidelity
XOR	EXclusive OR

Introduction générale

L'Internet des objets (IoT) est l'un des sujets les plus brûlants et les plus clivants dans le domaine des technologies de l'information et de la communication (TIC). Cette technologie offre de nouvelles possibilités d'interaction et de communication entre appareils, systèmes et utilisateurs en reliant des objets physiques à Internet. Cette connectivité a le potentiel de transformer notre quotidien en rendant nos environnements plus intelligents et réactifs.

Grâce à l'intégration de capteurs et de capacités de traitement avancées, les objets connectés peuvent collecter, analyser et partager des données en temps réel, favorisant ainsi le développement d'applications innovantes qui améliorent la qualité de vie, optimisent les processus industriels et permettent une gestion plus efficace des ressources. Malgré les défis liés à la confidentialité et à la sécurité des données, les avantages en termes d'efficacité, de durabilité et de confort font de l'IoT un moteur clé de la transformation numérique actuelle.

Les PUFs apparaissent comme une solution prometteuse et novatrice pour la sécurité des appareils IoT. En intégrant ces fonctions dans les dispositifs, chaque appareil devient unique et authentique, réduisant les risques de fraude et de cyberattaques. L'authentification des dispositifs est essentielle pour assurer la sécurité et l'intégrité des réseaux IoT.

Notre objectif dans ce travail est la vérification automatique de certains protocoles d'authentification basés sur ces fonctions pour les appareils IoT en utilisant l'outil de vérification AVISPA, afin de mettre en évidence leur robustesse ou leurs failles potentielles. Par la suite, nous proposons une amélioration pour l'un de ces protocoles lorsque la vérification révèle des faiblesses détectées dans le protocole d'origine.

Notre mémoire est structuré comme suit : Dans le premier chapitre, nous présentons une vue d'ensemble de l'IoT et l'impact qu'il aura sur notre mode de vie.
Le chapitre suivant est consacré aux fonctions physiquement non clonables et à leur rôle dans la sécurité de l'Internet des Objets.
Dans le troisième chapitre, nous offrons un aperçu général de divers travaux proposant des protocoles d'authentification basés sur les PUFs pour les appareils IoT.
Enfin, dans le dernier chapitre, nous introduisons l'outil SPAN/AVISPA et procédons à la vérification de la sécurité de plusieurs protocoles et nous términons une colclusion générale.

Chapitre 1

Internet des objets

1.1 Introduction

L'IoT a évolué au-delà de l'engouement initial pour devenir un catalyseur majeur de l'innovation. Avec des exemples concrets démontrant son impact positif, l'IoT connaît une croissance significative dans divers marchés. Les entreprises qui ont correctement implémenté l'IoT ont observé des résultats qui dépassent leurs attentes, avec 88% rapportant un retour sur investissement positif. Cette tendance souligne l'importance croissante de l'IoT dans notre société moderne, transformant des objets ordinaires en entités intelligentes à travers une architecture diversifiée

1.2 Définition l'internet des objets (IoT)

L'IoT est une technologie en constante progression qui facilite l'échange de données entre différents appareils électroniques. Le CERP-IdO définit l'IoT comme une infrastructure dynamique d'un réseau mondial capable de s'auto-configurer en s'appuyant sur des normes et protocoles de communication interopérables. Dans ce réseau, les objets physiques et virtuels possèdent des identités, des caractéristiques physiques, des personnalités virtuelles et des interfaces intelligentes, s'intégrant de manière transparente au réseau. [1] En termes simples, l'IoT englobe tous les objets pouvant être connectés au réseau Internet. Actuellement, l'accent est mis sur les objets connectés dotés de capteurs, de logiciels et d'autres technologies permettant l'échange de données pour l'information ou l'automatisation.

FIGURE 1.1 – Interaction des objets connectés en utilisant l'internet [2]

1.3 Evolution de l'IoT

Dans les années 1990, l'adoption de l'Internet a progressivement gagné du terrain dans les secteurs commerciaux et grand public. Cependant, son utilisation demeurait limitée en raison des performances relativement modestes des réseaux de connexion. Au cours des années 2000, l'Internet est devenu omniprésent pour de nombreuses applications et est désormais considéré comme une composante essentielle de nombreux produits destinés aux entreprises, à l'industrie et aux consommateurs ; offrant un accès inégal à l'information. Malgré cette évolution, la plupart de ces dispositifs restent essentiellement des entités en ligne qui requièrent une interaction humaine et une supervision par le biais d'applications et d'interfaces dédiées. En 2003, la population mondiale comptait environ 6,3 milliards d'individus, tandis que seulement 500 millions d'appareils étaient connectés à l'Internet. Le calcul du ratio entre le nombre d'appareils et la population mondiale (0,08) révèle qu'il y avait moins d'un appareil connecté par personne. Selon la définition établie par Cisco IBSG, le concept d'IoT n'était pas encore pleinement réalisé en 2003, en raison du nombre relativement restreint d'objets connectés. Grâce à l'essor des smartphones et des tablettes, le nombre d'appareils connectés à l'Internet a atteint 12,5 milliards en 2010, alors que la population mondiale avait atteint 6,8 milliards. C'est ainsi que le ratio d'appareils connectés par personne a dépassé 1 pour la première fois dans l'histoire, avec un chiffre exact de 1,84.Selon les prévisions de Cisco IBSG, environ 50 milliards de dispositifs seront connectés à Internet d'ici 2020. Il est essentiel de souligner que ces prévisions ne prennent pas en compte les progrès rapides de l'Internet ni les avancées technologiques à venir, mais se basent sur les données actuelles [3] [4].

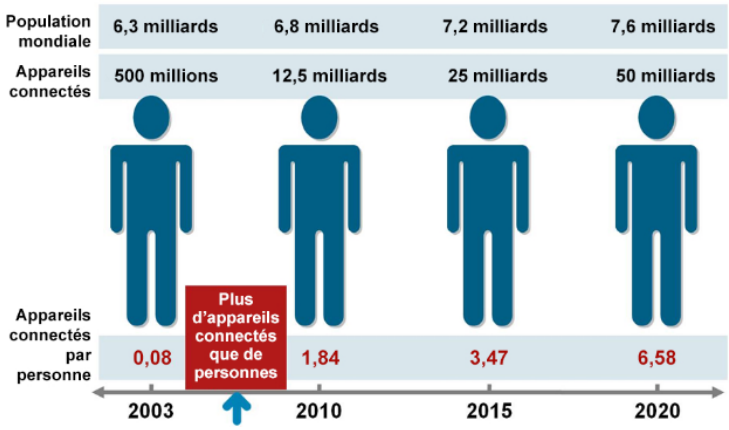


FIGURE 1.2 – Evolution des objets connectés
[5]

1.4 Architecture de L'IoT

L'architecture de l'IoT est composée de plusieurs couches technologiques qui soutiennent l'ensemble de l'écosystème IoT. Elle met en lumière les relations entre les différentes technologies et démontre la capacité d'évolutivité, la modularité et la configuration des déploiements IoT dans divers contextes d'application [6].

- **Capteurs :** la couche la plus fondamentale est constituée d'objets intelligents équipés de capteurs. Ces capteurs créent un lien entre les mondes physique et numérique en collectant et traitant des données en temps réel. Ils mesurent les propriétés physiques et les transforment en signaux interprétables par les instruments. Selon leur utilisation, les capteurs se répartissent en différentes catégories, notamment les capteurs environnementaux, corporels, d'appareils domestiques et de télématique des véhicules, entre autres.
- **Passerelles et réseaux :** les petits capteurs produiront un volume important de données, nécessitant une infrastructure réseau robuste et performante, qu'elle soit filaire ou sans fil, pour assurer leur transport. Les réseaux actuels, souvent basés sur des protocoles variés, ont été utilisés pour prendre en charge les applications machine à machine (M2M). Pour répondre à la demande croissante en matière de services et d'applications IoT, tels que les services transactionnels rapides et les applications contextuelles, plusieurs réseaux avec diverses technologies et protocoles d'accès doivent collaborer dans une configuration hétérogène.
- **Couche de service de gestion :** La couche de gestion des services permet le traitement des données à travers des opérations d'analyse, des contrôles de sécurité et la modélisation des processus. Elle utilise des moteurs de règles pour gérer les processus, répondant rapidement aux événements ou situations d'urgence. Les outils d'analyse, tels que l'analyse en mémoire et en flux, permettent d'extraire des informations pertinentes de grandes quantités de données brutes et de les traiter rapidement. La gestion des données assure le

contrôle des flux d'informations, protégeant les applications des couches supérieures contre le traitement de données inutiles et préservant la confidentialité des sources de données.

- **Couche d'application :** La couche d'application de l'IoT s'adresse à divers domaines, tels que les transports, les bâtiments, les villes intelligentes, l'agriculture, les chaînes d'approvisionnement, les soins de santé, et le commerce de détail. Elle permet l'implémentation de solutions intelligentes pour optimiser l'efficacité, la sécurité, la durabilité et la qualité de vie dans ces secteurs.

1.5 Technologies de connectivité

La communication est un élément clé pour établir un système IoT efficace, permettant l'interconnexion d'objets variés. L'objectif fondamental de ce concept est que tout soit relié. Initialement, les chercheurs se sont concentrés sur le développement des objets connectés, ayant compris que le succès de l'Internet repose sur l'adoption généralisée des protocoles de communication. Ainsi, les protocoles IoT ont fait l'objet de nombreuses recherches au cours des dernières années. Ces protocoles IoT peuvent être catégorisés en deux groupes principaux : les protocoles de communication tels que Bluetooth, Wi-Fi, LoRa, et les protocoles de transmission de données comme MQTT(Message Queuing Telemetry Transport).Nous allons nous concentrer sur certaines d'entre elles. Celles-ci sont décrites ci-dessous.

Nous allons nous pencher sur certaines de ces technologies, qui sont présentées ci-dessous [7] [8].

- **La puce RFID :** est une technologie à courte portée largement utilisée pour l'identification et la géolocalisation grâce à un petit dispositif appelé étiquette ou transpondeur qui peut être attaché à un objet. Cette dernière est composée d'une puce électronique, d'une antenne et d'un substrat ou matériau d'encapsulation. La puce électronique stocke les données tandis que l'antenne transmet et reçoit les données. L'alimentation et la communication avec une étiquette se fait à travers d'un dispositif qui est appelé lecteur d'étiquette. Sa portée courte est idéale pour des applications nécessitant une localisation précise et une récupération rapide d'informations, tout en minimisant les interférences avec d'autres dispositifs. Ainsi, la puce RFID offre une solution polyvalente et efficace pour diverses applications.
- **Le Bluetooth :** technologie à courte portée qui facilite la vie moderne en permettant la connexion sans fil d'équipements électroniques. Utilisé pour la communication dans les dispositifs à faible consommation d'énergie et à faible débit de données. Des enceintes aux montres connectées, il crée des liaisons rapides et fiables, éliminant les contraintes des câbles. Cette technologie discrète offre une expérience utilisateur fluide, faisant du Bluetooth un acteur incontournable de notre quotidien numérique.
- **Le Wifi :** technologie à moyenne portée, sans fil qui permet la transmission de données

entre des appareils électroniques via des ondes radio, il permet aux appareils de se connecter à Internet et de partager des données sans avoir besoin de câbles physiques. Il utilise des protocoles de communication standardisés, tels que les normes IEEE 802.11, pour assurer une compatibilité entre les différents équipements.

- **Le réseau Low-Power Wide-Area Network (LPWAN) :** conçus pour des distances moyennes à longues, sont des acteurs clés dans des applications comme les compteurs intelligents et les bornes de recharge pour véhicules électriques. Leur portée étendue et leur faible consommation d'énergie en font des choix idéaux pour la transmission fiable d'informations sur de grandes distances. Que ce soit pour surveiller la consommation d'énergie ou faciliter la recharge des véhicules électriques, les réseaux LPWAN contribuent à l'efficacité et à la durabilité de nos infrastructures modernes.
- **Message Queuing Telemetry Transpor (MQTT) :** Le protocole MQTT repose sur une architecture client-serveur où les capteurs se comportent comme des clients et se connectent à un serveur appelé broker. Ce protocole est léger et simple à déployer. Il est couramment utilisé pour la connectivité entre les appareils IoT et les communications M2M. MQTT facilite la publication de nouvelles données en temps réel sans nécessiter de solliciter un serveur, offrant ainsi une meilleure réactivité et une optimisation de l'utilisation de la bande passante.

1.6 Domaines d'application

L'IoT a révolutionné de nombreux secteurs en offrant des solutions innovantes et connectées. Voici quelques-uns des domaines d'application les plus notables où l'IoT apporte des changements significatifs :

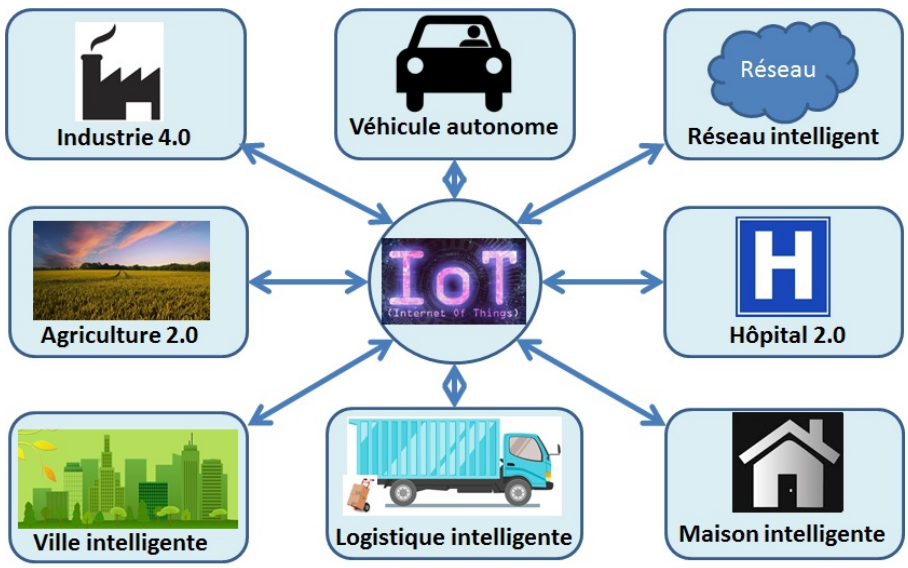


FIGURE 1.3 – Differents secteurs d'application de l'internet des objets

- **La santé :** Dans le domaine de la santé, l'utilisation de capteurs portables connectés aux patients permet aux professionnels de santé de surveiller l'état d'un patient en temps réel ou lorsqu'il est en dehors de l'établissement de soins. Grâce à un suivi progressif de différents indicateurs et à des alertes automatiques sur leurs signes vitaux, l'IoT contribue à améliorer les soins aux patients et à éviter les situations dangereuses chez les patients à risque élevé. De plus, l'intégration de l'IoT dans les lits d'hôpital représente une application importante de cette technologie dans le secteur de la santé. Cela a conduit à la création de lits intelligents dotés de capteurs pour surveiller les signes vitaux des patients, tels que la pression artérielle et la température corporelle [9].

FIGURE 1.4 – IoT dans la santé
[10]

- **Ville intelligente :** L'IoT facilite la gestion précise et continue des réseaux urbains essentiels, tels que l'eau, l'électricité et le gaz. Les capteurs intelligents améliorent l'économie d'eau, la gestion des parkings et la fluidité du trafic, réduisant les embouteillages et les émissions de CO2 [11].Les technologies IoT optimisent également l'efficacité énergétique, la gestion des déchets et l'éclairage public, contribuant ainsi à l'amélioration de la qualité de vie des citoyens.
- **Domotique :** L'IoT est fréquemment intégré dans les domiciles intelligents afin d'automatiser et superviser divers équipements à partir d'une interface centralisée, telle que le téléphone ou le panneau de contrôle. Son objectif principal est d'offrir des fonctionnalités axées sur le confort, comprenant la possibilité de commander à distance, la gestion énergétique (optimisation de l'éclairage et du chauffage, entre autres), la sécurité (systèmes d'alarme), et les aspects liés à la communication (interaction avec des individus externes via contacts et discussions) [12].
- **Agriculture intelligente :** La gestion et le contrôle à distance de paramètres agricoles

tels que la température, l'humidité, l'irrigation et les microclimats sont facilités par les technologies de l'IoT. Cela assure une production de qualité et prévient les pertes financières. Dans les systèmes agricoles intelligents, des capteurs sur les animaux permettent de surveiller leur comportement et leur santé, optimisant la gestion des cultures et de l'élevage [12].

- **Le Transport :** L'IoT, par le biais des véhicules connectés ou autonomes ainsi que des systèmes de transport et de logistique intelligents, contribue à sauver des vies, à fluidifier la circulation et à atténuer les impacts environnementaux liés aux véhicules [11].
- **Industrie :** La technologie IoT permet un suivi intégral des produits, de la production à la distribution, optimisant ainsi l'efficacité, la production et la sécurité dans les usines. Elle sert également à surveiller et contrôler les conditions agricoles, améliorant la gestion des exploitations [13].

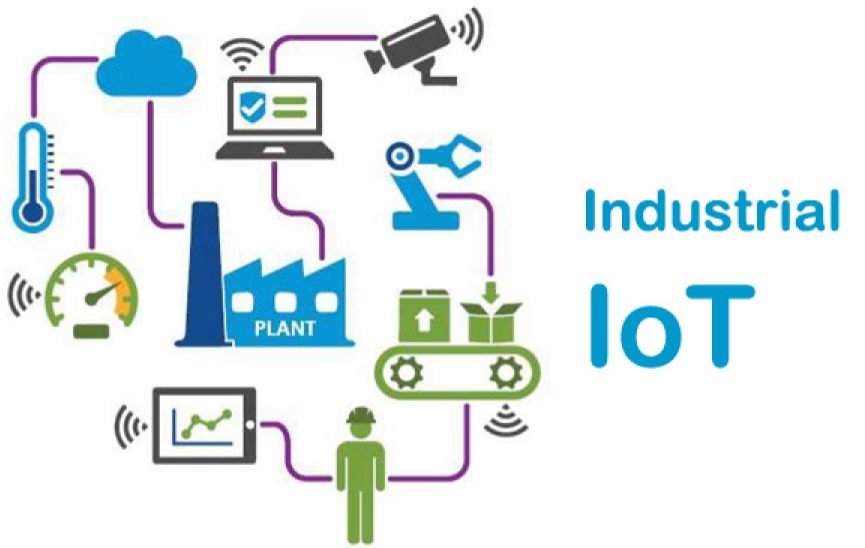


FIGURE 1.5 – Internet des objets dans l'industrie
[14]

1.7 Importance de L'IoT

L'IoT révolutionne notre manière d'interagir avec le monde en connectant des objets physiques à des systèmes informatiques. Cette interconnexion offre des avantages significatifs, de l'optimisation des processus industriels à l'amélioration de la prise de décision, renforçant la sécurité et contribuant à une gestion intelligente des ressources. Dans des domaines tels que la santé et la domotique, l'IoT améliore la qualité de vie en surveillant la santé des individus et en automatisant des tâches quotidiennes. En stimulant l'innovation, il ouvre la voie à de nouvelles applications et services. Cependant, son adoption pose des défis en matière de sécurité, confidentialité des données et gestion du volume massif de données. Une approche réfléchie est nécessaire pour garantir le succès à long terme de cette technologie prometteuse.

1.8 Principes fondamentaux de la sécurité des dispositifs IoT

La sécurité de l'IoT vise à garantir la protection des dispositifs connectés à Internet ainsi que des réseaux auxquels ils sont liés contre les menaces et les vulnérabilités en détectant et en surveillant les risques, et en corrigeant les failles de sécurité des appareils susceptibles de compromettre la sécurité de votre entreprise. En fin d'année 2020, une machine à laver connectée, par défaut non sécurisée, a été compromise par des pirates informatiques afin d'accéder au système informatique d'un établissement de santé, exposant ainsi toutes les données des patients.

Les principes fondamentaux de la sécurité des dispositifs IoT incluent [15] [16] :

- **Surveillance continue :** Mettre en place des systèmes de surveillance pour détecter les activités suspectes ou les comportements anormaux sur les dispositifs IoT et les réseaux associés. Cela peut inclure l'utilisation de solutions de détection d'intrusion et de gestion des événements de sécurité (SIEM).
- **Correction des vulnérabilités :** Appliquer régulièrement des correctifs de sécurité et des mises à jour logicielles pour remédier aux vulnérabilités connues. Les fabricants et les fournisseurs de services IoT doivent également être vigilants dans la fourniture de correctifs pour leurs dispositifs et logiciels
- **Authentification et autorisation :** Mettre en œuvre des mécanismes d'authentification robustes pour garantir que seuls les utilisateurs autorisés peuvent accéder aux dispositifs IoT et aux données qu'ils collectent. De plus, limiter les privilèges d'accès pour réduire les risques d'exploitation.
- **Chiffrement des données :** Utiliser le chiffrement pour protéger les données sensibles transitant entre les dispositifs IoT et les serveurs, ainsi que pour stocker les données sur les dispositifs eux-mêmes. Cela garantit que si les données sont interceptées, elles ne peuvent pas être facilement lues.
- **Gestion des accès :** Mettre en place des politiques de gestion des accès pour contrôler qui peut accéder aux dispositifs IoT et aux données qu'ils génèrent. Cela implique la gestion des identités, l'attribution de rôles et la révocation des droits d'accès lorsque cela est nécessaire.

1.9 Définition de L'Objet Connecté

Un objet connecté (OC) se définit par sa fonction première qui ne se limite ni à celle d'un système informatique ni à une interface web. L'intégration d'une connexion Internet à un OC permet d'interagir avec le monde physique de manière autonome, sans nécessiter d'intervention humaine en lui permettant de communiquer avec d'autres systèmes pour obtenir ou fournir de l'information. Il doit être adapté à un usage spécifique, possédant une forme d'intelligence

et la capacité de recevoir et de transmettre des données grâce à des logiciels intégrés et des capteurs embarqués. La communication entre les objets s'effectue par le biais d'identifications mutuellement reconnues. Ainsi, un objet doit disposer d'un ou plusieurs identifiants pour être reconnu par un autre objet et établir une connexion [1].

— **Fonctionnement d'un objet connecté**

Le processus de fonctionnement d'un oc peut être divisé en cinq phases distinctes [17] :

- **Acquisition de données :** Les données sont recueillies par le biais de capteurs et/ou de la technologie RFID12, permettant de suivre et d'identifier les données générées par l'objet connecté.
- **Stockage de données :** les données sont généralement stockées dans des centres de données, mais il est également possible de les stocker localement.
- **Analyse des données :** Les données stockées dans les centres de données sont soumises à une analyse.
- **Transmission des données :** les données captées par les capteurs, les SoC13 (system on a Chip) ou les dispositifs RFID (identification par radiofréquence) sont transmises aux centres de données. Après analyse, les données sont acheminées des centres de données vers les unités de traitement, puis vers les processeurs, les contrôleurs, les appareils ou les utilisateurs.
- **Distribution des résultats :** les résultats sont transmis sans altération.

1.10 Conclusion

Ce premier chapitre nous a permis d'explorer en détail le domaine passionnant de lIoT. Nous avons commencé par définir l'IoT et examiner son évolution, depuis ses débuts modestes jusqu'à son rôle central dans notre société moderne. Nous avons également examiné les composants techniques de l'IoT, y compris les capteurs, les réseaux de capteurs, les actionneurs et les technologies de connectivité.

Ensuite, nous avons exploré les divers domaines d'application de l'IoT, allant de la santé à l'industrie en passant par les villes intelligentes et la domotique. Nous avons vu comment l'IoT transforme ces secteurs en offrant des solutions innovantes et connectées pour relever les défis contemporains.

Enfin, nous avons souligné l'importance de l'IoT dans notre société, tout en reconnaissant les défis qu'il pose en termes de sécurité, de confidentialité des données et de gestion des volumes massifs de données.

Chapitre 2

Fonctions physiquement non clonable (PUFs) dans IoT

2.1 Introduction

Les fonctions physiquement unclonables (PUFs) sont devenues une solution prometteuse pour garantir la sécurité et l'authentification dans les systèmes IoT en exploitant les caractéristiques uniques des dispositifs physiques. Leur nature intrinsèquement difficile à cloner ou à reproduire offre un niveau de sécurité élevé, ce qui les rend essentielles dans la sécurisation des systèmes embarqués. Parallèlement, l'attrait croissant de la communauté de recherche et de l'industrie pour les PUFs témoigne de leur adoption généralisée en tant que technologie de sécurité physique de premier plan. Des entreprises leaders telles que NXP(Next eXPerience), Microsemi, Intel(Integrated Electronics) et Xilinx ont déjà mis en œuvre cette technologie pour développer des circuits intégrés sécurisés. Les chercheurs se sont également penchés sur les PUFs pour développer des protocoles d'authentification légers et sécurisés adaptés aux besoins des applications IoT.

2.2 Définition de la PUF

Une PUF, ou fonction physiquement non clonable, est un dispositif physique doté de propriétés uniques et complexes, issues de variations naturelles ou aléatoires pendant sa fabrication. Sa duplication est extrêmement difficile en raison de ces variations aléatoires [18]. Le concept de PUF est né de l'observation de la microstructure physique des circuits intégrés en silicium, révélant que les variations du processus de fabrication peuvent générer des réponses uniques, comparables à des empreintes digitales pour chaque circuit. Contrairement aux fonctions mathématiques traditionnelles [19], le PUF est un système de défi-réponse qui produit une réponse aléatoire et imprévisible à un défi donné, en se basant sur les caractéristiques uniques et aléatoires des composants physiques. Cette singularité empêche toute reproduction

exacte, même en recréant le circuit avec la même technologie, la réponse sera différente [20].

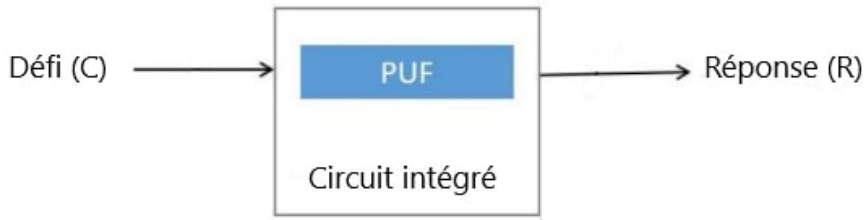


FIGURE 2.1 – Architecture d'un PUF

2.3 Motivation d'adopter les PUF

La popularité croissante des PUF en tant que technologie de sécurité physique fiable parmi de nombreux chercheurs et acteurs industriels s'explique par les raisons suivantes [21] :

- Les appareils à ressources limitées nécessitent des solutions de sécurité efficaces tout en maintenant des coûts matériels bas. Dans ce cas, les PUFs ont un coût matériel nettement inférieur. Cette économie de ressources les rend parfois parfaitement adaptées aux appareils à ressources limitées, garantissant le bon fonctionnement des systèmes numériques. Contrairement aux algorithmes cryptographiques traditionnels qui exigent une grande quantité de portes logiques pour les opérations de base.
- De nombreux dispositifs à ressources limitées sont peu coûteux et dépourvus de sécurité physique, les rendant ainsi vulnérables aux attaques d'intrusion. Intégrer une sécurité physique à ces appareils augmenterait leurs coûts. Les PUFs offrent une solution en permettant d'instaurer une sécurité physique en exploitant le circuit existant de l'appareil, sans nécessiter l'ajout de nouveaux composants.
- De manière traditionnelle, les clés cryptographiques sont enregistrées dans la mémoire non volatile des appareils (comme la ROM(Read-Only Memory), le fusible ou l'EEPROM(Electrically Erasable Programmable Read-Only Memory)). Ces clés sont principalement utilisées pour l'authentification, ce qui engendre un coût substantiel pour sécuriser cette mémoire non volatile. Grâce aux PUFs, les clés d'authentification ne sont plus stockées sur l'appareil, mais sont plutôt créées et extraites dynamiquement à partir du circuit de l'appareil.
- Les PUFs permettent de créer des circuits uniques, rendant ainsi les appareils correspondants non clonables. En outre, comme les clés cryptographiques ne sont pas conservées sur les appareils, ces derniers sont considérés comme incapable d'être violé, altéré ou compromis d'une manière quelconque.

2.4 Conception d'une PUF

La conception d'une PUF est un processus complexe et méthodique. Chaque étape de ce processus est importante pour garantir que la PUF est unique, fiable et sécurisée. Voici une explication détaillée des principales étapes impliquées dans la conception d'une PUF [18] :

1. **Identification du Désordre Physique**

   L'objectif est de trouver un désordre physique propre à chaque objet. Lors du processus de fabrication, des variations inévitables surviennent en raison de facteurs tels que les tolérances de fabrication, les impuretés des matériaux et les conditions environnementales. Ces variations, qui sont souvent aléatoires et imprévisibles, créent des différences minimes dans les dimensions des composants, des irrégularités dans la structure du silicium ou des variations dans les propriétés électriques des transistors. Ce désordre physique est la base sur laquelle la PUF sera construite.

2. **Mesure et Quantification du Désordre Physique**

   Après avoir identifié le désordre physique, il est essentiel de le mesurer précisément et de le quantifier. Cette étape implique l'utilisation de techniques de mesure de haute précision pour capturer les variations physiques identifiées. Par exemple, des capteurs électroniques peuvent être utilisés pour mesurer les variations de résistance, de capacité ou de fréquence. Ces mesures physiques doivent ensuite être transformées en données numériques, comme convertir les variations de tension en un code binaire. Cette quantification permet de manipuler et de traiter le désordre physique de manière efficace pour les étapes suivantes.

3. **Développement de Modèles Mathématiques et Algorithmiques**

   L'objectif ici est de créer des modèles robustes pour stocker et utiliser le désordre physique de manière sécurisée. Les algorithmes développés doivent être capables de générer une réponse unique à partir des caractéristiques physiques mesurées, tout en étant suffisamment sensibles pour distinguer les petites différences afin de garantir l'unicité de chaque PUF. De plus, ces modèles doivent être conçus pour résister aux attaques et ne pas pouvoir être facilement clonés ou reproduits, assurant ainsi la sécurité de la PUF

4. **Évaluation des Performances**

   Cette évaluation comprend plusieurs aspects. Premièrement, l'unicité : chaque PUF doit générer une réponse unique ; cela peut être testé en fabriquant plusieurs PUFs et en comparant leurs réponses. Deuxièmement, la fiabilité : la PUF doit produire la même réponse à une même question (challenge) à chaque fois que le test est effectué. Enfin, la résistance aux attaques : il est crucial de tester la PUF contre diverses attaques physiques et logiques pour s'assurer qu'elle ne peut pas être copiée ou contournée

5. **Implémentation Matérielle**

   C'est la dernière étape dans la conception d'une PUF. Cette étape consiste à implémenter la PUF de manière effective dans le matériel en exploitant les variations inévitables lors

de la fabrication. Par exemple, les différences de fréquence dans les oscillateurs en anneau ou les variations dans la SRAM(Static Random-Access Memory) peuvent être utilisées pour générer des réponses uniques.

2.5 Métriques d'évaluation des dispositifs PUF

Les métriques d'évaluation des dispositifs PUF sont des mesures utilisées pour évaluer la qualité et la performance des fonctions physiquement non clonables. Voici quelques métriques couramment utilisées pour évaluer les dispositifs PUF [18] [19] :

- **Unicité :** cette caractéristique mesure la capacité d'une PUF à produire des identifiants distincts et uniques. Elle est évaluée en comparant les réponses de différents PUFs soumis au même défi. Idéalement, chaque réponse devrait être unique, ce qui correspondrait à une valeur de 50% pour cette mesure.
- **Fiabilité :** la fiabilité d'une PUF se réfère à sa capacité à produire de manière cohérente la même réponse pour un défi donné, indépendamment des variations environnementales. Cette propriété est évaluée en mesurant la cohérence des réponses d'un même PUF face au même défi dans différentes conditions.
- **Uniformité :** mesure l'imprévisibilité des réponses d'une PUF. Elle est définie comme la proportion de 0 et de 1 dans les bits de réponse d'une PUF. Dans une réponse véritablement aléatoire, cette proportion est de 50%.
- **Résistance aux manipulations :** : mesure la capacité d'une PUF à résister aux tentatives de falsification. Idéalement, le comportement du PUF devrait changer complètement si sa conception ou sa structure est modifiée ou "manipulée" de quelque manière que ce soit par des adversaires.

2.6 Applications des PUF liées à la sécurité

L'approche traditionnelle des algorithmes cryptographiques implique l'installation préalable d'une clé dans les appareils informatiques, exposant ces dispositifs à des vulnérabilités telles que la lecture de la mémoire et les attaques par rémanence des données. De plus, cette méthode implique souvent l'implication des fabricants dans la gestion du processus d'insertion de la clé, ce qui accroît les risques de sécurité. La technologie PUF élimine le besoin de stocker la clé dans une mémoire sur puce, renforçant ainsi la sécurité en réduisant les risques d'attaques [18].

- **Authentification à faible coût :** la simplicité de la conception des PUF en fait une option attrayante pour les schémas d'authentification économiques, où le comportement unique de défi/réponse des PUF est utilisé comme identifiant pour les objets physiques. Cette approche est particulièrement bénéfique pour les systèmes à ressources limitées

comme les appareils IoT, qui ne peuvent pas supporter les solutions de sécurité conventionnelles.

- **Conception anti-contrefaçon :** la technologie PUF offre une solution contre la surproduction de circuits intégrés par des usines malveillantes en sécurisant chaque circuit PUF pour préserver la conception. Après la fabrication, chaque PUF est examinée et validée par le bureau d'études, permettant la création d'un mot de passe pour activer uniquement les puces authentiques.

- **Jetons matériels sécurisés :** les PUF peuvent également être intégrés dans des protocoles cryptographiques assistés par matériel en raison de leurs comportements de défi/réponse complexes, ce qui les rend naturellement plus résistants aux falsifications que d'autres jetons reposant sur des informations numériques stockées (comme les cartes à puce et les dispositifs de mémoire sécurisés).

- **Télédétection sécurisée :** les circuits PUF réagissent aux fluctuations des paramètres environnementaux comme la température et la tension d'alimentation, ce qui implique que leur réponse dépend à la fois des défis reçus et des conditions ambiantes. Cela permet d'utiliser les PUF comme des capteurs pour détecter les changements dans l'environnement.

2.7 Types de PUFs

La création des PUF peut se faire selon diverses approches les approches non électriques des PUF, telles que les PUF optiques, acoustiques et de revêtement. En revanche, les PUF électriques basés sur le silicium, comme les PUF SRAM et les PUF arbitres, utilisent des variations électriques dans les composants électroniques pour produire des signatures uniques. Ils sont souvent plus simples à fabriquer et plus économiques, tirant parti des technologies de semi-conducteurs bien établies.

2.7.1 Approches Non Électriques

La construction de ces dispositifs est généralement plus complexe que celle des PUF électriques, en raison des matériaux spécifiques et des techniques de fabrication avancées nécessaires, qui nécessitent souvent des équipements et des procédés spécialisés pour être fabriqués. Cette complexité de fabrication entraîne des coûts élevés, car la précision et la spécialisation nécessaires augmentent les coûts de production.

Ces PUFs reposent sur des mesures prises à partir d'une couche spéciale de matériau déposé, contenant des éléments spécifiques :

- **PUFs optiques :** ces PUFs utilisent les propriétés optiques uniques de matériaux pour générer des réponses. Par exemple, une lumière laser passant à travers un matériau aléatoire produit un motif de diffraction unique, utilisé comme réponse. Ils sont particulièrement adaptés aux applications nécessitant une très haute sécurité [22].

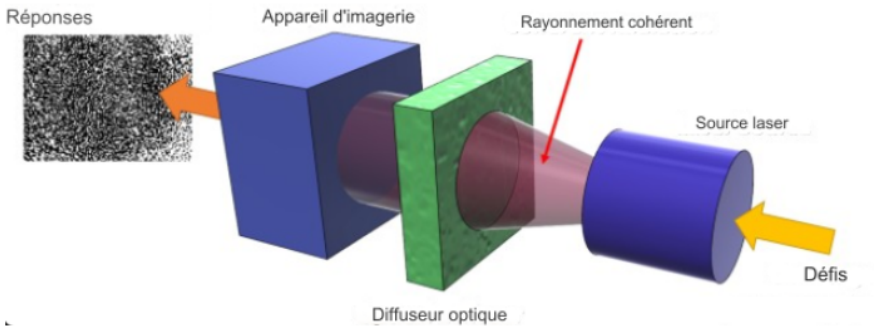


FIGURE 2.2 – PUF optique
[23]

- **PUF de Revêtement :** ces PUFs utilisent un revêtement aléatoire appliqué sur un circuit intégré. Ce revêtement contient des particules dispersées de manière aléatoire, créant une structure unique. Ils sont souvent utilisés pour la protection contre le clonage et la contrefaçon des dispositifs électroniques [22].

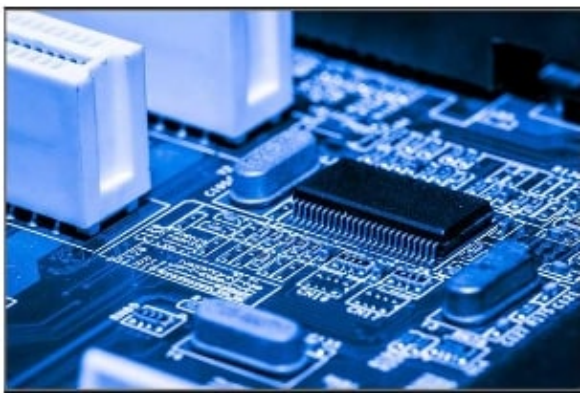


FIGURE 2.3 – PUF de revêtement
[24]

- **PUFs de magnétiques :** ces PUFs exploitent les variations naturelles dans les propriétés magnétiques des matériaux. Ils génèrent des réponses en fonction de l'interaction unique entre un champ magnétique et les imperfections microscopiques du matériau magnétique [21].

2.7.2 Approches Électriques Basées sur le Silicium

La fabrication des PUF électriques est souvent moins complexe que celle des PUF non électriques, car elle repose sur des technologies de fabrication de semi-conducteurs bien établies,

qui sont bien maîtrisées et largement disponibles. Les coûts de production peuvent ainsi être plus faibles, rendant les PUF électriques plus économiques pour des applications à grande échelle. La standardisation et l'efficacité des procédés de fabrication réduisent les coûts unitaires.

- **PUFs arbitres :** ces PUFs exploitent les variations intrinsèques de retard dans les dispositifs pour générer des identités uniques en utilisant les délais de propagation des signaux dans les circuits logiques. Premier PUF en silicium, il est idéal pour la sécurité matérielle légère et est couramment utilisé pour l'authentification des dispositifs grâce à sa capacité à produire des réponses uniques [22] [25].

- **PUFs SRAM :** ces PUFs tirent parti de l'état de démarrage aléatoire des cellules de mémoire SRAM pour générer une signature unique à chaque mise sous tension, basée sur les valeurs aléatoires des cellules. Les SRAM émettent de l'énergie lorsqu'elles changent d'état, ce qui peut être détecté en analysant la longueur d'onde du laser à l'aide d'un analyseur de signaux [21] [26].

2.8 Attaques liées aux PUFs

Malgré leurs avantages, les PUFs sont vulnérables à divers types d'attaques. Comprendre ces attaques est crucial pour développer des contre-mesures efficaces et garantir la sécurité des systèmes qui utilisent des PUFs [21].

- **Attaque d'invalidation PUF :** cette attaque suppose que l'attaquant ait un accès physique au PUF d'un appareil. L'attaquant modifie de manière aléatoire le circuit PUF, c'est-à-dire sa sortie, change et imite celui d'un autre PUF. Par exemple, l'attaquant peut altérer le circuit PUF en utilisant des substances chimiques pour modifier sa conductivité. De plus, l'attaquant peut placer des dispositifs électromagnétiques à proximité du circuit PUF afin de perturber sa sortie. Par conséquent, toute tentative ultérieure d'authentification avec le PUF du périphérique échouera car le PUF aura été altéré.

- **Attaque d'usurpation d'identité PUF :** un individu malveillant ayant accès à tous les CRP (réponses physiquement non clonables) d'un appareil peut se faire passer pour cet appareil légitime et réussir à s'authentifier auprès d'autres appareils utilisant les mêmes CRP pour l'authentification. Ce type d'attaque pourrait également être réalisé dans une architecture de type "Thing-to-Thing" (T2T), où les appareils stockent mutuellement les CRP à des fins d'authentification.

- **Attaque d'écoute clandestine :** un attaquant intercepte les échanges d'authentification qui incluent les défis et les réponses calculées par le PUF d'un appareil cible. Ensuite,

l'attaquant se fait passer pour l'appareil légitime et répond correctement au vérificateur. Cette attaque réussit généralement en raison de deux hypothèses : premièrement, le nombre de réponses physiquement non clonables (CRP) possibles est trop restreint, et deuxièmement, le même défi est réutilisé.

- **Attaque d'apprentissage automatique :** un attaquant collecte un grand nombre de paires défi-réponse (CRP) en interagissant avec le PUF d'un appareil, soit lors du processus d'authentification, soit en soumettant de nombreux défis au PUF pour obtenir les réponses correspondantes. Ensuite, ces CRP collectés sont utilisés pour entraîner des algorithmes d'apprentissage automatique afin de construire un modèle logiciel du PUF capable de prédire avec précision les réponses aux nouveaux défis.

2.9 Types d'adversaires

Les adversaires peuvent être classés en fonction de leurs capacités, de leurs objectifs et de leurs méthodes d'attaque. Comprendre ces distinctions permet de mieux concevoir des contre-mesures efficaces pour protéger les PUFs. Voici les principaux types d'adversaires rencontrés dans ce contexte [18] :

- **Adversaire d'Écoute Clandestine (Espionnage)**
  Ces adversaires se contentent de surveiller passivement les communications et les interactions du dispositif PUF avec son environnement. Ils interagissent pas directement avec le dispositif mais cherchent à obtenir des informations sensibles à travers l'observation des réponses PUF aux défis standards. L'objectif principal de cet adversaire est de collecter des données sans être détecté.

- **Adversaires de la Boîte Noire**
  Les adversaires de la boîte noire ont accès au dispositif PUF et peuvent y appliquer un nombre polynomial de défis. Ils enregistrent les réponses correspondantes sans connaître les détails internes du dispositif. Leur approche consiste à utiliser ces données pour construire un modèle prédictif du comportement du PUF, généralement à l'aide de techniques d'apprentissage automatique.

- **Adversaires de la Boîte Blanche**
  Contrairement aux adversaires de la boîte noire, ceux de la boîte blanche ont une connaissance approfondie de la structure interne et des mécanismes de fonctionnement du PUF. Ils peuvent manipuler directement le matériel, exécuter des attaques par canal auxiliaire pour extraire des informations sur la synchronisation, la consommation d'énergie

ou les fuites électromagnétiques. Cette connaissance leur permet de mener des attaques plus sophistiquées et potentiellement plus destructrices.

2.10 Conclusion

Dans ce chapitre, nous avons présenté les approfondie des PUFs et de leur rôle important dans la sécurité des systèmes IoT. Nous avons débuté par une introduction aux PUFs, soulignant leur importance croissante dans l'industrie et la recherche en raison de leurs propriétés uniques et difficiles à cloner. Après avoir défini les PUFs et exploré les motivations derrière leur adoption, nous avons détaillé les différentes approches de conception des PUFs ainsi que les métriques d'évaluation utilisées pour mesurer leur performance et leur fiabilité. Nous avons également examiné les diverses applications des PUFs dans le domaine de la sécurité, en particulier pour l'authentification à faible coût, et distingué les différents types de PUFs, en mettant en évidence ceux nécessitant des processus de fabrication spéciaux et ceux basés sur le silicium. Enfin, nous avons abordé les vulnérabilités potentielles des PUFs et les types d'attaques auxquelles elles peuvent être exposés, en identifiant les adversaires potentiels.

Chapitre 3

Protocoles d'authentification basés sur PUF

3.1 Introduction

Un protocole cryptographique est essentiellement un programme utilisant des primitives comme le chiffrement pour garantir des propriétés telles que le secret et l'authentification. Malgré sa simplicité apparente, la conception de tels protocoles est intrinsèquement complexe et sujette à de nombreuses erreurs [27]. Dans ce chapitre , nous explorons plusieurs protocoles innovants d'authentification pour les appareils IoT, basés sur les PUFs. Ces solutions sont conçues pour renforcer la sécurité et répondre à des critères exigeants tels que l'efficacité énergétique et la résistance aux attaques physiques, tout en exploitant les caractéristiques physiques uniques des PUF pour assurer la robustesse des systèmes IoT.

3.2 Travaux connexes

Dans cet article [28], les auteurs proposent des protocoles pour l'authentification légèrs et la communication sécurisée adaptées aux dispositifs IoT. Ils utilisent un seul circuit pour générer des clés à l'aide de PUF, ce qui contribue notamment à simplifier la gestion des clés sur des dispositifs matériels simples et des microcontrôleurs. Ces protocoles illustrent les possibilités de sécuriser la communication et l'authentification entre les appareils IoT de manière à éviter le stockage de secrets sur le dispositif matériel, tout en s'assurant que seules les parties autorisées puissent connaître la clé. Cette stratégie réduit les contraintes de mise en œuvre et optimise l'utilisation des ressources.

Dans cet article [29], les auteurs ont proposé un protocole léger d'authentification et d'échange de clés basés sur la PUF comme primitive de sécurité. Ce protocole est conçu pour les appareils IoT ayant des ressources limitées. Il stocke une seule paire de défis au niveau du serveur et utilise des opérations simples telles que les fonctions de hachage, la génération de nonces, les registres

à décalage à rétroaction linéaire (LFSR) et l'opérateur XOR pour garantir une communication sécurisée et protéger contre les tentatives de manipulation. Les auteurs ont analysé la sécurité du protocole et l'ont vérifiée en utilisant l'outil Proverif.

Les auteurs de cet article [30] ont proposé un protocole d'authentification léger pour les appareils IoT. Le protocole repose sur un circuit PUF robuste du côté de l'appareil contraint et un modèle PUF logiciel du côté de l'appareil non contraint. Les points d'accès sont définitivement désactivés après la création du modèle logiciel afin d'empêcher les attaques ciblant les points de mesure pour collecter les données défi-réponse. Le protocole n'utilise pas d'opérations cryptographiques pour éviter d'augmenter la taille d'implémentation matérielle, ce qui pourrait ne pas convenir aux petits appareils. Malgré cela, le protocole assure une protection contre les attaques de modélisation. Il garantit également l'échange de messages secrets authentifiés entre les parties de confiance. Au lieu d'utiliser des fonctions cryptographiques ou de hachage, le protocole utilise des fonctions de transformation non corrélées pour garantir l'intégrité des messages et maintenir une faible taille d'implémentation matérielle. La sécurité du protocole proposé a été testée contre les attaques de modélisation, démontrant ainsi sa résilience.

3.3 Contexte de notre travail

Dans cette partie, nous examinerons de manière détaillée les protocolesc cités précédemment dans les articles([28], [29], [30]) après avoir sélectionné leurs critères de comparaison parmi différents critères.Par la suite, on effectue des comparaisons entre ces protocoles en créant des parties de discussion et de critique, puis on les résume en un tableau et on propose une amélioration pour le dernier protocole de l'article [28].

3.4 Criteres de comparaison

- **Robustesse :** La robustesse d'un système fait référence à sa capacité à résister efficacement aux attaques et aux perturbations, tout en maintenant un fonctionnement correct et sécurisé. Cela inclut la capacité à contrer les tentatives de cryptanalyse en préservant l'intégrité et la confidentialité des données [31].

- **Scalabilité :** est la capacité d'un système à s'adapter à une augmentation de sa taille ou de sa charge de travail sans compromettre ses performances. Elle assure que le système continue de fonctionner efficacement même lorsque le nombre d'utilisateurs, de données ou d'opérations augmente [31].

- **Complexité :** mesurée en fonction de sa consommation d'énergie, de ses exigences de stockage et de la charge de calcul nécessaire pour ses opérations. Une architecture nécessitant moins de ressources dans ces domaines est considérée comme moins complexe,

ce qui peut la rendre plus efficace et plus facile à gérer [32].

3.5 Authentification légère et communication sécurisée Adaptées aux appareils IoT

Cet article propose trois protocoles et une unique phase d'inscription pour l'ensemble des trois protocoles.

3.5.1 Phase d'inscription

Cette phase comprend l'inscription d'un dispositif, désigné D1, auprès de l'autorité d'authentification (AA), qui maintient une base de données sécurisée nommée DBD1. Cette base contient des paires de défis et réponses (C, R). Les étapes détaillées de ce protocole sont les suivantes :

- L'autorité AA envoie un message à D1 comprenant un ensemble de défis (C1, C2, C3, etc.).
- À la réception de ce message, le dispositif D1 calcule des réponses (R1, R2, R3, etc.) selon la formule Ri = PUF(Ci), puis les transmet à l'autorité AA.
- L'autorité AA enregistre ensuite les paires (Ci, Ri) dans sa base de données, puis transmet sa clé publique à D1, qui la stocke.

3.5.2 Premier protocole

ce protocole, que nous appellerons protocole 1 permet d'authentifier un appareil D1 auprès de l'autorité AA. Les étapes détaillées de ce protocole sont les suivantes :

- L'autorité AA sélectionne une paire clé défi-réponse (C, R) et transmet le défi C accompagné d'un *"nonce"* fraîchement généré N à l'appareil D1.
- À réception du message, l'appareil D1 génère une réponse R'=PUF(C), puis l'associe au nonce N. Ensuite, D1 chiffre l'ensemble R'||N avec la clé publique de l'autorité PKAA (Public Key Authentication autorité), créant ainsi CR = EPKAA(R'||N), qu'il envoie ensuite à AA.
- AA décrypte le message, vérifie si la réponse R' reçue correspond à la réponse R stockée dans sa base de données, et s'assure que le nonce reçu est identique à celui envoyé.

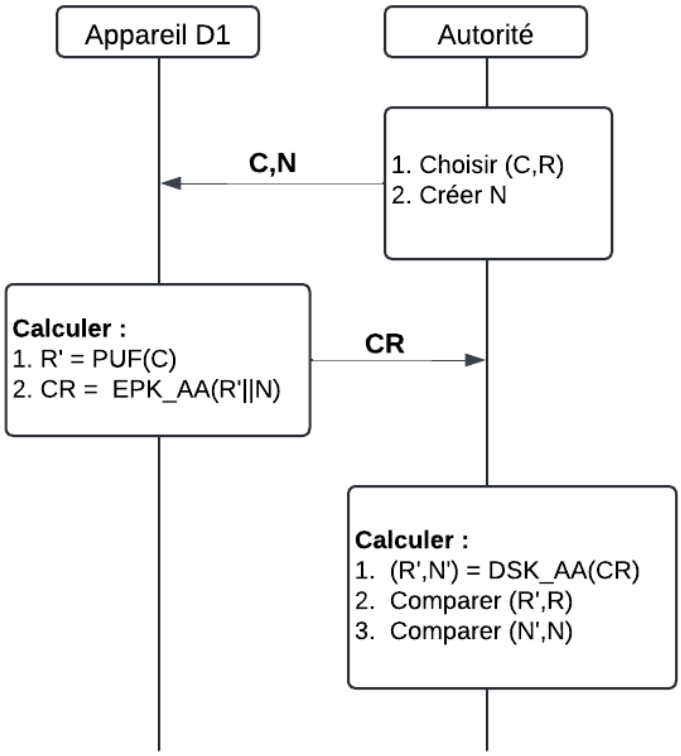


FIGURE 3.1 – Diagramme du premier protocole

3.5.3 Deuxieme protocole

Ce protocole, que nous appellerons protocole 2 propose un mécanisme d'authentification pour l'appareil D1 avec l'autorité AA, tout en établissant une clé symétrique partagée pour les communications chiffrées futures, ce qui le distingue du protocole précédent. Voici les étapes spécifiques de ce protocole :

- L'appareil D1 démarre le protocole en envoyant le message Call(D1) à l'autorité AA.
- L'autorité AA génère des données aléatoires r et sélectionne une paire défi-réponse (C, R). Elle crée ensuite une chaîne d'aide H = R xor encode(r).
- L'autorité dérive une clé K à partir de la fonction de dérivation de clé KDF(r), puis envoie le défi C et la chaîne d'aide H à l'appareil D1.
- L'appareil D1 génère une réponse R' à partir du défi reçu, en utilisant la fonction PUF(C). Il calcule ensuite les données r en décodant R' xor H, et dérive la clé partagée K grâce à la fonction KDF(r).

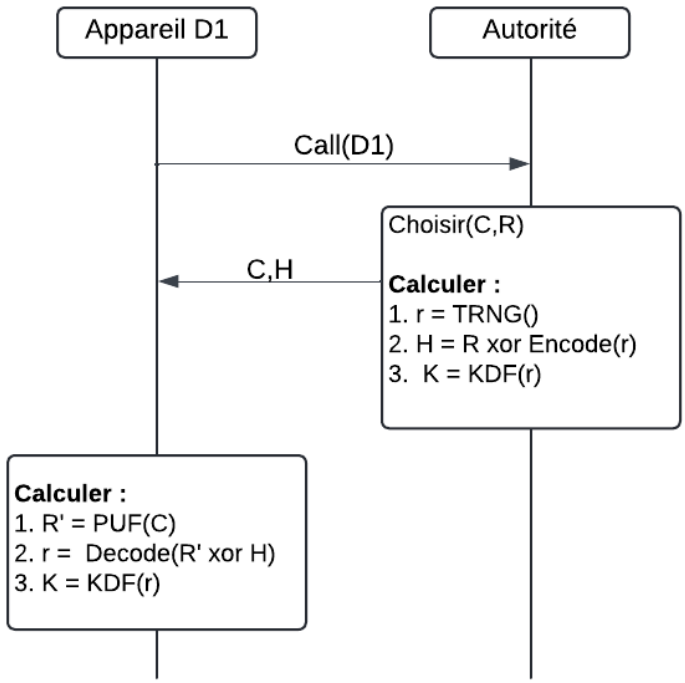


FIGURE 3.2 – Diagramme du deuxiéme protocole

3.5.4 Troisieme protocole

Ce protocole, que nous appellerons protocole 3 permet une authentification mutuelle entre les appareils D1 et D2, tout en établissant une clé symétrique partagée K entre ces appareils pour les communications chiffrées futures. Il utilise également la fonction de hachage Hash pour sécuriser le processus.

Voici les étapes détaillées de ce protocole :

- L'appareil D1 initie le protocole en envoyant le message Call(D1, D2) à l'autorité AA pour commencer la procédure de connexion avec D2.
- Ensuite, l'autorité AA génère deux composants aléatoires, rD1 et rD2, à partir des préimages de l'ECC et les encode, créant ainsi des codes aléatoires sélectionnés.
- L'autorité sélectionne deux paires défi-réponse de sa base de données (CD1, RD1) et (CD2, RD2), puis crée les chaînes d'aide HD1 et HD2 de la manière suivante : HD1 = RD1 xor Encode(rD1) et HD2 = RD2 xor Encode(rD2).
- Les deux composants aléatoires sont hachés et combinés par un XOR, ce qui donne r = Hash(rD1) xor Hash(rD2). L'autorité envoie alors les triplets (CD1, HD1, r) et (CD2, HD2, r) aux appareils D1 et D2.
- Les deux appareils génèrent leurs réponses aux défis respectifs acceptés, à savoir R'Di = PUF(CDi). Les appareils appliquent ensuite un XOR à leurs réponses avec les données d'aide HDi, puis les décodent pour obtenir le composant aléatoire rDi correspondant à chacun.

- Chaque appareil calcule le hash du composant aléatoire de l'autre appareil en utilisant un XOR de r avec le hash de son propre composant aléatoire : Hash(rD2) = Hash(rD1) xor r.
- Enfin, chaque appareil calcule la clé partagée K à l'aide de la fonction de dérivation de clé (KDF) sur les hash de rD1 et rD2 combinés : K = KDF(Hash(rD1) || Hash(rD2)).

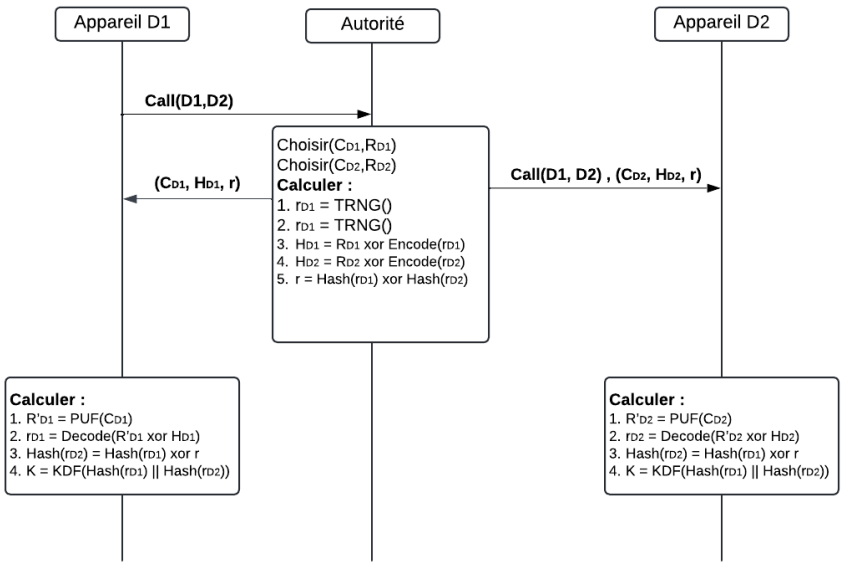


FIGURE 3.3 – Diagramme du troisiéme protocole

‘

3.5.5 Discussion et critique

Les auteurs ont proposé trois protocoles d'authentification basés sur les PUF pour les appareils IoT. La différence entre le premier protocole et les deux autres réside dans le fait que ces derniers partagent une clé symétrique pour les communications futures. Le premier protocole se base sur la comparaison d'un nonce et de la réponse R pour garantir l'authentification. Cependant, si un attaquant se fait passer pour l'autorité, il peut détourner ces protocoles, les rendant vulnérables car l'appareil n'a aucun moyen de vérifier l'identité de l'autre partie communicante. Bien que les deux autres protocoles partagent une clé pour les futures communications et que le troisième vise à garantir une authentification mutuelle entre les appareils, ces derniers n'ont également aucun moyen de vérifier s'ils partagent réellement la même clé [33].

Dans ces protocoles, l'autorité stocke plusieurs paires de défis. Si le nombre d'appareils augmente, le nombre de paires stockées par l'autorité doit également augmenter, ce qui peut entraîner une surcharge et ralentir le temps de recherche et de réponse.

Les deux premiers protocoles utilisent une seule opération cryptographique, tandis que le troisième emploie des opérations cryptographiques et des fonctions de hachage, ce qui peut consommer plus d'énergie. Cela peut poser un problème pour les appareils IoT ayant des ressources limitées.

3.6 Protocole d'authentification légère et d'échange de clés basé sur PUF pour IoT

Les auteurs de cet article ont présenté un protocole léger d'authentification que nous appellerons protocole 4 pour les dispositifs IoT utilisant les PUF. Dans la phase d'inscription, le serveur génère un défi Ci qu'il envoie au nœud. Ensuite, ce nœud crée une réponse Ri qu'il renvoie au serveur. Le serveur stocke la paire de clés Ci-Ri avec l'ID du nœud, ce qui signifie qu'une seule paire de clés est stockée auprès du serveur.

Dans la phase d'authentification de ce protocole, le nœud envoie son ID au serveur, qui recherche dans sa base de données la paire Ci-Ri précédemment stockée. Ensuite, il génère un nonce et le masque en le concaténant avec Ri à l'aide de l'opérateur XOR :
Mnode ← Ri XOR nonce. Ensuite, il calcule le hachage du défi Ci concaténé avec Ri et le nonce : Hnode ← hash(Ci|| Ri||nonce), Ensuite, il génère une séquence de bits pseudo-aléatoires en utilisant le registre à décalage LFSR, qui prend en entrée le nonce. Il envoie alors le défi Ci, le hachage Hnode et le message Mnode au nœud.

Le nœud, après avoir reçu le défi Ci, utilise sa fonction PUF pour générer la réponse Ri :
Ri = PUF(Ci). Ensuite, il calcule le nonce en effectuant un XOR entre la réponse générée et le message Mnode reçu : nonce ← Ri XOR Mnode, et vérifie le Hnode. Ensuite, il met à jour son défi en le remplaçant par le nonce collecté et calcule une réponse à ce nouveau défi :
Ri+1 ← PUF(Ci+1). Le nœud IoT améliore le nonce en nonce' en le passant dans un registre à décalage à rétroaction linéaire (LFSR), puis il calcule le message Mserver en effectuant un XOR entre la réponse calculée et nonce', et calcule le hachage du défi Ci+1 concaténé avec la réponse Ri+1 et nonce', et envoie au serveur le Mserver et le Hserver.

À la réception du message par le serveur, il calcule Ri+1 en effectuant un XOR entre le Mserver reçu et nonce' précédemment calculé, puis vérifie le Hserver. Ensuite, il met à jour le défi stocké dans la base de données en le remplaçant par nonce' et stocke cette nouvelle paire de clés (Ci+1,Ri+1). De cette manière, il garantit une authentification mutuelle entre les deux entités.

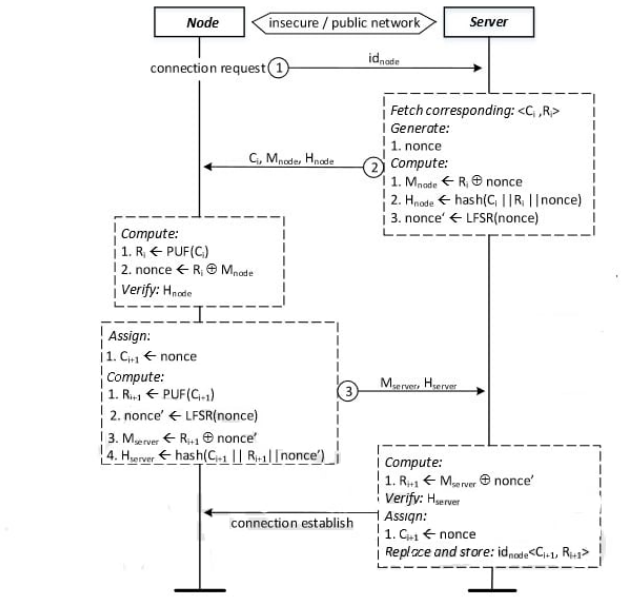


FIGURE 3.4 – Diagramme du protocole
[29]

Discussion et critique

Les auteurs ont proposé un protocole qui assure une authentification mutuelle entre un nœud et un serveur, renforçant ainsi la sécurité contre les attaques de l'homme du milieu. L'utilisation de *"nonces"*et l'utilisation de fonctions de hachage contribuent à la robustesse du protocole. Dans ce protocole, le serveur stocke une seule paire de défis-réponses pour chaque nœud, et lors de la mise à jour des paires, l'ancienne paire est remplacée par la nouvelle. Cela réduit la charge du serveur, même avec l'augmentation du nombre d'appareils, car il n'a à stocker qu'une seule paire par nœud, ce qui permet des recherches et des temps de réponse rapides. Toutefois, les diverses fonctions utilisées dans ce protocole, ainsi que les opérations et communications qu'il implique, peuvent entraîner une consommation d'énergie élevée, ce qui peut poser des problèmes pour les appareils disposant de ressources énergétiques limitées.

3.7 Protocole d'authentification légère basé sur PUF utilisant la reconnaissance de motifs secrets pour les dispositifs IoT contraints

Les auteurs de cet article ont introduit un protocole d'authentification léger pour les dispositifs IoT basé sur les PUF que nous appellerons protocole 5. Chaque

appareil est associé à un ensemble de schémas secrets de "g". Au début de chaque session d'authentification, un schéma de "g" est sélectionné au hasard parmi cet ensemble et reste confidentiel pour toutes les parties. Chaque appareil doit utiliser au moins deux schémas secrets, sinon la sélection aléatoire de "g" n'aurait pas de sens. Les schémas "g" sont complémentaires entre eux, de sorte que chaque entité ne stocke qu'une moitié du schéma tandis que l'autre moitié est générée lors de l'exécution.

Les deux entités collaborent pour créer un pseudo-défi aléatoire en générant chacune une moitié, puis les échangent. Le vérificateur reçoit la moitié générée par le prouveur, crée sa propre moitié et les concatène avant de renvoyer la partie qu'il a générée au prouveur. Le prouveur concatène ensuite la moitié reçue avec la sienne.

Le pseudo-défi du vérificateur est transformé en trois défis indépendants, donnant chacun une réponse unique. Cela produit une réponse de 3 bits RV, qui constitue le pseudo-défi du vérificateur. Ainsi, les pseudo-défis ne sont jamais utilisés comme entrées directes pour le circuit PUF.

Le protocole débute par un message d'initialisation et un échange d'un numéro d'identification PUF. Le schéma g pour la session est choisi parmi un ensemble de schémas secrets assignés à l'appareil. Après l'échange de défis transformés effectué m fois pour obtenir les réponses RV, où le pseudo-défi du vérificateur est transformé en trois défis indépendants, produisant chacun une réponse unique, le prouveur génère deux pseudo-défis aléatoires, C2 et C3, de manière à ce que PUF(Ti(C2)) xor PUF(Tj(C3)) = g, où Ti et Tj sont des fonctions de transformation de défis en un défi d'entrée unique, puis envoie C2 et C3 au vérificateur.

Comme le vérificateur n'a pas accès à la valeur de "g", il conserve la trace de PUF(Ti(C1)) xor PUF(Tj(C3)) = "g'". Le vérificateur vérifie si le schéma "g'" correspond à l'un des schémas assignés à l'appareil. Si le schéma "g'" ne correspond à aucun des schémas assignés, le vérificateur refuse d'authentifier l'autre partie.

Discussion et critique

Ce protocole repose sur une authentification mutuelle robuste, où le pseudo-défi est généré conjointement par les deux entités. La moitié de ce défi est échangée entre les entités, ce qui rend difficile pour un tiers de l'identifier. Chaque session

d'authentification est unique grâce à la sélection aléatoire d'un schéma secret "g". Le protocole est protégé contre les attaques tolérantes en utilisant des fonctions de transformation de défis, permettant aux PUF de travailler avec les défis transformés plutôt qu'avec les pseudo-défis eux-mêmes.

Le protocole se base sur des opérations simples au niveau des bits et des calculs basiques, évitant ainsi le recours à des fonctions cryptographiques ou de hachage qui consomment davantage de ressources énergétiques. Chaque appareil doit stocker au moins deux schémas secrets. Si l'appareil est un serveur et que le nombre d'appareils souhaitant s'authentifier auprès de lui augmente, cela entraînera une augmentation des schémas secrets nécessaires, ce qui pourrait provoquer une surcharge du serveur.

3.8 Tableau de comparaison

Articles	**Robustesse**	**Scalabilité**	**Complexité**
Authentification légère et communication sécurisée Adaptées aux appareils IoT	★★	★★	★★★
Protocole d'authentification légère et d'échange de clés basé sur PUF pour IoT	★★★★	★★★★	★★
Protocole d'authentification légère basé sur PUF utilisant la reconnaissance de motifs secrets pour les dispositifs IoT contraints	★★★★	★★	★★★★

Nous tenons à préciser que :

— ★★ signifie faible

— ★★★ signifie moyen

— ★★★★ signifie élevé

3.9 Amélioration du troisiéme protocole de l'article [?]

D'aprés l'article [33], Les protocoles décrits dans le premier article sont critiqués pour leur manque de sécurité. Bien que le dernier protocole, qui vise à obtenir une authentification mutuelle, soit plus sécurisé que les précédents, il présente toujours des failles, comme le confirme notre analyse avec l'outil AVISPA dans le chapitre suivant. Le principal problème réside dans l'incapacité de chaque appareil à vérifier si sa clé est en corrélation avec celle de l'autre appareil. En cas d'attaque, un intrus peut usurper le rôle de l'autorité, intercepter un des messages destinés à l'un des appareils, le modifier, engendrant ainsi la réception de deux clés distinctes par les deux appareils, sans que ces derniers ne soient informés de cette altération

Nous proposons deux solutions pour remédier à cette situation. La première consiste à ce que les appareils confirment leur clé auprès de l'autorité, ou qu'ils communiquent directement entre eux pour la confirmer. Nous avons tenté d'améliorer le protocole en adoptant la première solution. En effet, dans cette approche, l'autorité calcule également la clé partagée. Cette clé est ensuite partagée entre les trois entités. L'autorité inclut alors le hachage de cette clé dans le message envoyé. Lorsque l'appareil reçoit le message, il calcule la clé et son hachage, qu'il compare ensuite avec celui reçu. Ainsi, en cas de modification d'un message par un attaquant, l'appareil peut détecter l'altération grâce au calcul du hachage, ce qui lui permet de reconnaître une tentative d'usurpation de l'autorité.

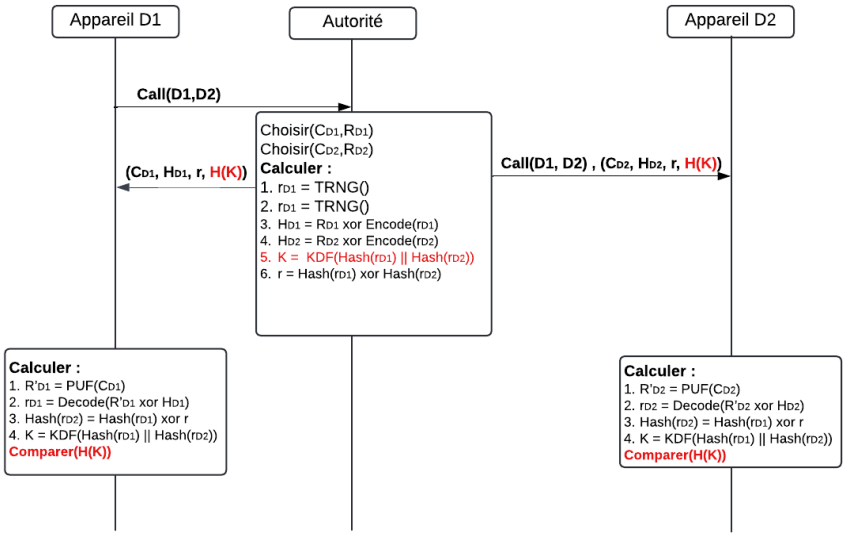


FIGURE 3.5 – Diagramme du protocole amélioré

3.10 Conclusion

L'authentification des appareils IoT est essentielle pour assurer la sécurité et l'intégrité des réseaux connectés. Avec la croissance exponentielle des appareils IoT, les risques de cyberattaques augmentent, rendant indispensable l'implémentation de protocoles d'authentification robustes.

Dans ce chapitre, nous avons examiné trois articles proposant des protocoles d'authentification basés sur des PUFs pour les appareils IoT. Chaque protocole présente des avantages et des inconvénients. Nous avons utilisé trois critères de comparaison : la robustesse, la scalabilité et la complexité. En nous basant sur ces critères, nous avons discuté en détail de ces protocoles et proposé une amélioration pour le dernier protocole de l'article [28].

Dans le chapitre suivant, nous vérifierons la sécurité du dernier protocole du premier article, ainsi que celle du protocole proposé dans l'article [29] et du protocole amélioré que nous avons proposé, en utilisant l'outil de vérification AVISPA.

Chapitre 4

Analyse des performances des protocoles

4.1 Introduction

La vérification formelle des protocoles de sécurité revêt une importance dans l'assurance de leur robustesse contre les attaques potentielles. Elle permet d'identifier et de corriger les vulnérabilités avant leur déploiement opérationnel, garantissant ainsi une protection adéquate des systèmes et des données sensibles. Dans ce domaine, plusieurs outils d'analyse sont disponibles, parmi lesquels la plateforme AVISPA se distingue comme l'outil le plus réputé [34].

Ce chapitre se focalise sur la vérification des protocoles d'authentification basés sur les PUFs en employant l'outil SPAN/AVISPA et le langage HLPSL. L'intégration de cet outil permet une analyse et une validation formelles des propriétés de sécurité de ces protocoles, assurant ainsi leur robustesse et leur fiabilité.

4.2 Définition de l'outil Avispa

AVISPA est un outil dédié à la validation automatisée de protocoles et applications critiques pour la sécurité sur Internet. Cet outil utilise un langage formel modulaire et expressif qui permet de spécifier les protocoles ainsi que leurs propriétés de sécurité. Il intègre également divers back-ends exécutant une gamme de techniques d'analyse automatique avancées pour vérifier ces spécifications [35]. Si

une vulnérabilité est détectée dans un protocole, AVISPA peut l'illustrer à l'aide d'un diagramme de séquence de messages, facilitant ainsi la compréhension de la faille et la manière dont elle peut être exploitée [36].Quatre outils principaux au sein d'AVISPA se chargent de cette vérification automatisée, chacun appliquant des méthodologies différentes pour assurer une analyse complète et efficace des protocoles de sécurité.

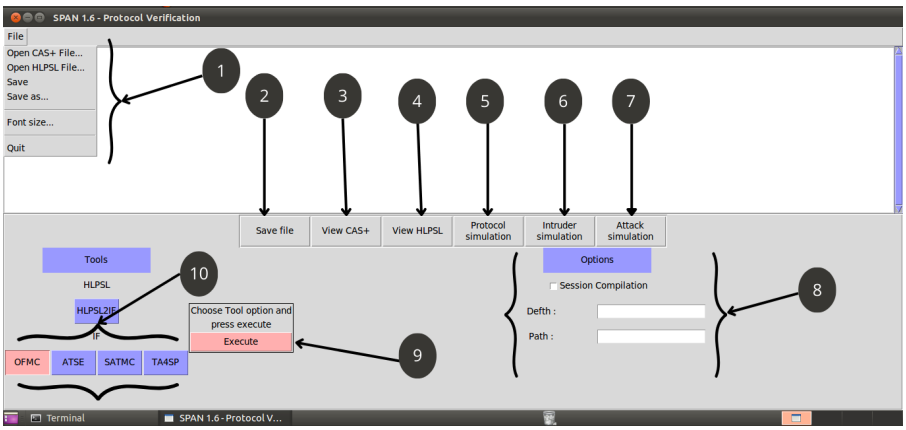


FIGURE 4.1 – Interface de SPAN AVISPA

1 : Ouvrir ou enregistrer une spécification HLPSL ou CAS
2 : Enregistrer le fichier HLPSL ou CAS
3 : Voir le code CAS
4 : Voir le code HLPSL
5 : Animation de protocole
6 : Animation de protocole avec un intrus
7 : Voir le code HLPSL
8 : Options de l'outil sélectionné
9 : Exécuter le code avec l'outil sélectionné
10 : Outils de vérification

4.3 Architecture de Avispa

AVISPA, un outil avancé pour la validation automatisée des protocoles et applications de sécurité sur Internet, utilise un processus structuré pour évaluer la robustesse des protocoles de sécurité. Au cœur de cet outil se trouve le langage

de spécification HLPSL, conçu pour une description modulaire et abstraite des protocoles et de leurs propriétés de sécurité.

Le processus commence par la conversion des spécifications HLPSL en un format intermédiaire (IF) à l'aide d'un traducteur appelé hlpsl2if. Le format IF est un langage de niveau inférieur, optimisé pour être interprété directement par les différents back-ends d'AVISPA. Ce format intermédiaire est essentiel car il sert de pont entre le langage de spécification haut-niveau et les mécanismes d'analyse bas-niveau, facilitant ainsi l'intégration de nouveaux outils d'analyse qui peuvent utiliser IF comme leur langage d'entrée.

Une fois la spécification d'un protocole traduite en IF, elle est analysée par les back-ends d'AVISPA pour déterminer si les objectifs de sécurité sont atteints ou violés. AVISPA intègre quatre back-ends, chacun offre une méthode unique de vérification [37] :

OFMC(The On-the-fly Model-Checker) : L'outil OFMC, originaire du projet AVISS et développé sous AVISPA, effectue une vérification bornée des protocoles en utilisant une spécification IF pour naviguer à travers les systèmes de transitions. Il applique des techniques symboliques correctes et complètes, et supporte des opérateurs avec propriétés algébriques telles que l'opérateur OU exclusif et l'Exponentielle. OFMC est essentiel pour analyser des protocoles où les propriétés algébriques des fonctions cryptographiques sont critiques, facilitant la détection rapide des attaques et la validation de l'exactitude des protocoles [38] [39].

CL-AtSe (Constraint-Logic-based Attack Searcher) : C'est un outil qui s'appuie sur la résolution de contraintes en intégrant des heuristiques de simplification et des techniques d'élimination de redondances [40]. Il permet de convertir une spécification d'un protocole de sécurité sous forme de relations de transition IF en un ensemble de contraintes. Ces contraintes peuvent être exploitées pour identifier d'éventuelles attaques sur le protocole concerné. L'outil applique la résolution de contraintes en employant des heuristiques de simplification et des techniques d'élimination de redondances [38].

SATMC (SAT-based Model-Checker) : est un vérificateur de modèles

bornés basé sur la SAT qui analyse les systèmes critiques de sécurité [41]. Il crée une formule propositionnelle pour représenter un déploiement borné du système de transition IF, l'état initial, et les états de violation des propriétés de sûreté en IF (et donc en HLPSL). La formule est résolue par un solveur SAT choisi parmi zCHAFF, mCHAFF, SIM, ou SATO. Tout modèle satisfaisant la formule est renvoyé sous forme d'attaque. En recherchant des violations potentielles dans un protocole, il génère une formule représentant la violation et la transforme en attaque [38].

TA4SP (Tree-Automata-based Protocol Analyzer) : En utilisant un langage d'arbres régulier avec réécriture, on évalue les connaissances de l'intrus en produisant des sous- et sur-approximations. Cet outil de vérification se distingue par le fait qu'il réalise soit une sur-approximation soit une sous-approximation des connaissances de l'intrus à partir d'un état initial, en utilisant des automates d'arbres. Grâce à cette méthode, il est possible de déterminer si un état spécifique est accessible ou non, ainsi que si l'intrus peut acquérir certaines connaissances ou non, ce qui permet de conclure qu'il n'y a pas d'attaque sur le secret pour des scénarios exécutés indéfiniment [38] .

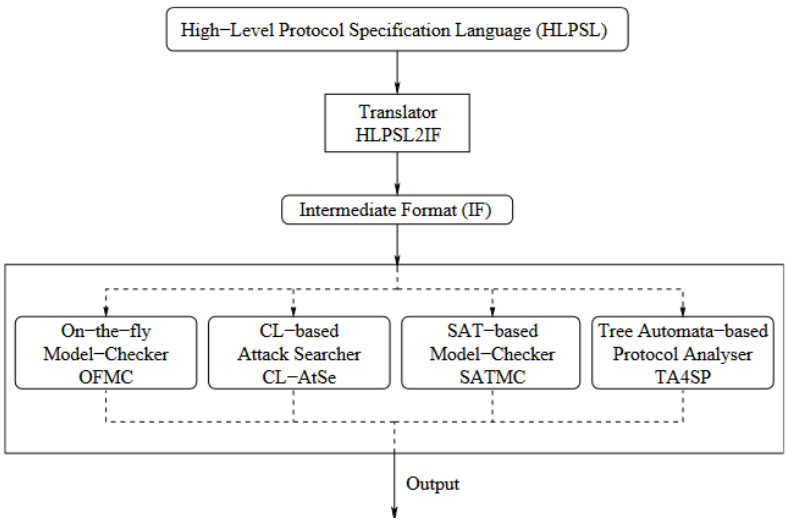
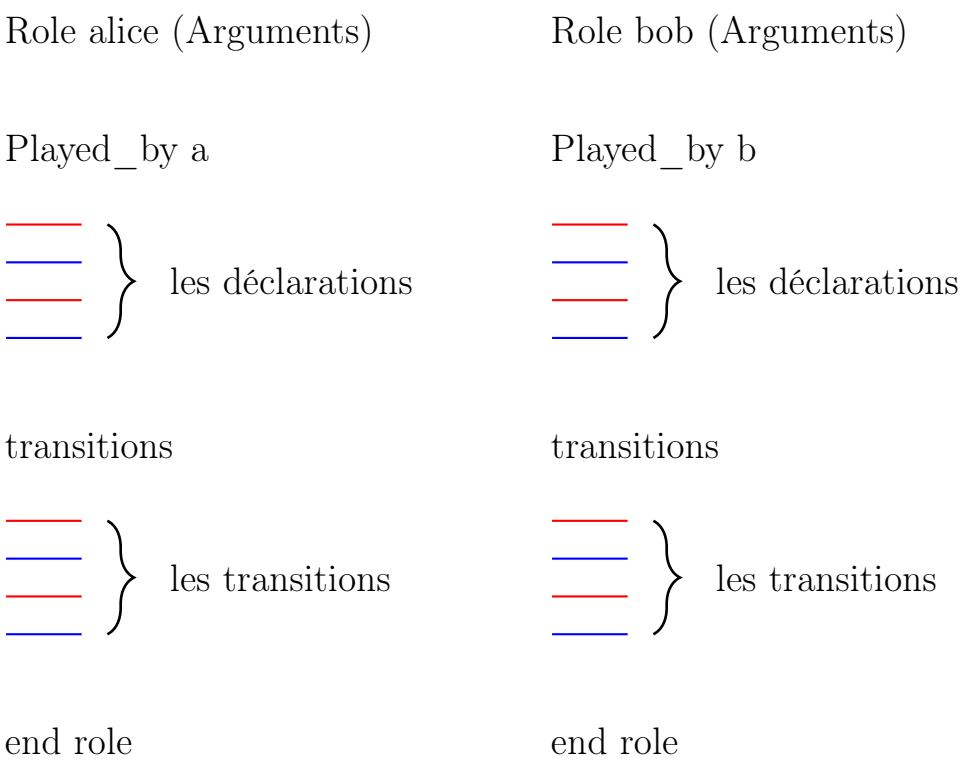


FIGURE 4.2 – Architecture de AVISPA

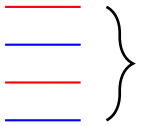
4.4 Définition de HLPSL

HLPSL est un langage formel destiné à spécifier des protocoles de sécurité informatique de manière modulaire et expressive. Il se base sur des descriptions de rôles, ce qui lui permet de représenter les différentes entités participant à un protocole et leurs interactions de manière claire et structurée. Ce langage supporte un ensemble de primitives cryptographiques, telles que les clés symétriques et asymétriques ainsi que les fonctions de hachage, en intégrant leurs propriétés algébriques telles que l'opération XOR ou l'exponentiation. L'objectif principal des spécifications HLPSL est de vérifier les propriétés de sécurité des protocoles, notamment l'authentification et la confidentialité. Pour ce faire, HLPSL permet de représenter les protocoles sous forme de systèmes d'états et de transitions, ce qui facilite l'analyse formelle à l'aide de logiques temporelles linéaires (LTL). Les spécifications HLPSL sont généralement divisées en deux catégories de rôles distinctes : les rôles de base, qui représentent les agents participant directement aux protocoles, et les rôles de composition, qui décrivent les scénarios impliquant plusieurs rôles de base [34].

4.5 Structuration du HLPSL

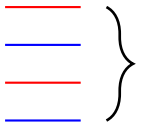


role session(Arguments)

 les déclarations

alice(les arguments) $\wedge$ bob(les arguments) composition de session

role d'environnement

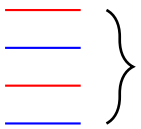
 les déclarations

Composisition

alice(les arguments) $\wedge$ bob(les arguments) composition de session

end role

goal

 propriétés à vérifie

end role

role environnement()

4.6 Vérification du protocole 3

4.6.1 Scénario du protocole

```
role environment()
def=
  const a,b,c: agent,
        call:text,
        sec1,ab,ac: protocol_id,


        h,puf:hash_func
intruder_knowledge = {a,b,c,h,puf}
  composition
        session(a,b,c,h,puf,call)

end role

goal
  authentication_on ab,ac
end goal

environment()
```

FIGURE 4.3 – Scénario du protocole 3

Dans notre travail, la fonction PUF est considérée comme une fonction de hachage, car dans le langage HLPSL, nous ne pouvons pas définir de fonction. Dans ce cas, nous utilisons les fonctions prédéfinies dans HLPSL qui sont similaires à notre fonction. Comme la fonction de hachage prend une entrée et génère une sortie, elle correspond exactement à ce que nous recherchions.

4.6.2 Notations

Notations	Signification
a,c	appareils
b	autorité
call	message d'initiation du protocole
h	fonction de hashage
puf	exprime la fonction puf

TABLE 4.1 – notations du protocole 3

4.6.3 Execution du protocole

Une fois le code terminé, on clique sur "File" puis sur "Open HLPSL file". Ensuite, on choisit un outil de vérification parmi les quatre disponibles pour notre cas, nous avons choisi OFMC, puis on clique sur "Exécuter" :

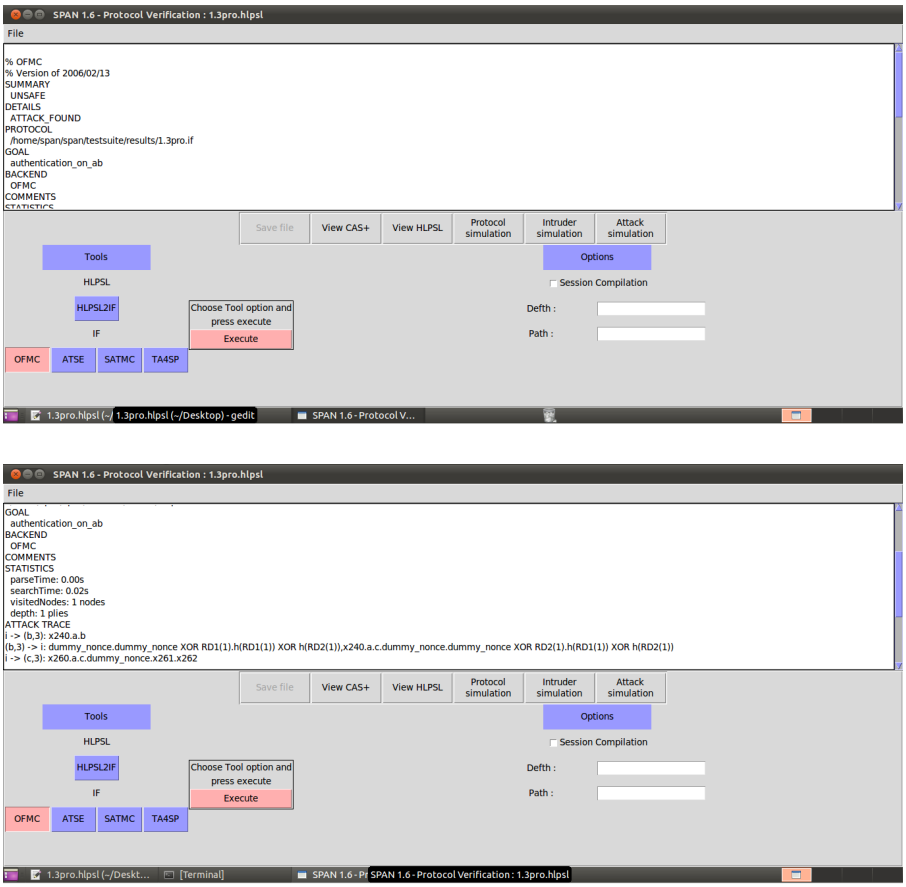


FIGURE 4.4 – Résultat d'execution du protocole 3

— La section SUMMARY indique si le protocole est sécurisé ou non. Dans notre cas, le protocole n'est pas sécurisé.

— La section DETAILS précise les conditions sous lesquelles le protocole est déclaré sécurisé ou non, les circonstances dans lesquelles une attaque est détectée, et les raisons pour lesquelles l'analyse n'a pas été concluante. Dans notre cas, elle indique qu'une attaque a été trouvée.

— La section PROTOCOL rappelle le nom du protocole analysé.

— La section GOAL présente l'objectif de l'analyse, par exemple, l'authentification entre Alice et Bob.

— La section BACKEND désigne le traducteur des spécifications HLPSL.

Ce résultat montre que le protocole n'est pas sur comme mentionné dans l'articlesous quelles conditions le protocole est déclaré sur ou non, sous quelles conditions une attaque est trouvée et finalement pourquoi l'analyse n'a pas été concluante

4.6.4 Simulation du protocole

Après avoir exécuté le code HLPSL du protocole sans rencontrer d'erreurs, on clique sur le bouton "Protocol Simulation" pour visualiser la simulation du protocole de la manière suivante :

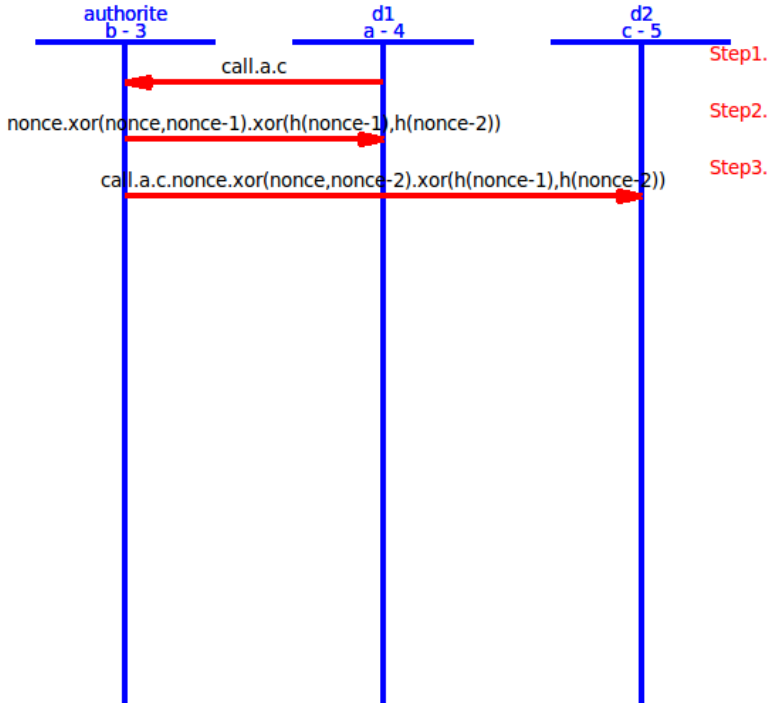


FIGURE 4.5 – Vérification du protocole 3

4.7 Vérification du protocole amélioré

4.7.1 Code du protocole

— Role de l'appareil D1

```
role d1(A,B,C: agent, H,Puf,KDF:hash_func, Call:text, SND, RCV: channel(dy)) played_by A
def=
  local State:nat,
  RD1,RD2,CD1,R1,R,HD1:text,
  ENCODE:symmetric_key,
  K:message

 const ab,secc:protocol_id
  init State:= 0
  transition
        1. State = 0 /\ RCV(start)  =|> State' := 2
        /\ SND(Call.A.B)
        2. State = 2 /\ RCV(CD1.R'.HD1'.H(K'))  =|> State' := 4
        /\ R1' := Puf(CD1)
        /\ RD1' := xor(R1',HD1')
        /\ K' := KDF (H(RD1').xor(H(RD1'),R'))
        /\ request(A,B, ab,K)

end role
```

FIGURE 4.6 – Le role de l'agent D1

— Role de l'autorité

```
role authorite(A,B,C: agent, H,KDF:hash_func, SND, RCV: channel(dy)) played_by B
def=
   local State:nat,
    Call,RD1,RD2,CD1,CD2,R1,R2,R,HD1,HD2:text,
    ENCODE:symmetric_key,
    K:message
 const ab,sec,ac:protocol_id
   init State:= 1
   transition
        1. State = 1 /\ RCV(Call'.A.B) =|> State':=3
        /\ RD1' := new ()
        /\ RD2' := new ()
        /\ R' := xor(H(RD1'),H(RD2'))
        /\ HD1' := xor(R1,RD1')
        /\ HD2' := xor(R1,RD2')
        /\ K' := KDF (H(RD1').H(RD2'))
        /\ SND(CD1.R'.HD1'.H(K'))
        /\ SND(Call'.A.C.CD2.R'.HD2'.H(K'))
        /\ witness(B, A, ab, K)
        /\ witness(B, C, ac, K)
```

FIGURE 4.7 – Role de l'agent Authorité

— Role de l'appareil D2

```
role d2(A,B,C: agent, H,Puf,KDF:hash_func, SND, RCV: channel(dy)) played_by C
def=
   local State:nat,
    Call,RD1,RD2,CD1,CD2,R1,R2,R,HD1,HD2:text,
    KDC:symmetric_key,
    K:message
 const ab,ac:protocol_id
   init State:= 3
   transition
         1. State = 3 /\ RCV(Call'.A.C.CD2.R'.HD2'.H(K'))  =|> State' := 5
         /\ R2' := Puf(CD2)
        /\ RD2' := xor(R2',HD2')
        /\ K' := KDF (H(RD2').xor(H(RD2'),R'))
        /\ request(C,B, ac,K)
end role
```

FIGURE 4.8 – Role de l'agent D2

— Role de session

```
role session (A,B,C: agent,H,Puf,KDF:hash_func,Call:text)
def=
  local SA, SB, RA, RB: channel (dy)
  composition
         authorite(A,B,C,H,KDF,SB,RB) /\ d1(A,B,C,H,Puf,KDF,Call,SA,RA) /\ d2 (A,B,C,H,Puf,KDF,SB,RB)
end role
```

FIGURE 4.9 – Role de la session du protocole

— **Scénario du protocole**

```
role environment()
def=
  const a,b,c: agent,
        call:text,
        sec1,ab,ac: protocol_id,
        h,puf,kdf:hash_func
intruder_knowledge = {a,b,c,h,puf}
  composition
        session(a,b,c,h,puf,kdf,call)
end role
goal
  authentication_on ab,ac
end goal

environment()
```

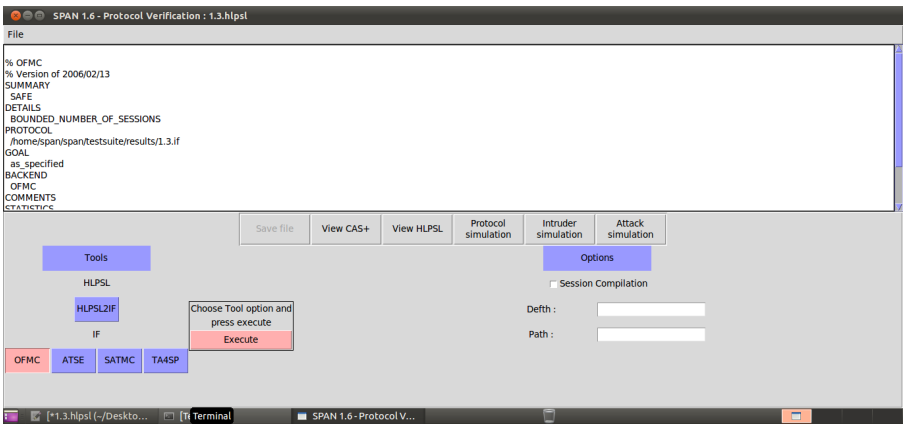
FIGURE 4.10 – Scénario du protocole amélioré

4.7.2 Notations

Ce sont les mêmes notations que dans le tableau précédent, sauf que nous avons ajouté KDF, qui signifie fonction de dérivation de clé.

4.7.3 Execution du protocole

Après avoir finalisé le code, nous avons l'exécuter de la même manière que précédemment pour vérifier la sécurité du protocole. Les résultats présentés ci-dessus ont été générés par l'outil OFMC.



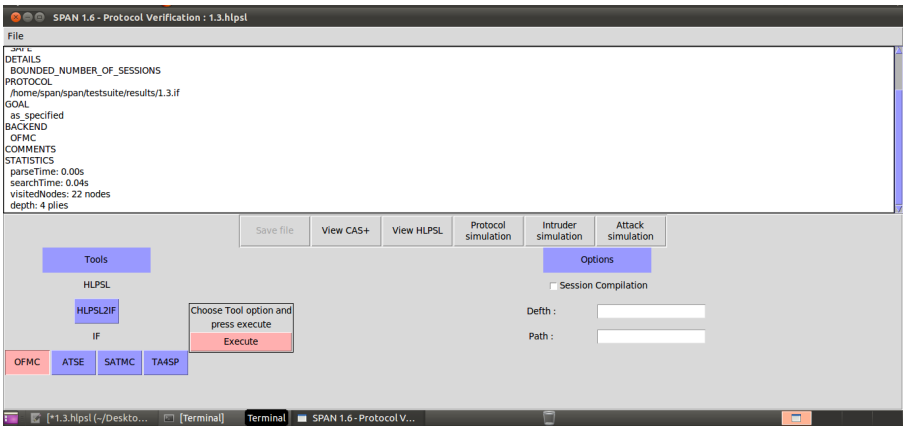


FIGURE 4.11 – Résultat du protocole amélioré

D'aprés la section SUMMURY on voit bien que que le protocole que nous avons proposé est sûr.

4.7.4 Simulation du protocole

Après avoir exécuté le protocole, nous lançons la simulation pour observer les messages échangés entre les différentes entités, qui sont visualisés graphiquement comme illustré dans la figure suivante.

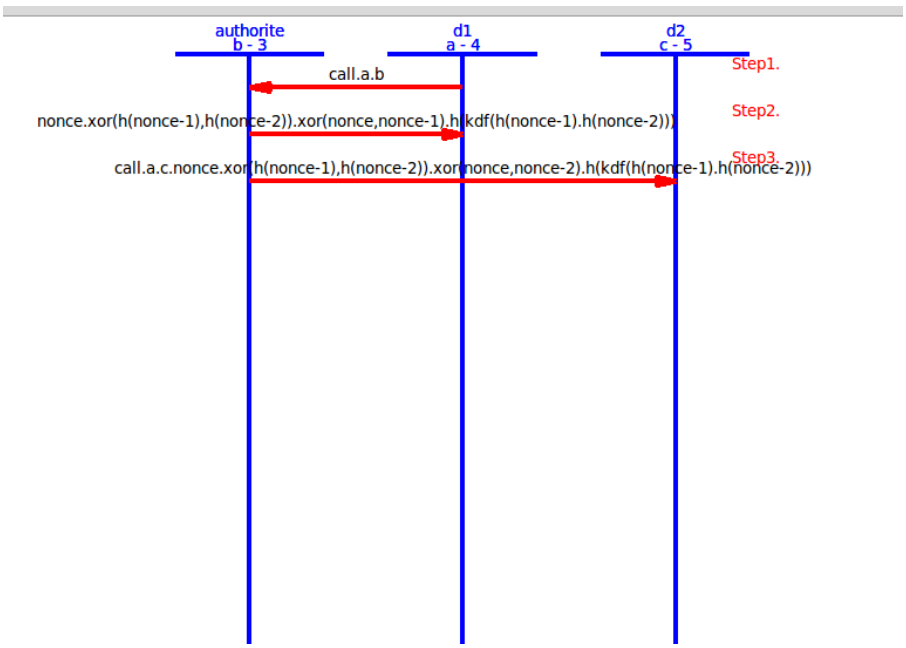


FIGURE 4.12 – Vérification du protocole amélioré

4.8 Vérification du protocole 4

4.8.1 Scénario du protocole

```
role environment()
def=
  const a,b: agent,
        id,ci,ri:text,
        sec1,ab,ac: protocol_id,
        h,puf,lfsr:hash_func
intruder_knowledge = {a,b,h,puf,lfsr}
  composition
        session(a,b,h,puf,lfsr,id,ci,ri)

end role

goal
  authentication_on ab,ac
end goal

environment()
```

FIGURE 4.13 – Scénario du duexiéme protocole

4.8.2 Notations

Notations	Signification
a	Le serveur
b	le noeud
id	L'identifiant de la paire de défi
ci	le défis qui est stocké auprés du seveur
ri	la reponse au défis qui est stocké auprés du seveur
h	la fonction de hachage
puf	la fonction PUF
lsfr	la fonction LSFR

TABLE 4.2 – Les notations du deuxiéme protocole

4.8.3 Execution du protocole

Une fois le code finalisé, nous l'avons exécuté de la même manière que précédemment pour valider la sécurité du protocole. Les résultats affichés ci-dessus :

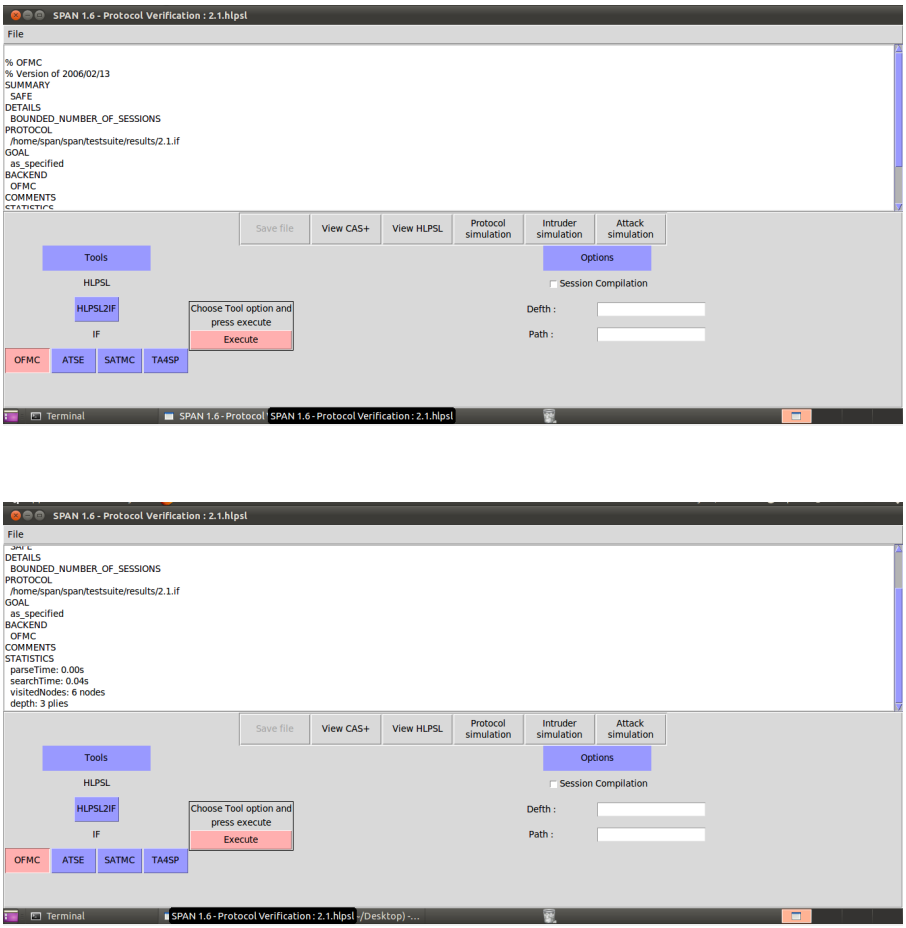


FIGURE 4.14 – Résultat d'éxecution du deuxieme protocole

D'aprés la section SUMMURY on voit bien que le protocole proposé dans l'article [29] est sûr.

4.9 Conclusion

La vérification formelle est une méthode permettant de décrire mathématiquement un protocole afin de s'assurer de la fiabilité des protocoles cryptographiques. Dans ce chapitre, nous avons présenté quelques notions de base sur les outils de vérification SPAN et AVISPA, ainsi que sur le langage HLPSL. Nous avons également vérifié la sécurité des protocoles décrits dans le troisième chapitre ainsi que celle du protocole amélioré. Nos analyses ont confirmé que les protocoles proposés dans l'article [?] sont vulnérables, tandis que l'amélioration proposée est sécurisée. À l'avenir, nous envisageons d'étudier des protocoles d'authentification plus complexes pour les appareils IoT et de tenter de découvrir des vulnérabilités dans ces protocoles. Cela nous permettra de les améliorer par la suite. De plus, nous souhaitons approfondir notre utilisation de l'outil de

vérification AVISPA pour une évaluation plus rigoureuse de la sécurité de ces protocoles.

Conclusion générale

Dans le cadre de ce mémoire sur les PUFs dans l'IoT, plusieurs constats essentiels se dégagent. La sécurisation et la gestion des clés des dispositifs IoT représentent un défi majeur dans un environnement de connectivité croissante, exposant ces dispositifs à des risques accrus. Les PUFs émergent comme une solution innovante et efficace pour répondre à ces défis, grâce à leur capacité à générer des identifiants uniques et à renforcer les protocoles d'authentification.

Cette étude a permis d'atteindre plusieurs objectifs clés. Tout d'abord, une analyse approfondie des différents types de PUFs a été menée, mettant en lumière leurs caractéristiques distinctives et leurs applications potentielles dans le domaine de l'IoT. Les PUFs en silicium, notamment, ont démontré des promesses significatives en termes de robustesse et de coût de production.

Ensuite, l'élaboration et la vérification de protocoles d'authentification basés sur les PUFs ont constitué une autre contribution majeure de cette recherche. Les protocoles proposés ont été rigoureusement évalués à l'aide de l'outil Avispa pour garantir leur sécurité et leur efficacité contre diverses attaques. Les simulations réalisées ont confirmé leur applicabilité pratique et leurs performances dans des scénarios réels.

Les PUFs fournissent une solution de sécurité intrinsèque, pratiquement impossible à contourner par des méthodes de clonage ou de falsification. De plus, les protocoles d'authentification développés démontrent une capacité à protéger efficacement les communications et les données des dispositifs connectés.

Ce projet a été d'une valeur pédagogique immense, car il nous a permis d'approfondir notre compréhension du domaine des PUFs, en particulier en ce qui concerne la sécurité. De plus, il nous a donné l'occasion de nous familiariser avec des environnements de travail tels que SPAN/AVISPA et HLPSL, que nous découvrions pour la première fois.

Cependant, malgré les avancées significatives réalisées, ce domaine de recherche demeure en évolution. De plus, les protocoles d'authentification développés démontrent une capacité à protéger efficacement les communications et les données des dispositifs connectés. Des améliorations continues sont nécessaires pour rendre les PUFs encore plus résistants aux attaques sophistiquées et pour surmonter les défis techniques et logistiques de leur intégration à grande échelle dans des systèmes IoT complexes.

Bibliographie

[1] I. Saleh, "Les enjeux et les défis de l'Internet des Objets (IdO)," *Internet des objets*, Apr. 2017. [Online]. Available : https://www.openscience.fr/Les-enjeux-et-les-defis-de-l-Internet-des-Objets-IdO

[2] B. O. TRIDI, HAMOUDI, "Surveillance médicale a distance basée sur l'IoT," UNIVERSITÉ AKLI MOHAND OULHADJ DE BOUIRA, mémoire, 2018.

[3] R. Langmann, "Automatisierungssysteme mit web-technologien," *atp edition - Automatisierungstechnische Praxis*, vol. 56, 09 2014.

[4] M. Martin, "Comment l'évolution de l'internet des objets impacte-t-elle le marketing et le mode de consommation ?" UNIVERSITE CATHOLIQUE DE LOUVAIN, mémoire, 2022.

[5] O. A. TALEB, MANKOURI, "Programmation de la sécurité Internet des Objet, Etude de cas module WIFI Electric imp," UNIVERSITÉ DE TLEMCEN, mémoire, May 2016.

[6] K. K. Patel, S. M. Patel, and P. Scholar, "Internet of things-iot : Definition, characteristics, architecture, enabling technologies, application & future challenges," 2016. [Online]. Available : https://api.semanticscholar.org/CorpusID:189924732

[7] H. TAKROUNI, "Dévelopemment d'une plateforme internet des objets À connéctivité hybrdide," UNIVERSITÉ DU QUÉBEC EN OUTAOUAIS, mémoire, Oct. 2020.

[8] S. Elhadi, A. Marzak, N. Sael, and S. Merzouk, "Comparative Study of IoT Protocols," *SSRN Electronic Journal*, May 2018. [Online]. Available : https://www.ssrn.com/abstract=3186315

[9] F. Masmali, S. Miah, and N. Noman, *Different Applications and Technologies of Internet of Things (IoT)*, 01 2023, pp. 41–54.

[10] “Sécurisation des objets connectés de santé | Veyan,” consulté le 2024-05-20. [Online]. Available : https://www.veyan.fr/securite-des-objets-connectes-dans-le-domaine-de-la-sante/

[11] A. MIMI, “Suivi médicale à distance dans l’internet des objets. mémoire,” UNIVERSITÉ MOHAMED KHIDER – BISKRA, mémoire, 2021.

[12] Y. HARBI, “Security in Internet of Things.” FERHAT ABBAS UNIVERSITY SETIF 1, thèse de doctorat, 2021.

[13] L. CHALAL and S. SIROUAKNE, “Gestion des cl´es dans l’internet des objets,” Universit´e A/Mira de B´ejaia, mémoire, 2016.

[14] “Comment l’industrie fait sa révolution avec l’Internet des Objets ? | E-media, the Econocom blog.” [Online]. Available : https://blog.econocom.com/blog/comment-lindustrie-fait-sa-revolution-avec-linternet-des-objets/

[15] “Sécurité internet des objets | IoT Journey,” 2024. [Online]. Available : https://iotjourney.orange.com/fr-FR/explorer/les-solutions-iot/securite-internet-des-objets

[16] “Qu’est-ce que la sécurité des IoT ? | Fortinet,” 2024. [Online]. Available : https://www.fortinet.com/fr/resources/cyberglossary/iot-security

[17] I. SAFFRAY, “Massification des objets connectés : vers un renforcement de l’autdit hadrware ?” 2020.

[18] B. Halak, *Physically Unclonable Functions*. Cham : Springer International Publishing, 2018. [Online]. Available : http://link.springer.com/10.1007/978-3-319-76804-5

[19] Y. Cao and al, “Advances in Physical Unclonable Functions Based on New Technologies : A Comprehensive Review,” Dec. 2023. [Online]. Available : https://www.mdpi.com/2227-7390/12/1/77

[20] H. Balagopal, “A NOVEL PHYSICAL UNCLONABLE FUNCTION (PUF) FEATURING 0.113 FJ/B FOR IOT DEVICES.”

[21] K. Lounis and M. Zulkernine, “Lessons Learned : Analysis of PUF-based Authentication Protocols for IoT,” Jun. 2023. [Online]. Available : https://dl.acm.org/doi/10.1145/3487060

[22] S. Eiroa, I. Baturone, A. J. Acosta, and J. Dávila, "Using Physical Unclonable Functions for Hardware Authentication : A Survey."

[23] C. Mesaritakis and al, "Physical Unclonable Function based on a Multi-Mode Optical Waveguide," Jun. 2018. [Online]. Available : https://www.nature.com/articles/s41598-018-28008-6

[24] "Revêtement pcb 5 minutes, laissez-vous savoir comment choisir la fonction," consulté le 2024-05-20. [Online]. Available : https://pcbassemblyfrance.com/revetement-pcb.html

[25] S. Hemavathy and V. S. K. Bhaaskaran, "Arbiter PUF—A Review of Design, Composition, and Security Aspects," 2023. [Online]. Available : https://ieeexplore.ieee.org/document/10091112/

[26] A. Shamsoshoara and al, "A survey on physical unclonable function (PUF)-based security solutions for Internet of Things," Dec. 2020. [Online]. Available : https://linkinghub.elsevier.com/retrieve/pii/S1389128620312275

[27] B. Vincent, "Théories de l'intrus pour la vérification des protocoles cryptographiques," 'ÉCOLE NORMALE SUPERIEUR DE CACHAN, thése, 2006.

[28] S. Buchovecká and al, "Lightweight Authentication and Secure Communication Suitable for IoT Devices :." Valletta, Malta : SCITEPRESS - Science and Technology Publications, 2020, pp. 75–83. [Online]. Available : http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0008959600750083

[29] S. Roy and al, "PUF based Lightweight Authentication and Key Exchange Protocol for IoT :," 2021. [Online]. Available : https://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0010550906980703

[30] T. A. Idriss and al, "A Lightweight PUF-Based Authentication Protocol Using Secret Pattern Recognition for Constrained IoT Devices," 2021. [Online]. Available : https://ieeexplore.ieee.org/document/9444356/

[31] A. Fayad, "Secure authentication protocol for Internet of Things," Ph.D. dissertation, Feb. 2021.

[32] G. Abdourahime and M. O. Ndiaye, "Proposition d'une architecture d'authentification mutuelle pour la gestion de la mobilité et de la disponibilité

dans les IoT-Fog : Cas d'un bracelet électronique," 2023. [Online]. Available : https://hal.science/hal-04299939/

[33] T. Rabas, "Verification of PUF-based IoT Protocols with AVISPA and Scyther :," Lisbon, Portugal, 2022. [Online]. Available : https://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0011299000003283

[34] N. Chikouche, U. de M'sila, and M. Benmohammed, "Vérification automatique des protocoles d'authentification des systèmes RFID."

[35] "AVISPA v1.1 User Manual," Jun. 2006. [Online]. Available : https://www.avispa-project.org/package/user-manual

[36] A. Armando and al, "The AVISPA Tool for the Automated Validation of Internet Security Protocols and Applications," in *Computer Aided Verification*, 2005, series Title : Lecture Notes in Computer Science. [Online]. Available : http://link.springer.com/10.1007/11513988_27

[37] "HLPSL Tutorial," Jun. 2006. [Online]. Available : https://www.avispa-project.org/package/tutorial.pdf

[38] S. M. MOHAMEDI, IKERBANE, "Vérification automatique d'un protocole de sécurité dans les systèmes RFIDs à base d'outils AVISPA & SPAN." UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU, mémoire, 2016.

[39] a. Aouamri, "Conception d'un schéma d'authentification sécurisé pour l'internet des drones," UNIVERSITE DE 8 MAI 1945 – GUELMA -, mémoire, Sep. 2021.

[40] K. S. BELLIL Amine, "Réalisation d'une application android de navigation par gps en utilisant des cartes osm," UNIVERSITE MOULOUD MAMMERI de TIZI-OUZOU, mémoire, 2013.

[41] A. Armando, R. Carbone, and L. Compagna, "Satmc : A sat-based model checker for security-critical systems," in *Tools and Algorithms for the Construction and Analysis of Systems*, E. Ábrahám and K. Havelund, Eds. Berlin, Heidelberg : Springer Berlin Heidelberg, 2014, pp. 31–45.

Résumé

L'impact croissant de l'Internet des objets (IoT) réside dans la connectivité entre objets physiques et virtuels, ouvrant de nouvelles perspectives dans des domaines comme la domotique, la santé et les infrastructures intelligentes. Nous examinons ensuite les Physically Unclonable Functions (PUFs), des technologies clés permettant de garantir l'authenticité des dispositifs IoT grâce à leurs propriétés uniques et difficilement reproduisibles. En se concentrant sur l'authentification, nous analysons divers protocoles basés sur les PUFs, et nous améliorons l'un d'eux pour renforcer sa sécurité et son efficacité. Enfin, nous évaluons rigoureusement ces protocoles améliorés pour assurer leur robustesse et leur conformité aux exigences de sécurité des réseaux IoT modernes.

Mots clés : authentification,, nternet des objets, protocoles, sécurité.

Abstract

The increasing impact of the Internet of Things (IoT) lies in the connectivity between physical and virtual objects, opening new perspectives in domains such as home automation, healthcare, and smart infrastructure. We then examine Physically Unclonable Functions (PUFs), key technologies ensuring the authenticity of IoT devices due to their unique and difficult-to-replicate properties. Focusing on authentication, we analyze various protocols based on PUFs and enhance one of them to bolster its security and efficiency. Finally, we rigorously evaluate these enhanced protocols to ensure their robustness and compliance with the security requirements of modern IoT networks.

Keywords : authentication, Internet of Things, protocols, security.